ON Semiconductor

Is Now

onsemi™

To learn more about onsemi™, please visit our website at
www.onsemi.com

CMOS 8-BIT MICROCONTROLLER

LC871D00 SERIES USER'S MANUAL

REV : 1.00

ON Semiconductor®

http://onsemi.com

ON Semiconductor

Digital Solution Division

Microcontroller & Flash Business Unit

Contents

Contents

Contents

Contents

Contents

1. Overview

1.1 Overview

The LC871D00 series is an 8-bit microcomputer that, centered around a CPU running at a minimum bus cycle time of 62.5ns, integrates on a single chip a number of hardware features such as 64K-byte flash ROM (onboard programmable), 4096-byte RAM, an on-chip debugger, a sophisticated 16-bit timer/counter (may be divided into 8-bit timers), 16-bit timer/counter (may be divided into 8-bit timers or 8-bit PWMs), two 8-bit timers with a prescaler, a base timer serving as a time-of-day clock, a high-speed clock counter, two synchronous SIO interfaces (with automatic block transmission/reception function), an asynchronous/synchronous SIO interface, two UART interfaces (full duplex), a full-speed USB interface (with function control features), 12-channel 12-bit A/D converter with 12-/8-bit resolution selector, two 12-bit PWM channels, a system clock frequency divider, an infrared remote control receive function, and a 30-source 10-vector address interrupt function.

1.2 Features

● **Flash ROM**
  - Capable of onboard programming with a wide range of voltage source: 3.0 to 5.5V
  - Block-erasable in 128-byte units
  - Writes data in 2-byte units
  - 65536×8 bits

● **RAM**
  - 4096×9 bits

● **Bus cycle time**
  - 62.5 ns (at 16 MHz)   VDD = 3.0 to 5.5V

  Note: The bus cycle time here refers to the ROM read speed.

● **Minimum instruction cycle time (Tcyc)**
  - 187 ns (at 16 MHz)   VDD = 3.0 to 5.5V

● **Ports**
  - I/O ports

Ports whose I/O direction can be designated in 1-bit units	28 (P10 to P17, P20 to P27, P30 to P34, P70 to P73, PWM0, PWM1, XT2)
Ports whose I/O direction can be designated in 4-bit units	8 (P00 to P07)

  - USB ports   2 (D+, D-)
  - Dedicated oscillator ports   2 (CF1, CF2)
  - Input-only port (also used for oscillation) 1 (XT1)
  - Reset pin   1 ($\overline{\text{RES}}$)
  - Power pins   6 (VSS1 to VSS3, VDD1 to VDD3)

● **Timers**

- Timer 0: 16-bit timer/counter with a capture register

  Mode 0: 8-bit timer with an 8-bit programmable prescaler (with an 8-bit capture register) × 2 channels

  Mode 1: 8-bit timer with an 8-bit programmable prescaler (with an 8-bit capture register) + 8-bit counter (with an 8-bit capture register)

  Mode 2: 16-bit timer with an 8-bit programmable prescaler (with a 16-bit capture register)

  Mode 3: 16-bit counter (with a 16-bit capture register)

- Timer 1: 16-bit timer/counter that supports PWM/toggle output

  Mode 0: 8-bit timer with an 8-bit prescaler (with toggle output) + 8-bit timer/counter with an 8-bit prescaler (with toggle output)

  Mode 1: 8-bit PWM with an 8-bit prescaler × 2 channels

  Mode 2: 16-bit timer/counter with an 8-bit prescaler (with toggle output) (toggle output also possible from the lower-order 8 bits)

  Mode 3: 16-bit timer with an 8-bit prescaler (with toggle output) (The lower-order 8 bits can be used as PWM.)

- Timer 6: 8-bit timer with a 6-bit prescaler (with toggle output)
- Timer 7: 8-bit timer with a 6-bit prescaler (with toggle output)

- Base timer
  1) The clock can be selected from among a subclock (32.768 kHz crystal oscillation), system clock, and timer 0 prescaler output.
  2) Interrupts are programmable in 5 different time schemes.

● **Serial interface**

- SIO0: synchronous serial interface
  1) LSB first/MSB first selectable.
  2) Transfer clock rate: 4/3 to 512/3 Tcyc
  3) Automatic continuous data communication (1 to 256 bits, specifiable in 1-bit units). (Suspension and resumption of data transfer possible in 1-byte units)

- SIO1: 8-bit asynchronous/synchronous serial interface

  Mode 0: Synchronous 8-bit serial I/O (2- or 3-wire configuration, 2 to 512 Tcyc transfer rate)

  Mode 1: Asynchronous serial I/O (half-duplex, 8 data bits, 1 stop bit, 8 to 2048 Tcyc baudrate)

  Mode 2: Bus mode 1 (start bit, 8 data bits, 2 to 512 Tcyc transfer rate)

  Mode 3: Bus mode 2 (start detect, 8 data bits, stop detect)

- SIO4: Synchronous serial interface
  1) LSB first/MSB first mode is selectable
  2) Transfer clock rate: 4/3 to 1020/3 Tcyc
  3) Automatic continuous data communication (1 to 4096 bytes, specifiable in 1-byte units). (suspension and resumption of transfer possible in 1-byte or 1-word units)
  4) Auto-start with falling edge detection
  5) Clock polarity selectable
  6) Built-in CRC16 calculation circuit

● **UART × 2 channels**
1) Full duplex
2) Data length: 7/8/9 bits selectable
3) Stop bits: 1 bit (2 bits in continuous transmission mode)
4) Baudrate generator built in (16/3 to 8192/3 Tcyc baudrate)

● **AD converter: 12 bits × 12 channels**
- 12-/8- bit AD converter resolution selectable

● **PWM: Multifrequency 12-bit PWM × 2 channels**

● **Infrared remote control receiver circuit**
1) Noise rejection function (Noise filter time constant: approx. 120 μs when selecting a 32.768 kHz crystal oscillator as a clock.)
2) Supports PPM (Pulse Position Modulation), Manchester and other coding systems.
3) X'tal HOLD mode release function

● **USB Interface (function control)**
- Compliant with USB 2.0 Full-Speed
- Supports a maximum of 4 user-defined endpoints.

Endpoint		EP0	EP1	EP2	EP3	EP4
Transfer type	Control	○	-	-	-	-
	Bulk	-	○	○	○	○
	Interrupt	-	○	○	○	○
	Isochronous	-	○	○	○	○
Max. payload		64	64	64	64	64

● **Watchdog timer (External RC)**
- Interrupt and reset are selectable.

● **Watchdog timer (Internal counter)**
- Generates an internal reset on overflow occurring in a timer that runs on a dedicated low-speed RC oscillator clock (30kHz).
- The operating mode that continues counting, stops counting, or retains count can be selected when HALT or HOLD mode is entered.

● **Interrupts**
- 30 sources, 10 vector addresses
1) It provides three levels (low (L), high (H), and highest (X)) of multiplex interrupt control. Any interrupt request of the level equal to or lower than the current interrupt is not accepted.
2) When interrupt requests to two or more vector addresses occur at the same time, the interrupt of the highest level takes precedence over the other interrupts. For interrupts of the same level, the interrupt into the smallest vector address is given priority

No.	Vector	Level	Interrupt source
1	00003H	X or L	INT0
2	0000BH	X or L	INT1
3	00013H	H or L	INT2/T0L/INT4/USB bus active/Remote control receive
4	0001BH	H or L	INT3/INT5/Base timer
5	00023H	H or L	T0H
6	0002BH	H or L	TIL/TIH
7	00033H	H or L	SIO0/USB bus reset/USB suspend/ UART1 receive/UART2 receive
8	0003BH	H or L	SIO1/USB endpoint/USB-SOF/SIO4/ UART1 transmit/UART2 transmit
9	00043H	H or L	ADC/T6/T7
10	0004BH	H or L	Port 0/PWM0/PWM1

• Priority levels: X > H > L

• When interrupts of the same level occur at the same time, an interrupt with a smaller vector address is given priority.

● **Subroutine stack levels: 2048 (max.) (the stack is allocated in RAM)**

● **High-speed multiplication/division instructions**
   • 16 bits × 8 bits   (5 Tcyc execution time)
   • 24 bits × 16 bits   (12 Tcyc execution time)
   • 16 bits ÷ 8 bits   (8 Tcyc execution time)
   • 24 bits ÷ 16 bits   (12 Tcyc execution time)

● **Oscillator circuits and PLL**
   • RC oscillator circuit (internal):   For system clock   1MHz (typ)
   • Low-speed RC oscillator circuit (internal):   For watchdog timer   30kHz (typ)
   • CF oscillator circuit: For system clock
   • Crystal oscillator circuit: For system clock, time-of-day clock
   • PLL circuit (internal):   For USB interface

● **Standby function**
- HALT mode: Halts instruction execution while allowing the peripheral circuits to continue operation.
    1) Oscillation is not halted automatically.
    2) There are three ways of releasing HALT mode.
        (1) Setting the reset pin to a low level.
        (2) Generating a reset by the watchdog timer
        (3) Establishing an interrupt

- HOLD mode: Suspends instruction execution and the operation of the peripheral circuits.
    1) The PLL base clock generator, CF, RC and crystal oscillators automatically stop operation.
        Note: The oscillation of the low-speed RC oscillator circuit is also controlled directly by the watchdog timer and oscillation in standby mode time is also controlled.
    2) There are five ways of releasing HOLD mode.
        <1> Setting the reset pin to the low level.
        <2> Generating a reset by the watchdog timer
        <3> Establishing an interrupt source at either INT0, INT1, INT2, INT4, and INT5 pins
            * INT0 and INT1 HOLD mode release is available only when level detection is set.
        <4> Establishing an interrupt source at Port 0
        <5> Establishing a bus active interrupt source in the USB interface circuit

- X'tal HOLD mode: Suspends instruction execution and the operation of the peripheral circuits except the base timer and the infrared remote control receiver.
    1) The PLL base clock generator, CF and RC oscillators automatically stop operation.
        Note: The oscillation of the low-speed RC oscillator circuit is also controlled directly by the watchdog timer and oscillation in standby mode time is also controlled.
    2) The state of crystal oscillation established when X'tal HOLD mode is entered is retained.
    3) There are seven ways of releasing X'tal HOLD mode.
        <1> Setting the reset pin to the low level.
        <2> Generating a reset by the watchdog timer
        <3> Establishing an interrupt source at either INT0, INT1, INT2, INT4, and INT5 pins
            * INT0 and INT1 X'tal HOLD mode release is available only when level detection is set.
        <4> Establishing an interrupt source at Port 0
        <5> Establishing an interrupt source in the base timer circuit
        <6> Establishing a bus active interrupt source in the USB interface circuit
        <7> Establishing an interrupt source in the infrared remote control receiver circuit

● **Package form**
- TQFP48J (7×7)　(Lead-free and halogen-free specification)

● **Development tools**
- On-chip debugger: TCB87 Type B + LC87F1D64A

1.3   Pinout
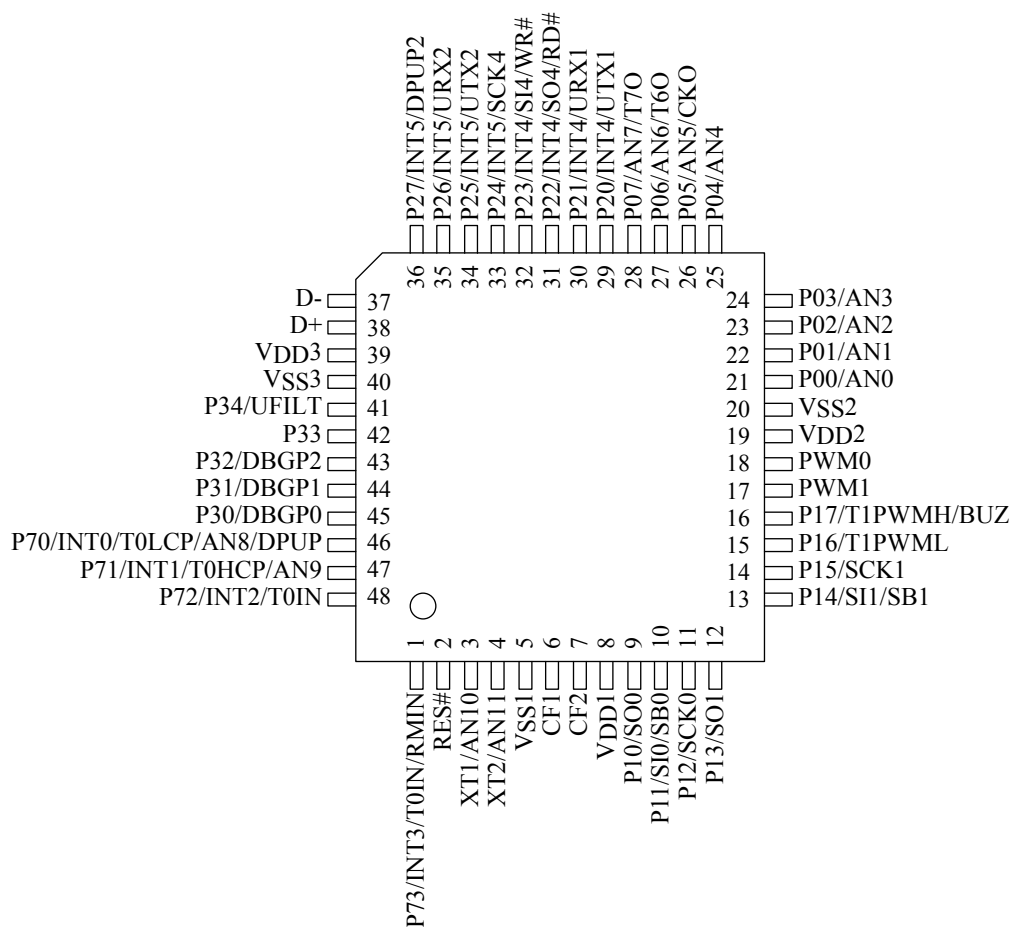
SANYO: TQFP48J (7 x 7)

TQFP48	NAME	TQFP48	NAME
1	P73/INT3/T0IN/RMIN	25	P04/AN4
2	RES#	26	P05/AN5/CKO
3	XT1/AN10	27	P06/AN6/T6O
4	XT2/AN11	28	P07/AN7/T7O
5	VSS1	29	P20/INT4/UTX1
6	CF1	30	P21/INT4/URX1
7	CF2	31	P22/INT4/SO4/RD#
8	VDD1	32	P23/INT4/SI4/WR#
9	P10/SO0	33	P24/INT5/SCK4
10	P11/SI0/SB0	34	P25/INT5/UTX2
11	P12/SCK0	35	P26/INT5/URX2
12	P13/SO1	36	P27/INT5/DPUP2
13	P14/SI1/SB1	37	D-
14	P15/SCK1	38	D+
15	P16/T1PWML	39	VDD3
16	P17/T1PWMH/BUZ	40	VSS3
17	PWM1	41	P34/UFILT
18	PWM0	42	P33
19	VDD2	43	P32/DBGP2
20	VSS2	44	P31/DBGP1
21	P00/AN0	45	P30/DBGP0
22	P01/AN1	46	P70/INT0/T0LCP/AN8/DPU
23	P02/AN2	47	P71/INT1/T0HCP/AN9
24	P03/AN3	48	P72/INT2/T0IN

1.4 System Block Diagram
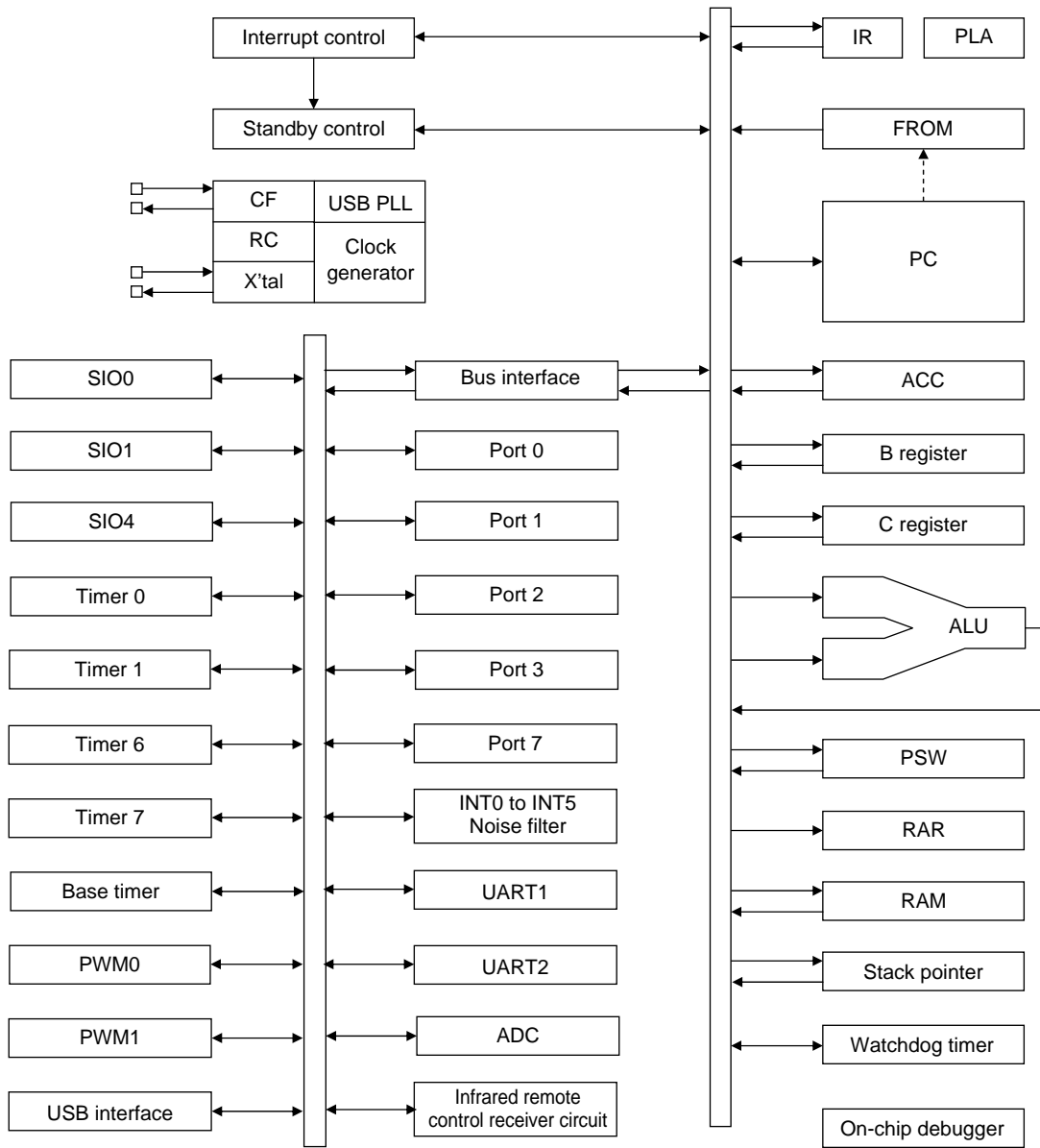
1.5   Pin Functions

Pin	I/O	Description	Option
VSS1, VSS2, VSS3	—	– power supply pin	No
VDD1, VDD2	—	+ power supply pin	No
VDD3	—	USB reference voltage pin	Yes
Port 0 P00 to P07	I/O	• 8-bit I/O port • I/O specifiable in 4-bit units • Pull-up resistors can be turned on and off in 4-bit units • HOLD release input • Port 0 interrupt input • Pin functions 　　AD converter input port: AN0 to AN7 (P00 to P07) 　　P05: System clock output 　　P06: Timer 6 toggle output 　　P07: Timer 7 toggle output	Yes
Port 1 P10 to P17	I/O	• 8-bit I/O port • I/O specifiable in 1-bit units • Pull-up resistors can be turned on and off in 1-bit units • Pin functions 　　P10: SIO0 data output 　　P11: SIO0 data input/bus I/O 　　P12: SIO0 clock I/O 　　P13: SIO1 data output 　　P14: SIO1 data input/bus I/O 　　P15: SIO1 clock I/O 　　P16: Timer 1 PWML output 　　P17: Timer 1 PWMH output/buzzer output	Yes
Port 2 P20 to P27	I/O	• 8-bit I/O port • I/O specifiable in 1-bit units • Pull-up resistors can be turned on and off in 1-bit units. • Pin functions 　　P20 to P23: INT4 I/O/ HOLD release input/timer 1 event input/ 　　　　　　timer 0L capture input/timer 0H capture input 　　P24 to P27: INT5 input/HOLD release input/timer 1 event input/timer 　　　　　　0L capture input/ timer 0H capture input 　　P20: UART1 transmit 　　P21: UART1 receive 　　P22: SIO4 data I/O/parallel interface RD# output 　　P23: SIO4 data I/O/parallel interface WR# output 　　P24: SIO4 clock I/O 　　P25: UART2 transmit 　　P26: UART2 receive 　　P27: D+ 1.5kΩ pull-up resistor connection pin Interrupt acknowledge type	Yes

	Rising	Falling	Rising & Falling	H level	L level
INT4	○	○	○	×	×
INT5	○	○	○	×	×

Pin	I/O	Description	Option

Port 3
 	
P30 to P34
	I/O	• 5-bit I/O port
		• I/O specifiable in 1-bit units
		• Pull-up resistors can be turned on and off in 1-bit units.
		• Pin functions
		    P34: USB interface PLL filter circuit connection pin
		    On-chip debugger pins: DBGP0 to DBGP2 (P30 to P32)
			Yes

Port 7
 	
P70 to P73
	I/O	• 4-bit I/O port
		• I/O specifiable in 1-bit units
		• Pull-up resistors can be turned on and off in 1-bit units.
		• Pin functions
		    P70: INT0 input/HOLD release input/timer 0L capture input/
		       watchdog timer output/D+ 1.5kΩ pull-up resistor connection pin
		    P71: INT1 input/HOLD release input/timer 0 capture input
		    P72: INT2 input/HOLD release input/timer 0 event input/timer 0L
		       capture input/high-speed clock counter input
		    P73: INT3 input (with noise filter)/timer 0 event input/timer 0H
		       capture input/infrared remote control receiver input
		    AD converter input port: AN8 (P70), AN9 (P71)
		   Interrupt acknowledge type
		
		<table><tr><th></th><th>Rising</th><th>Falling</th><th>Rising & Falling</th><th>H level</th><th>L level</th></tr><tr><td>INT0</td><td>○</td><td>○</td><td>×</td><td>○</td><td>○</td></tr><tr><td>INT1</td><td>○</td><td>○</td><td>×</td><td>○</td><td>○</td></tr><tr><td>INT2</td><td>○</td><td>○</td><td>○</td><td>×</td><td>×</td></tr><tr><td>INT3</td><td>○</td><td>○</td><td>○</td><td>×</td><td>×</td></tr></table>
			No

PWM0 PWM1	I/O	• PWM0 and PWM1 output port • General-purpose input port	No
D-	I/O	• USB data I/O pin D-/general-purpose I/O port	No
D+	I/O	• USB data I/O pin D+/general-purpose I/O port	No
$\overline{\text{RES}}$	I	Reset pin	No

XT1	I	• 32.768kHz crystal resonator input pin
		• Pin functions
		  General-purpose input port
		  AD converter input port: AN10
		  Must be connected to VDD1 if not to be used.
			No

XT2	I/O	• 32.768kHz crystal resonator output pin
		• Pin functions
		  General-purpose I/O port
		  AD converter input port: AN11
		  Must be set for oscillation and kept open if not to be used.
			No

CF1	I	Ceramic resonator input pin	No
CF2	O	Ceramic resonator output pin	No

1.6   On-chip Debugger Pin Connection Requirements

For the treatment of the on-chip debugger pins, refer to the separately available documents entitled "RD87 On-chip Debugger Installation Manual".

1.7   Recommended Unused Pin Connections

Pin Name	Recommended Unused Pin Connections	
	Board	**Software**
P00 to P07	Open	Output low
P10 to P17	Open	Output low
P20 to P27	Open	Output low
P31 to P34	Open	Output low
P70 to P73	Open	Output low
PWM0, PWM1	Open	Output low
D+, D–	Open	Output low
XT1	Pulled down with a 100kΩ resistor or less	General-purpose input port
XT2	Open	Output low

1.8 Port Output Types

The table below lists the types of port outputs and the presence/absence of a pull-up resistor.

Data can be read into any input port even if it is in output mode.

Port	Options Selected in Units of	Option Type	Output Type	Pull-up Resistor
P00 to P07	1 bit	1	CMOS	Programmable (Note 1)
		2	N-channel open drain	No
P10 to P17	1 bit	1	CMOS	Programmable
P20 to P27		2	N-channel open drain	Programmable
P30 to P34				
P70	–	No	N-channel open drain	Programmable
P71 to P73	–	No	CMOS	Programmable
PWM0, PWM1	–	No	CMOS	No
D+, D-	–	No	CMOS	No
XT1	–	No	Input only	No
XT2	–	No	32.768 kHz crystal resonator output (N-channel open drain when selected as general-purpose output port)	No

Note 1: Programmable pull-up resistors for port 0 are controlled in 4-bit units (P00 to P03, P04 to P07).

1.9   User Option Table

Option Name	Option to be Applied on	Mask-ROM Version*1	Flash-ROM Version	Option Selected in Units of	Option Selection
Port output type	P00 to P07	○	○	1 bit	CMOS
					N-channel open drain
	P10 to P17	○	○	1 bit	CMOS
					N-channel open drain
	P20 to P21	○	○	1 bit	CMOS
					N-channel open drain
	P30 to P34	○	○	1 bit	CMOS
					N-channel open drain
Program start address	-	× *2	○	-	00000h
					0FE00h
USB regulator	USB regulator	○	○	-	Use
					Nonuse
	USB regulator (HOLD mode)	○	○		Use
					Nonuse
	USB regulator (HOLD mode)	○	○		Use
					Nonuse

*1 Since the options are selected as mask options, no change can be made after masking is completed.

*2 The program start address of the mask version of the microcontroller is set to 00000h.

1.10   USB Reference Power Option

When a voltage 4.5 to 5.5V is supplied to VDD1 and the internal USB reference voltage circuit is activated, the reference voltage for USB port output is generated.   The active/inactive state of reference voltage circuit can be switched by the option selection.

The option selection must be made according to the voltage supplied to VDD1 as described below.

VDD1 voltage (V)		4.5 to 5.5			3.0 to 3.6
Option selection	USB Regulator	Use	Use	Use	Nonuse
	USB Regulator at HOLD mode	Use	Nonuse	Nonuse	Nonuse
	USB Regulator at HALT mode	Use	Nonuse	Use	Nonuse
Reference voltage circuit state	Normal state	Active	Active	Active	Inactive
	HOLD mode	Active	Inactive	Inactive	Inactive
	HALT mode	Active	Inactive	Active	Inactive
		(1)	(2)	(3)	(4)

When the USB reference voltage circuit is made inactive, the level of the reference voltage for USB port output is equal to VDD1.

Selection (2) or (3) can be used to set the reference voltage circuit inactive in HOLD or HALT mode.

When the reference voltage circuit is activated, the current drain increases by approximately 100μA compared with that when the reference voltage circuit is inactive.

2. Internal Configuration

2.1 Memory Space

LC870000 series microcontrollers have the following three types of memory space:

1) Program memory space: 256K bytes (128K bytes × 2 banks)
2) Internal data memory space: 64K bytes (0000H to FDFFH out of 0000H to FFFFH is shared with the stack area.)
3) External data memory space: 16M bytes
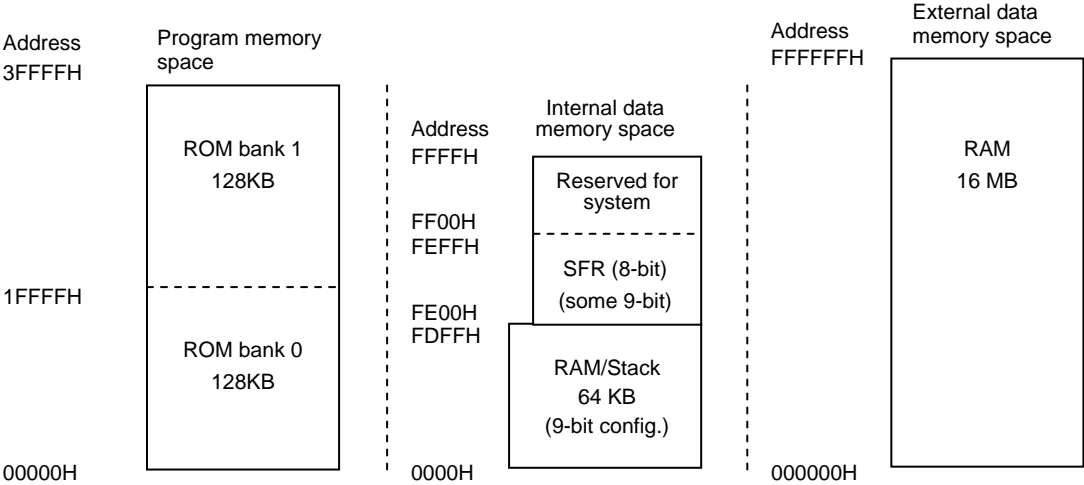
Note: SFR is the area in which special function registers such as the accumulator are allocated (see Appendixes A-I).

Figure 2.1.1   Types of Memory Space

2.2 Program Counter (PC)

The program counter (PC) is made up of 17 bits and a bank flag BNK. The value of BNK determines the bank. The lower-order 17 bits of the PC allows linear access to the 128K ROM space in the current bank.

Normally, the PC advances automatically in the current bank on each execution of an instruction. Bank switching is accomplished by executing a Return instruction after pushing necessary addresses onto the stack. When executing a branch or subroutine instruction, when accepting an interrupt, or when a reset is generated, the value corresponding to each operation is loaded into the PC.

Table 2.2.1 lists the values that are loaded into the PC when the respective operations are performed.

Table 2.2.1   Values Loaded in the PC

Operation			PC value	BNK value
Inter-rupt	Reset (Note)		00000H	0
			0FE00H	0
	INT0		00003H	0
	INT1		0000BH	0
	INT2/T0L/INT4/USB bus active/Remote control receive		00013H	0
	INT3/INT5/Base timer		0001BH	0
	T0H		00023H	0
	T1L/T1H		0002BH	0
	SIO0/USB bus reset/USB suspend/UART1 receive/ UART2 receive		00033H	0
	SIO1/USB end point/USB-SOF/SIO4/UART1 transmit/ UART2 transmit		0003BH	0
	ADC/T6/T7		00043H	0
	Port 0/PWM0/ PWM1		0004BH	0
Unconditional branch instructions	JUMP a17		PC=a17	Unchanged
	BR r12		PC=PC+2+r12[-2048 to +2047]	Unchanged
Conditional branch instructions	BE, BNE, DBNZ, DBZ, BZ, BNZ, BZW, BNZW, BP, BN, BPC		PC=PC+nb+r8[-128 to +127] nb: Number of instruction bytes	Unchanged
Call instructions	CALL a17		PC=a17	Unchanged
	RCALL r12		PC=PC+2+r12[-2048 to +2047]	Unchanged
	RCALLA		PC=PC+1+Areg[0 to +255]	Unchanged
Return instructions	RET, RETI		PC16 to 08=(SP) PC07 to 00=(SP-1) (SP) denotes the contents of RAM address designated by the value of the stack pointer SP.	BNK is set to bit 8 of (SP-1).
Standard instructions	NOP, MOV, ADD, …		PC=PC+nb nb: Number of instruction bytes	Unchanged

Note: The reset-time program start address can be selected through a user option in the flash version of the microcontroller. In the mask version, the program start address is fixed at address 00000H.

2.3   Program Memory (ROM)

This series of microcontrollers has a program memory space of 256K bytes, but the size of the ROM that is actually incorporated in the microcontroller varies with the type of microcontroller. The ROM table look-up instruction (LDC) can be used to reference all ROM data within the bank. Of the ROM space, the 256 bytes in ROM bank 0 (1FF00H-1FFFFH for ROM size of 64K and above and 0FF00H-0FFFFH for ROM size of 64K and below) is reserved as the option area. Consequently, this area is not available as a program area.

2.4   Internal Data Memory (RAM)

LC870000 series microcontrollers have an internal data memory space of 64K bytes, but the size of the RAM that is actually incorporated in the microcontroller varies with the type of the microcontroller. Nine bits are used to access addresses 0000H to FDFFH of the 128K ROM space and 8 or 9 bits are used to access addresses FE00H to FFFFH. The 9th bit of RAM is implemented by bit 1 of the PSW and can be read and written.

The 128 bytes of RAM from 0000H to 007FH are paired to form 64 2-byte indirect address registers. The bit length of these indirect registers is normally 16 bits (8 bits × 2). When they are used by the ROM table look-up instruction (LDC), however, their bit length is set to 17 bits (9 higher-order bits + 8 lower-order bits).

As shown in Figure 2.4.1, the available instructions vary depending on the RAM address.

The efficiency of the ROM used and a higher execution speed can be attempted using these instructions properly.
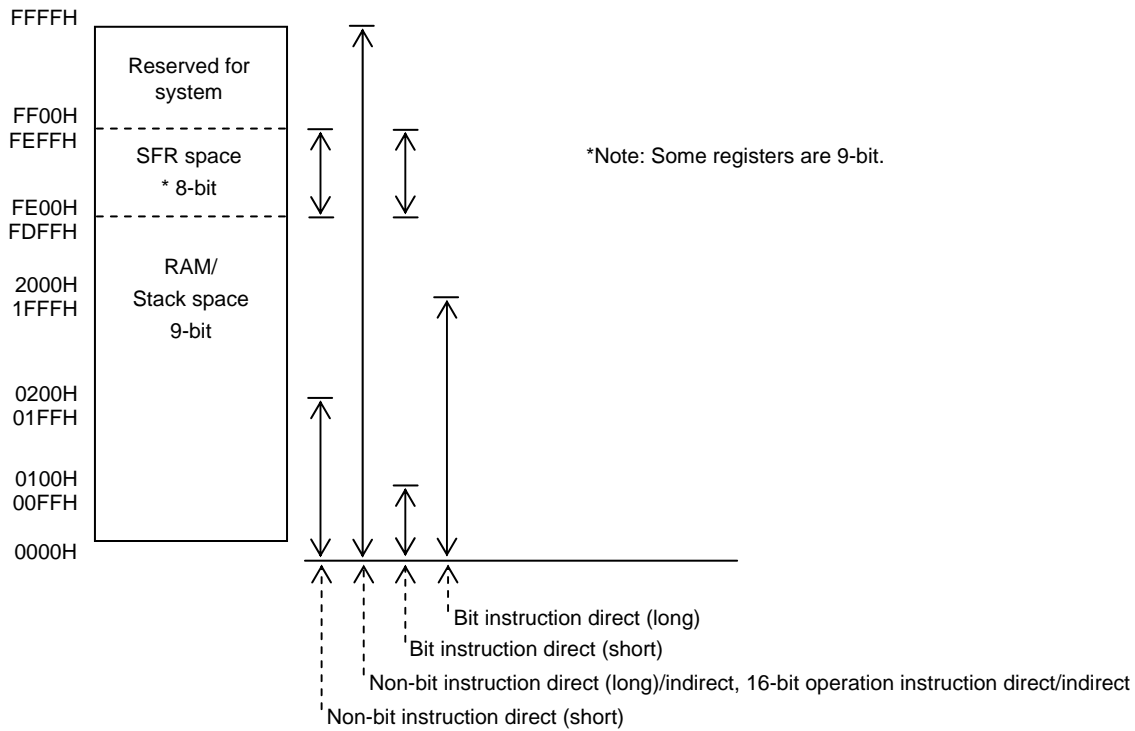
Figure 2.4.1 RAM Addressing Map

When the value of the PC is stored in RAM during the execution of a subroutine call instruction or interrupt, assuming that SP represents the current value of the stack pointer, the value of BNK and the lower-order 8 bits of the (17-bit) PC are stored in RAM address SP+1 and the higher-order 9 bits in SP+2, after which SP is set to SP+2.

2.5　Accumulator/A Register (ACC/A)

The accumulator (ACC), also called the A register, is an 8-bit register that is used for data computation, transfer, and I/O processing. It is allocated to address FE00H in the internal data memory space and initialized to 00H when a reset is performed.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE00	0000 0000	R/W	AREG	AREG7	AREG6	AREG5	AREG4	AREG3	AREG2	AREG1	AREG0

2.6　B Register (B)

The B register is combined with the ACC to form a 16-bit arithmetic register during the execution of a 16-bit arithmetic instruction. During a multiplication or division instruction, the B register is used with the ACC and C register to store the results of computation. In addition, during an external memory access instruction (LDX or STX), the B register designates the higher-order 8 bits of the 24-bit address.

The B register is allocated to address FE01H of the internal data memory space and initialized to 00H when a reset is performed.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE01	0000 0000	R/W	BREG	BREG7	BREG6	BREG5	BREG4	BREG3	BREG2	BREG1	BREG0

2.7 C Register (C)

The C register is used with the ACC and B register to store the results of computation during the execution of a multiplication or division instruction. In addition, during a C register offset indirect instruction, the C register stores the offset data (-128 to +127) to the contents of an indirect register.

The C register is allocated to address FE02H of the internal data memory space and initialized to 00H when a reset is performed.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE02	0000 0000	R/W	CREG	CREG7	CREG6	CREG5	CREG4	CREG3	CREG2	CREG1	CREG0

2.8 Program Status Word (PSW)

The program status word (PSW) is made up of flags that indicate the status of computation results, a flag to access the 9th bit of RAM, and a flag to designate the bank during the LDCW instruction. The PSW is allocated to address FE06H of the internal data memory space and initialized to 00H when a reset is performed.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE06	0000 0000	R/W	PSW	CY	AC	PSWB5	PSWB4	LDCBNK	OV	P1	PARITY

CY (bit 7): Carry flag

CY is set (to 1) when a carry occurs as the result of a computation and cleared (to 0) when no carry occurs. There are the following four types of carries:

1) Carry resulting from an addition
2) Borrow resulting from a subtraction
3) Borrow resulting from a comparison
4) Carry resulting from a rotation

There are some instructions that do not affect this flag at all.

AC (bit 6): Auxiliary carry flag

AC is set (to 1) when a carry or borrow occurs in bit 3 (bit 3 of the higher-order byte during a 16-bit computation) as the result of an addition or subtraction and cleared (to 0) otherwise.

There are some instructions that do not affect this flag at all.

PSWB5, PSWB4 (bits 5 and 4): User bits

These bits can be read and written through instructions. They can be used by the user freely.

LDCBNK (bit 3): Bank flag for the table look-up instruction (LDCW)

This bit designates the ROM bank to be specified when reading the program ROM with a table look-up instruction.

(0: ROM-ADR = 0 to 1FFFF, 1: ROM-ADR = 20000 to 3FFFF)

OV (bit 2): Overflow flag

OV is set (to 1) when an overflow occurs as the result of an arithmetic operation and cleared (to 0) otherwise. An overflow occurs in the following cases:

1) When MSB is used as the sign bit and when the result of negative number + negative number or negative number – positive number is a positive.
2) When MSB is used as the sign bit and when the result of positive number + positive number or positive number – negative number is a negative number.

3) When the higher-order 8 bits of a 16 bits × 8 bits multiplication is nonzero

4) When the higher-order 16 bits of a 24 bits × 16 bits multiplication is nonzero

5) When the divisor of a division is 0.

There are some instructions that do not affect this flag at all.

P1 (bit 1): RAM bit 8 data flag

P1 is used to manipulate bit 8 of 9-bit internal data RAM (0000H to FDFFH). Its behavior varies depending on the instruction executed. See Table 2.4.1 for details.

PARITY (bit 0): Parity flag

This bit shows the parity of the accumulator (A register). The parity flag is set (to 1) when there is an odd number of 1s in the A register. It is cleared (to 0) when there is an even number of 1s in the A register.

2.9   Stack Pointer (SP)

LC870000 series microcontrollers can use RAM addresses 0000H to FDFFH as a stack area. The size of RAM, however, varies depending on the microcontroller type. The SP is 16 bits long and made up of two registers: SPL (at address FE0A) and SPH (at address FE0B). It is initialized to 0000H when a reset is performed.

The SP is incremented by 1 before data is saved in stack memory and decremented by 1 after the data is restored from stack memory.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE0A	0000 0000	R/W	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
FE0B	0000 0000	R/W	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8

The value of the SP changes as follows:

1) When the PUSH instruction is executed: SP = SP + 1, RAM (SP) = DATA

2) When the CALL instruction is executed: SP = SP + 1, RAM (SP) = ROMBANK + ADL

   SP = SP + 1, RAM (SP) = ADH

3) When the POP instruction is executed: DATA = RAM (SP), SP = SP - 1

4) When the RET instruction is executed: ADH = RAM (SP), SP = SP - 1

   ROMBANK + ADL = RAM(SP), SP = SP - 1

2.10 Indirect Addressing Registers

LC870000 series microcontrollers are provided with three addressing schemes ([Rn], [Rn+C], [off]), which use the contents of indirect registers (indirect addressing modes). (See Section 2.11 for the addressing modes.) These addressing modes use 64 2-byte indirect registers (R0 to R63) allocated to RAM addresses 0 to 7EH. The indirect registers can also be used as general-purpose registers (e.g., for saving 2-byte data). Naturally, these addresses can be used as ordinary RAM (on a 1-byte (9 bits) basis) if they are not used as indirect registers. R0 to R63 are "system reserved words" to the assembler and need not be defined by the user.

	RAM	Reserved for system

Address

7FH	R63 (Upper)	
7EH	R63 (Lower)	R63 = 7EH
·	·	·
·	·	·
03H	R1 (Upper)	
02H	R1 (Lower)	R1 = 2
01H	R0 (Upper)	
00H	R0 (Lower)	R0 = 0

Figure 2.10.1   Allocation of Indirect Registers

2.11 Addressing Modes

LC870000 series microcontrollers support the following seven addressing modes:

1)   Immediate (immediate data refers to data whose value has been established at program preparation (assembly) time.)
2)   Indirect register (Rn) indirect ($0 \le n \le 63$)
3)   Indirect register (Rn) + C register indirect ($0 \le n \le 63$)
4)   Indirect register (R0) + Offset value indirect
5)   Direct
6)   ROM table look-up
7)   External data memory access

The rest of this section describes these addressing modes.

2.11.1   Immediate Addressing (#)

The immediate addressing mode allows 8-bit (1-byte) or 16-bit (1-word) immediate data to be handled. Examples are given below.

Examples:

```
        LD    #12H;      Loads the accumulator with byte data (12H).
  L1:   LDW   #1234H;    Loads the BA register pair with word data (1234H).
        PUSH  #34H;      Loads the stack with byte data (34H).
        ADD   #56H;      Adds byte data (56H) to the accumulator.
        BE    #78H, L1;  Compares byte data (78H) with the accumulator for a branch.
```

2.11.2   Indirect Register Indirect Addressing ([Rn])

In indirect register indirect addressing mode, it is possible to select one of the indirect registers (R0 to R63) and use its contents to designate an address in RAM or SFR. When the selected register contains, for example, "FE02H," it designates the C register.

Example: When R3 contains "123H" (RAM address 6: 23H, RAM address 7: 01H)

	LD	[R3];	Transfers the contents of RAM address 123H to the accumulator.
L1:	STW	[R3];	Transfers the contents of BA register pair to RAM address 123H.
	PUSH	[R3];	Saves the contents of RAM address123H in the stack.
	SUB	[R3];	Subtracts the contents of RAM address 123H from the accumulator.
	DBZ	[R3], L1;	Decrements the contents of RAM address 123H by 1 and causes a branch if zero.

2.11.3   Indirect Register + C Register Indirect Addressing ([Rn, C])

In indirect register + C register indirect addressing mode, the result of adding the contents of one of the indirect registers (R0 to R63) to the contents of the C register (-128 to +127 with MSB being the sign bit) designates an address in RAM or SFR. For example, if the selected indirect register contains "FE02H" and the C register contains "FFH (-1)," the address "B register (FE02H + (-1) = FE01H" is designated.

Examples: When R3 contains "123H" and the C register contains "02H"

	LD	[R3, C];	Transfers the contents of RAM address 125H to the accumulator.
L1:	STW	[R3, C];	Transfers the contents of the BA register pair to RAM address 125H.
	PUSH	[R3, C];	Saves the contents of RAM address 125H in the stack.
	SUB	[R3, C];	Subtracts the contents of RAM address 125H from the accumulator.
	DBZ	[R3, C], L1;	Decrements the contents of RAM address 125H by 1 and causes a branch if zero.

<Notes on this addressing mode >

The internal data memory space is divided into three closed functional areas as explained in Section 2.1, namely, 1) system reserved area (FF00H to FFFFH), 2) SFR area (FE00H to FEFFH), and 3) RAM/stack area (0000H to FDFFH). Consequently, it is not possible to point to a different area using the value of the C register from the basic area designated by the contents of Rn. For example, if the instruction "LD [R5,C]" is executed when R5 contains "0FDFFH" and the C register contains "1," since the basic area is 3) RAM/stack area (0000H to FDFFH), the intended address "0FDFFH+1 = 0FE00H" lies outside the basic area and "0FFH" is placed in the ACC as the result of LD. If the instruction "LD [R5,C]" is executed when R5 contains "0FEFFH" and the C register contains "2," since the basic area is 2) SFR area (FE00H to FEFFH), the intended address "0FEFFH+2 = 0FF01H" lies outside the basic area. In this case, since SFR is confined in an 8-bit address space, the part of the address data addressing outside the 8-bit address space is ignored and the contents of 0FE01H (B register) are placed in the ACC as the result of the computation "0FF01H&0FFH+0FE00H = 0FE01H."

2.11.4 Indirect Register (R0) + Offset Value Indirect Addressing ([off])

In this addressing mode, the result of adding the 7-bit signed offset data off (-64 to + 63) to the contents of the indirect register R0 designate an address in RAM or SFR. If R0 contains "FE02H" and off has a value of "7EH(-2)," for example, the A register (FE02H+(-2) = FE00H) is designated.

Examples: When R0 contains "123H" (RAM address 0: 23H, RAM address 1: 01H)

	LD	[10H];	Transfers the contents of RAM address 133H to the accumulator.
L1:	STW	[10H];	Transfers the contents of the BA register pair to RAM address 133H.
	PUSH	[10H];	Saves the contents of RAM address 133H in the stack.
	SUB	[10H];	Subtracts the contents of RAM address 133H from the accumulator.
	DBZ	[10H], L1;	Decrements the contents of RAM address 133H by 1 and causes a branch if zero.

<Notes on this addressing mode>

The internal data memory space is divided into three closed closed functional areas as explained in Section 2.1, namely, 1) system reserved area (FF00H to FFFFH), 2) SFR area (FE00H to FEFFH), and 3) RAM/stack area (0000H to FDFFH). Consequently, it is not possible to point to a different area using an offset value from the basic area designated by the contents of R0. For example, if the instruction "LD [1]" is executed when R0 contains "0FDFFH," since the basic area is 3) RAM/stack area (0000H to FDFFH), the intended address "0FDFFH+1 = 0FE00H" lies outside the basic area and "0FFH" is placed in the ACC as the results of LD. If the instruction "LD [2]" is executed when R0 contains "0FEFFH," since the basic area is 2) SFR area (FE00H to FEFFH), the intended address "0FEFFH+2 = 0FF01H" lies outside the basic area. In this case, since SFR is confined in an 8-bit address space, the part of the address data addressing outside the 8-bit address space is ignored and the contents of "0FE01H (B register) are placed in the ACC as the result of computation "0FF01H&0FFH+0FE00H = 0FE01H."

2.11.5 Direct Addressing (dst)

Direct addressing mode allows a RAM or SFR address to be specified directly in an operand. In this addressing mode, the assembler automatically generates the optimum instruction code from the address specified in the operand (the number of instruction bytes varies according to the address specified in the operand). Long (middle) range instructions (identified by an "L (M)" at the end of the mnemonic) are available to make the byte count of instructions constant (align instructions with the longest one).

Examples:

	LD	123H;	Transfers the contents of RAM address 123H to the accumulator (2-byte instruction).
	LDL	123H;	Transfers the contents of RAM address 123H to the accumulator (3-byte instruction).
L1:	STW	123H;	Transfers the contents of the BA register pair to RAM address 123H.
	PUSH	123H;	Saves the contents of RAM address 123H in the stack.
	SUB	123H;	Subtracts the contents of RAM address 123H from the accumulator.
	DBZ	123H, L1;	Decrements the contents of RAM address 123H by 1 and causes a branch if zero.

2.11.6 ROM Table Look-up Addressing

LC870000 series microcontrollers can read 2-byte data into the BA register pair at once using the LDCW instruction. Three addressing modes ([Rn], [Rn, C], and [off]) are available for this purpose. (In this case only, Rn is configured as 17-bit registers (128K-byte space)).

For models with banked ROM, it is possible to reference the ROM data in the ROM bank (128K bytes) identified by the LDCBNK flag (bit 3) in the PSW. Consequently, when looking into the ROM table on a series model with banked ROM, execute the LDCW instruction after switching the bank using the SET1 or CLR1 instruction so that the LDCBNK flag designates the ROM bank where the ROM table resides.

Examples:

TBL: DB	34H	
DB	12H	
DW	5678H	
•	•	
•	•	
LDW	#TBL;	Loads the BA register pair with the TBL address.
CHGP3	(TBL >> 17) & 1;	Loads LDCBNK in PSW with bit 17 of the TBL address. *(Note 1)*
CHGP1	(TBL >> 16) & 1;	Loads P1 in PSW with bit 16 of the TBL address.
STW	R0;	Loads indirect register R0 with the TBL address (bits 16 to 0).
LDCW	[1];	Reads the ROM table (B=78H, ACC=12H).
MOV	#1, C;	Loads the C register with "01H."
LDCW	[R0, C];	Reads the ROM table (B=78H, ACC=12H).
INC	C;	Increments the C register by 1.
LDCW	[R0, C]:	Reads the ROM table (B=56H, ACC=78H).

Note 1: LDCBNK (bit 3) of PSW needs to be set up only for models with banked ROM.

2.11.7 External Data Memory Addressing

LC870000 series microcontrollers can access external data memory spaces of up to 16M bytes (24 bits) using the LDX and STX instructions. To designate a 24-bit space, specify the contents of the B register (8 bits) as the highest-order byte of the address and the contents (16 bits) of either (Rn), (Rn) + (C), or (R0) + off as the lower-order bytes of the address.

Examples:

LDW	#3456H;	Sets up the lower-order 16 bits.
STW	R0;	Loads the indirect register R0 with the lower-order 16 bits of the address.
MOV	#12H, B;	Sets up the higher-order 8 bits of the address.
LDX	[1];	Transfers the contents of external data memory (address 123457H) to the accumulator.

2.12 Wait Sequence

2.12.1   Wait Sequence Occurrence

This series of microcontrollers performs wait sequences that suspend the execution of instructions in the following cases:

1) When continuous data transfer is performed over the SIO0 with SI0CIR (SCON0, bit 4) set, a wait request is generated ahead of each transfer of 8-bit data, in which case a 1-cycle wait sequence (RAM data transfer) is performed.

2) When continuous data transfer is performed over the SIO4, a wait request is generated for each transfer of 8-bit data, in which case a 1-cycle wait sequence (RAM data transfer) is performed.

3) When data packet transmission/reception is performed by the USB interface circuit, a wait request is generated for each transfer of 4 bytes, in which case a 1-cycle wait sequence (RAM data transfer) is performed.

2.12.2   What is a Wait Sequence?

1) When a wait request occurs by the event explained in Subsection 2.12.1, the CPU suspends the execution of the instruction for one cycle, during which the required data is transferred. This is called a wait sequence.

2) The peripheral circuits such as timers and PWM continue processing during the wait sequence.

3) A wait sequence extends over no more than two cycles.

4) The microcontroller does not perform a wait sequence when it is in HALT or HOLD mode.

5) Note that one cycle of discrepancy is introduced between the progress of the program counter and time once a wait sequence occurs.

Table 2.4.1   Chart of State Transitions of Bit 8 (RAM / SFR) and P1

Instruction	BIT8 (RAM/SFR)	P1 (PSW BIT 1)	Remarks
LD#/LDW#	—	—	
LD	—	P1←REG8	
LDW	—	P1←REGH8	
ST	REG8←P1	—	
STW	REGL8, REGH8←P1	—	
MOV	REG8←P1	—	
PUSH#	RAM8←P1	—	
PUSH	RAM8←REG8	P1←REG8	
PUSHW	RAMH8←REGH8, RAML8←REGL8	P1←REGH8	
PUSH_P	RAM8←Pl	—	
PUSH_BA	RAMH8←P1, RAML8←P1	—	
POP	REG8←RAM8	P1←RAM8	P1←bit1 when PSW is popped
POPW	REGH8←RAMH8, REGL8←RAML8	Pl←RAMH8	P1←bit1 when higher-order address of PSW is popped
POP_P	—	P1←RAMl (bit l)	BIT8 ignored
POP_BA	—	P1←RAMH8	
XCH	REG8↔P1	Same as left.	
XCHW	REGH8←P1, REGL8←Pl, P1←REGH8	Same as left.	
INC	INC 9 bits	P1←REG8 after computation	INC 9 bits
INCW	INC 17 bits, REGL8←lower-byte of CY	P1←REGH8 after computation	INC 17 bits
DEC	DEC 9 bits	P1←REG8 after computation	DEC 9 bits
DECW	DEC 17 bits, REGL8← lower byte of CY inverted	P1←REGH8 after computation	DEC 17 bits
DBNZ	DEC 9 bits	P1←REG8	DEC 9 bits, check lower-order 8 bits
DBZ	DEC 9 bits	P1←REG8	DEC 9 bits, check lower-order 8 bits
SET1	—	—	
NOT1	—	—	
CLR1	—	—	
BPC	—	—	
BP	—	—	
BN	—	—	
MUL24 /DIV24	RAM8←"1"	—	Bit 8 of RAM address for storing results is set to 1.
FUNC	—	—	

Note: *A "1" is read and processed if the processing target is an 8-bit register (no bit 8).*

Legends:

REG8:   *Bit 8 of a RAM or SFR location*
REGH8/REGL8: Bit 8 of the higher-order byte of a RAM location or SFR/bit 8 of the lower-order byte
RAM8:   *Bit 8 of a RAM location*
RAMH8/RAML8: Bit 8 of the higher-order byte of a RAM location/bit 8 of the lower-order byte

3. Peripheral System Configuration

This chapter describes the internal functional blocks (peripheral system) of the this series of microcontrollers except the CPU core, RAM, and ROM. Port block diagrams are provided in Appendix A-II for reference.

3.1   Port 0

3.1.1   Overview

Port 0 is an 8-bit I/O port equipped with programmable pull-up resistors. It is made up of a data latch, a data direction register, and a control circuit. Control of the input/output signal direction and the pull-up resistors is accomplished through the data direction register in 4-bit units.

This port can also serve as a pin for external interrupts and can release HOLD mode. As a user option, either CMOS output with a programmable pull-up resistor or N-channel open drain output can be selected as the output type in 1-bit units.

3.1.2   Functions

1)   Input/output port (8 bits: P00-P07)

- The port output data is controlled by port 0 data latch (P0: FE40) in 1-bit units.
- I/O control of P00 to P03 is accomplished by P0LDDR (P0DDR: FE41, bit 0).
- I/O control of P04 to P07 is accomplished by P0HDDR (P0DDR: FE41, bit 1).
- Ports selected as CMOS outputs by mask option are provided with programmable pull-up resistors.
- The programmable pull-up resistors for P00 to P03 are controlled by P0LPU (P0DDR: FE41, bit 2).
- The programmable pull-up resistors for P04 to P07 are controlled by P0HPU (P0DDR: FE41, bit 3).

2)   Interrupt pin function

P0FLG (P0DDR: FE41, bit 5) is set when an input port is specified and 0 level data is input to one of the ports whose corresponding bit in the port 0 data latch (P0: FE40) is set to 1.

In this case, if P0IE (P0DDR: FE41, bit 4) is 1, HOLD mode is released and an interrupt request to vector address 004BH is generated.

3)   Multiplexed pin functions

Pin P05 is also used as system clock output, pin P06 as the timer 6 toggle output, and pin P07 as the timer 7 toggle output. Pins P00 to P07 are also used as analog input channel pins AN0 to AN7.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE40	0000 0000	R/W	P0	P07	P06	P05	P04	P03	P02	P01	P00
FE41	HH00 0000	R/W	P0DDR	-	-	P0FLG	P0IE	P0HPU	P0LPU	P0HDDR	P0LDDR
FE4F	0000 0000	R/W	P0FCRU	T7OE	T6OE	SCK0SL5	SCK0SL4	CLKOEN	CKODV2	CKODV1	CKODV0

3.1.3 Related Registers

3.1.3.1 Port 0 data latch (P0)

1) The port 0 data latch is an 8-bit register for controlling port 0 output data and port 0 interrupts.

2) When this register is read with an instruction, data at pins P00 to P07 is read in. If P0 (FE40) is manipulated using the NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced instead of the data at port pins.

3) Port 0 data can always be read regardless of the I/O state of the port.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE40	0000 0000	R/W	P0	P07	P06	P05	P04	P03	P02	P01	P00

3.1.3.2 Port 0 data direction register (P0DDR)

1) This register is a 6-bit register that controls the I/O direction of port 0 data in 4-bit units, the pull-up resistors in 4-bit units, and port 0 interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE41	HH00 0000	R/W	P0DDR	-	-	P0FLG	P0IE	P0HPU	P0LPU	P0HDDR	P0LDDR

P0FLG (bit 5): P0 interrupt source flag

This flag is set when a low level is applied to a port 0 pin that is set up for input and the corresponding P0 (FE40) bit is set.

A HOLD mode release signal and an interrupt request to vector address 004BH are generated when this bit and the interrupt request enable bit (P0IE) are set to 1.

This bit must be cleared with an instruction as it is not cleared automatically.

P0IE (bit 4): P0 interrupt request enable

Setting this bit and P0FLG to 1 generates a HOLD mode release signal and an interrupt request to vector address 004BH

P0HPU (bit 3): P07 to P04 pull-up resistor control

When this bit is set to 1 and P0HDDR to 0, pull-up resistors are connected to ports P07 to P04 that are selected as CMOS output by mask option.

P0LPU (bit 2): P03 to P00 pull-up resistor control

When this bit is set to 1 and P0LDDR to 0, pull-up resistors are connected to ports P03 to P00 that are selected as CMOS output by mask option.

P0HDDR (bit 1): P07 to P04 I/O control

A 1 in this bit places P07 to P04 into the output mode and the contents of the corresponding port 0 data latch (P0) are output from the port.

When this bit is set to 0, P07 to P04 are placed into the input mode and P0FLG is set when a low level is detected at a port whose corresponding port 0 data latch (P0) bit is set to 1.

P0LDDR (bit 0): P03 to P00 I/O control

A 1 in this bit places P03 to P00 into the output mode and the contents of the corresponding port 0 data latch (P0) are output from the port.

When this bit is set to 0, P03 to P00 are placed into the input mode and P0FLG is set when a low level is detected at a port whose corresponding port 0 data latch (P0) bit is set to 1.

3.1.3.3 Port 0 function control register (P0FCRU)

1) This register is an 8-bit register that controls the port 0 multiplexed output pin.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE4F	0000 0000	R/W	P0FCRU	T7OE	T6OE	SCKOSL5	SCKOSL4	CLKOEN	CKODV2	CKODV1	CKODV0

T7OE (bit 7):

This bit controls the output data of pin P07. It is disabled when P07 is in the input mode.

When P07 is in the output mode:

0: Carries the value of the port data latch.

1: Carries the OR of the waveform that toggles at the interval determined by timer 7 and the value of the port data latch.

T6OE (bit 6):

This bit controls the output data of pin P06. It is disabled when P06 is in the input mode.

When P06 is in the output mode:

0: Carries the value of the port data latch.

1: Carries the OR of the waveform that toggles at the interval determined by timer 6 and the value of the port data latch.

SCKOSL5 (bit 5):
SCKOSL4 (bit 4):

These bits are used to select the clock source that is to be sent to P05.

SCKOSL5	SCKOSL4	P05 Output Clock Source
0	0	Source oscillator clock selected as the system clock
0	1	Internal RC clock
1	0	USB frequency divided clock
1	1	CF clock

CLKOEN (bit 3):

This bit controls the output data of pin P05. It is disabled when P05 is in the input mode.

When P05 is in the output mode:

0: Carries the value of the port data latch.

1: Carries the OR of the system clock output and the value of the port data latch.

CKODV2 (bit 2):
CKODV1 (bit 1):
CKODV0 (bit 0):

These bits define the frequency of the clock to be placed at P05.

000: Frequency of source oscillator clock to be placed at P05

001: 1/2 of frequency of source clock to be placed at P05

010: 1/4 of frequency of source clock to be placed at P05

011: 1/8 of frequency of source clock to be placed at P05

100: 1/16 of frequency of source clock to be placed at P05

101: 1/32 of frequency of source clock to be placed at P05

110: 1/64 of frequency of source clock to be placed at P05

111: Frequency of source oscillator clock selected as subclock

PORT 0

<Notes on the use of the clock output function>

Take notes 1) to 4) given below when using the clock output function. Anomalies may be observed in the waveform of the port clock output if these notes are violated.

1) Do not change the frequency of the clock output when CLKOEN (bit 3) is set to 1.

   → Do not change the settings of CKODV2 to CKODV0 (bits 2 to 0).

2) Do not change the output clock source selection when CLKOEN (bit 3) is set to 1.

   → Do not change the settings of SCK0SL5 and SCK0SL4 (bits 5 and 4).

3) Do not change the system clock selection when CLKOEN (bit 3) is set to 1.

   → Do not change the settings of CLKCB5 and CLKCB4 (bits 5 and 4) of the OCR register.

4) CLKOEN (bit 3) will not go to 0 immediately even when the user executes an instruction to load the P0FCRU register with the data that sets the state of CLKOEN from 1 to 0. CLKOEN is set to 0 at the end of the clock that is being output (on detection of a falling edge of the clock). Accordingly, when changing the clock divider setting or changing the system clock selection after setting CLKOEN to 0 with an instruction, be sure to read the CLKOEN value in advance and make sure that it is 0.
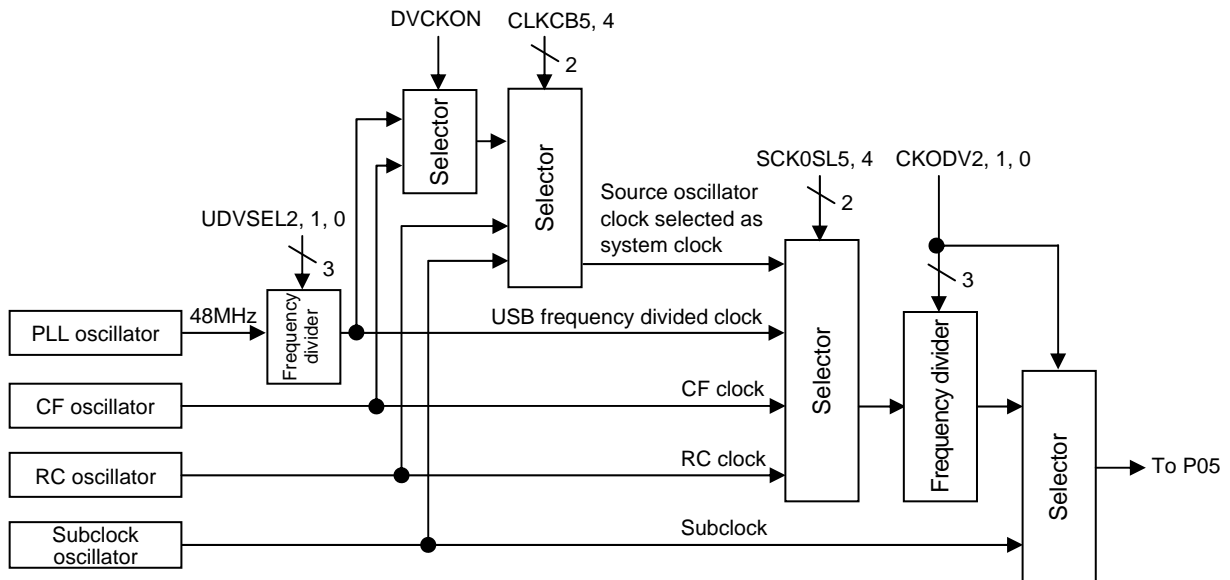
Figure 3.1.1   P05 Output Clock Selection

3.1.4　Options

Two user options are available.

1)　　CMOS output (with a programmable pull-up resistor)

2)　　N-channel open drain output

3.1.5　HALT and HOLD Mode Operation

When in HALT or HOLD mode, port 0 retains the state that is established when HALT or HOLD mode is entered.

3.2   Port 1

3.2.1   Overview

Port 1 is an 8-bit I/O port with programmable pull-up resistors. It is made up of a data latch, a data direction register, a function control register, and a control circuit. Control of the input/output signal direction is accomplished through the data direction register in 1-bit units. Port 1 can also be used as a serial interface I/O port or PWM output port by manipulating its function control register.

As a user option, either CMOS output with a programmable pull-up resistor or N-channel open drain output with a programmable pull-up resistor can be selected as the output type in 1-bit units.

3.2.2   Functions

1)   Input/output (8 bits: P10 to P17)

- The port output data is controlled by the port 1 data latch (P1: FE44) and the I/O direction is controlled by the port 1 data direction register (P1DDR: FE45).
- Each port is equipped with a programmable pull-up resistor.

2)   Multiplexed functions

P17 is also used as the timer 1 PWMH / base timer BUZ output, P16 as the timer 1 PWML output, P15 to P13 as SIO1 I/O, and P12 to P10 as SIO0 I/O.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE44	0000 0000	R/W	P1	P17	P16	P15	P14	P13	P12	P11	P10
FE45	0000 0000	R/W	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
FE46	0000 0000	R/W	P1FCR	P17FCR	P16FCR	P15FCR	P14FCR	P13FCR	P12FCR	P11FCR	P10FCR

3.2.3   Related Registers

3.2.3.1   Port 1 data latch (P1)

1)   The port 1 data latch is an 8-bit register for controlling port 1 output data and pull-up resistors.

2)   When this register is read with an instruction, data at pins P10 to P17 is read in. If P1 (FE44) is manipulated using the NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced instead of the data at port pins.

3)   Port 1 data can always be read regardless of the I/O state of the port.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE44	0000 0000	R/W	P1	P17	P16	P15	P14	P13	P12	P11	P10

3.2.3.2 Port 1 data direction register (P1DDR)

1) This register is an 8-bit register that controls the I/O direction of port 1 data in 1-bit units. Port P1n is placed in output mode when bit P1nDDR is set to 1 and in input mode when bit P1nDDR is set to 0.

2) When bit P1nDDR is set to 0 and bit P1n of the port 1 data latch is set to 1, port P1n becomes an input with a pull-up resistor.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE45	0000 0000	R/W	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR

Register Data		Port P1n State		Internal Pull-up Resistor
P1n	P1nDDR	Input	Output	
0	0	Enabled	Open	OFF
1	0	Enabled	Internal pull-up resistor	ON
0	1	Enabled	Low	OFF
1	1	Enabled	High/open (CMOS/N-channel open drain)	OFF

3.2.3.3 Port 1 function control register (P1FCR)

1) This register is an 8-bit register that controls the port 1 multiplexed output pin.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE46	0000 0000	R/W	P1FCR	P17FCR	P16FCR	P15FCR	P14FCR	P13FCR	P12FCR	P11FCR	P10FCR

n	P1nFCR	P1n	P1n Pin Data in Output Mode (P1nDDR=1)
7	0	–	Value of port data latch (P17)
	1	0	AND of timer 1 PWMH and base timer BUZ
	1	1	NAND of timer 1 PWMH and base timer BUZ
6	0	–	Value of port data latch (P16)
	1	0	Timer 1 PWML data
	1	1	Timer 1 PWML inverted data
5	0	–	Value of port data latch (P15)
	1	0	SIO1 clock output data
	1	1	High output
4	0	–	Value of port data latch (P14)
	1	0	SIO1 output data
	1	1	High output
3	0	–	Value of port data latch (P13)
	1	0	SIO1 output data
	1	1	High output
2	0	–	Value of port data latch (P12)
	1	0	SIO0 clock output data
	1	1	High output
1	0	–	Value of port data latch (P11)
	1	0	SIO0 output data
	1	1	High output
0	0	–	Value of port data latch (P10)
	1	0	SIO0 output data
	1	1	High output

The high data output at a pin that is selected as an N-channel open drain output (user option) is represented by an open circuit.

PORT 1

P17FCR (bit 7): P17 function control (timer 1 PWMH & base timer BUZ output control)

This bit controls the output data at pin P17.

When P17 is placed in the output mode (P17DDR=1) and P17FCR is set to 1, the AND of timer 1 PWMH output and BUZ output from the base timer is EORed with the port data latch and the result is placed at pin P17.

P16FCR (bit 6): P16 function control (timer 1 PWML output control)

This bit controls the output data at pin P16.

When P16 is placed in the output mode (P16DDR=1) and P16FCR is set to 1, the EOR of timer 1 PWML output data and the port data latch is placed at pin P16.

P15FCR (bit 5): P15 function control (SIO1 clock output control)

This bit controls the output data at pin P15.

When P15 is placed in the output mode (P15DDR=1) and P15FCR is set to 1, the OR of the SIO1 clock output data and the port data latch is placed at pin P15.

P14FCR (bit 4): P14 function control (SIO1 data output control)

This bit controls the output data at pin P14.

When P14 is placed in the output mode (P14DDR=1) and P14FCR is set to 1, the OR of the SIO1 output data and the port data latch is placed at pin P14.

When the SIO1 is active, SIO1 input data is read from P14 regardless of the I/O state of P14.

P13FCR (bit 3): P13 function control (SIO1 data output control)

This bit controls the output data at pin P13.

When P13 is placed in the output mode (P13DDR=1) and P13FCR is set to 1, the OR of the SIO1 output data and the port data latch is placed at pin P13.

P12FCR (bit 2): P12 function control (SIO0 clock output control)

This bit controls the output data at pin P12.

When P12 is placed in the output mode (P12DDR=1) and P12FCR is set to 1, the OR of the SIO0 clock output data and the port data latch is placed at pin P12.

P11FCR (bit 1): P11 function control (SIO0 data output control)

This bit controls the output data at pin P11.

When P11 is placed in the output mode (P11DDR=1) and P11FCR is set to 1, the OR of the SIO0 output data and the port data latch is placed at pin P11.

When the SIO0 is active, SIO0 input data is read from P11 regardless of the I/O state of P11.

P10FCR (bit 0): P10 function control (SIO0 data output control)

This bit controls the output data at pin P10.

When P10 is placed in the output mode (P10DDR=1) and P10FCR is set to 1, the OR of the SIO0 output data and the port data latch is placed at pin P10.

3.2.4   Options

Two user options are available.

1)	CMOS output	(with a programmable pull-up resistor)
2)	N-channel open drain output	(with a programmable pull-up resistor)

3.2.5   HALT and HOLD Mode Operation

When in HALT or HOLD mode, port 1 retains the state that is established when HALT or HOLD mode is entered.

3.3   Port 2

3.3.1   Overview

Port 2 is an 8-bit I/O port with programmable pull-up resistors. It is made up of a data latch, a data direction register, and a control circuit. Control of the input/output signal direction is accomplished through the data direction register in 1-bit units.

Port 2 can also serve as an input port for external interrupts. It can also be used as an input port for the timer 1 count clock input, timer 0 capture signal input, and HOLD mode release signal input.

As a user option, either CMOS output with a programmable pull-up resistor or N-channel open drain output with a programmable pull-up resistor can be selected as the output type in 1-bit units.

3.3.2   Functions

1)   Input/output port (8 bits: P20 to P27)

- The port 2 data latch (P2: FE48) is used to control port output data, and the port 2 data direction register (P2DDR: FE49) to control the I/O direction of port data.

- Each port is equipped with a programmable pull-up resistor.

2)   Interrupt input pin function

- The port (INT4) selected out of P20 to P23 and the port (INT5) selected out of P24 to P27 are provided with a pin interrupt function. This function detects a low edge, a high edge, or both edges and sets the interrupt flag. These two selected ports can also serve as timer 1 count clock input and timer 0 capture signal input.

3)   Hold mode release function

- When the interrupt flag and interrupt enable flag are set by INT4 or INT5, a HOLD mode release signal is generated, releasing HOLD mode. The CPU then enters HALT mode (main oscillation by CR). When the interrupt is accepted, the CPU switches from HALT mode to normal operating mode.

- When a signal change that sets the interrupt flag is input to INT4 or INT5 in HOLD mode, the interrupt flag is set. In this case, HOLD mode is released if the corresponding interrupt enable flag is set.

    The interrupt flag, however, cannot be set by a rising edge occurring when INT4 or INT5 data which is established when HOLD mode is entered is in the high state, or by a falling edge occurring when INT4 or INT5 data which is established when HOLD mode is entered is in the low state. Consequently, to release HOLD mode with INT4 or INT5, it is recommended that INT4 or INT5 be used in the double edge interrupt mode.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE48	0000 0000	R/W	P2	P27	P26	P25	P24	P23	P22	P21	P20
FE49	0000 0000	R/W	P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
FE4A	0000 0000	R/W	145CR	INT5HEG	INT5LEG	INT5IF	INT5IE	INT4HEG	INT4LEG	INT4IF	INT4IE
FE4B	0000 0000	R/W	145SL	15SL3	15SL2	15SL1	15SL0	14SL3	14SL2	14SL1	14SL0

3.3.3   Related Registers

3.3.3.1   Port 2 data latch (P2)

1) The port 2 data latch is an 8-bit register for controlling port 2 output data and pull-up resistors.

2) When this register is read with an instruction, data at pins P20 to P27 is read in. If P2 (FE48) is manipulated using the NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced instead of the data at port pins.

3) Port 2 data can always be read regardless of the I/O state of the port.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE48	0000 0000	R/W	P2	P27	P26	P25	P24	P23	P22	P21	P20

3.3.3.2   Port 2 data direction register (P2DDR)

1) This register is an 8-bit register that controls the I/O direction of port 2 data in 1-bit units. Port P2n is placed in output mode when bit P2nDDR is set to 1 and in input mode when bit P2nDDR is set to 0.

2) When bit P2nDDR is set to 0 and bit P2n of the port 2 data latch is set to 1, port P2n becomes an input with a pull-up resistor

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE49	0000 0000	R/W	P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR

Register Data		Port P2n State		Internal Pull-up Resistor
P2n	P2nDDR	Input	Output	
0	0	Enabled	Open	OFF
1	0	Enabled	Internal pull-up resistor	ON
0	1	Enabled	Low	OFF
1	1	Enabled	High/open (CMOS/N-channel open drain)	OFF

3.3.3.3   External interrupt 4/5 control register (I45CR)

1) This register is an 8-bit register for controlling external interrupts 4 and 5.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE4A	0000 0000	R/W	I45CR	INT5HEG	INT5LEG	INT5IF	INT5IE	INT4HEG	INT4LEG	INT4IF	INT4IE

INT5HEG (bit 7): INT5 rising edge detection control
INT5LEG (bit 6): INT5 falling edge detection control

INT5HEG	INT5LEG	INT5 Interrupt Conditions (Pin Data)
0	0	No edge detected
0	1	Falling edge detected
1	0	Rising edge detected
1	1	Both edges detected

INT5IF (bit 5): INT5 interrupt source flag

This bit is set when the conditions specified by INT5HEG and INT5LEG are satisfied.

When this bit and the INT5 interrupt request enable bit (INT5IE) are set to 1, a HOLD mode release signal and an interrupt request to vector address 001BH are generated.

The interrupt flag, however, cannot be set by a rising edge occurring when INT5 data which is established when HOLD mode is entered is in the high state, or by a falling edge occurring when INT5 data which is established when HOLD mode is entered is in the low state. Consequently, to release HOLD mode with INT5, it is recommended that INT5 be used in the double edge interrupt mode.

This bit must be cleared with an instruction as it is not cleared automatically.

INT5IE (bit 4): INT5 interrupt request enable

When this bit and INT5IF are set to 1, a HOLD mode release signal and an interrupt request to vector address 001BH are generated.

INT4HEG (bit 3): INT4 rising edge detection control
INT4LEG (bit 2): INT4 falling edge detection control

INT4HEG	INT4LEG	INT4 Interrupt Conditions (Pin Data)
0	0	No edge detected
0	1	Falling edge detected
1	0	Rising edge detected
1	1	Both edges detected

INT4IF (bit 1): INT4 interrupt source flag

This bit is set when the conditions specified by INT4HEG and INT4LEG are satisfied.

When this bit and the INT4 interrupt request enable bit (INT4IE) are set to 1, a HOLD mode release signal and an interrupt request to vector address 0013H are generated.

The interrupt flag, however, cannot be set by a rising edge occurring when INT4 data which is established when HOLD mode is entered is in the high state, or by a falling edge occurring when INT4 data which is established when HOLD mode is entered is in the low state. Consequently, to release HOLD mode with INT4, it is recommended that INT4 be used in the double edge interrupt mode.

This bit must be cleared with an instruction as it is not cleared automatically.

INT4IE (bit 0): INT4 interrupt request enable

When this bit and INT4IF are set to 1, a HOLD mode release signal and an interrupt request to vector address 0013H are generated.

3.3.3.4   External interrupt 4/5 pin select register (I45SL)

1)   This register is an 8-bit register used to select pins for the external interrupts 4 and 5.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE4B	0000 0000	R/W	I45SL	15SL3	15SL2	15SL1	15SL0	14SL3	14SL2	14SL1	14SL0

I5SL3 (bit 7): INT5 pin select

I5SL2 (bit 6): INT5 pin select

I5SL3	I5SL2	Pin Assigned to INT5
0	0	Port P24
0	1	Port P25
1	0	Port P26
1	1	Port P27

I5SL1 (bit 5): INT5 pin function select

I5SL0 (bit 4): INT5 pin function select

When the data change specified in the external interrupt 4/5 control register (I45CR) is given to the pin that is assigned to INT5, timer 1 count clock input and timer 0 capture signal are generated.

I5SL1	I5SL0	Function other than INT5 Interrupt
0	0	None
0	1	Timer 1 count clock input
1	0	Timer 0L capture signal input
1	1	Timer 0H capture signal input

I4SL3 (bit 3): INT4 pin select

I4SL2 (bit 2): INT4 pin select

I4SL3	I4SL2	Pin Assigned to INT4
0	0	Port P20
0	1	Port P21
1	0	Port P22
1	1	Port P23

I4SL1 (bit 1): INT4 pin function select

I4SL0 (bit 0): INT4 pin function select

When the data change specified in the external interrupt 4/5 control register (I45CR) is given to the pin that is assigned to INT4, timer 1 count clock input and timer 0 capture signal are generated.

I4SL1	I4SL0	Function other than INT4 Interrupt
0	0	None
0	1	Timer 1 count clock input
1	0	Timer 0L capture signal input
1	1	Timer 0H capture signal input

Notes:

1) *If timer 0L capture signal input or timer 0H capture signal input is specified for INT4 or INT5 together with port 7, the signal from port 7 is ignored.*

2) *If INT4 and INT5 are specified in duplicate for timer 1 count clock input, timer 0L capture signal input, or timer 0H capture signal input, both interrupts are accepted. If both INT4 and INT5 events occur at the same time, however, only one event is recognized.*

3) *When at least one of INT4 and INT5 is specified as timer 1 count clock input, timer 1L functions as an event counter. If neither INT4 nor INT5 is specified for timer 1 count clock input, the timer 1L counts on every 2 Tcyc.*

3.3.4 Options

Two user options are available.

1) CMOS output   (with a programmable pull-up resistor)
2) N-channel open drain output   (with a programmable pull-up resistor)

3.3.5 HALT and HOLD Mode Operation

When in HALT or HOLD mode, port 2 retains the state that is established when HALT or HOLD mode is entered.

3.4　Port 3

3.4.1　Overview

Port 3 is a 5-bit I/O port with programmable pull-up resistors. It is made up of a data latch, a data direction register, and a control circuit. Control of the input/output signal direction is accomplished through the data direction register in 1-bit units.

As a user option, either CMOS output with a programmable pull-up resistor or N-channel open drain output with a programmable pull-up resistor can be selected as the output type in 1-bit units.

Note: Port 30 is temporally set low when the microcontroller is reset.

3.4.2　Functions

1)　Input/output port (5 bits: P30 to P34)

- The port 3 data latch (P3: FE4C) is used to control the port output data, and the port 3 data direction register (P3DDR: FE4D) to control the I/O direction of port data.
- Each port is equipped with a programmable pull-up resistor.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE4C	HHH0 0000	R/W	P3	-	-	-	P34	P33	P32	P31	P30
FE4D	HHH0 0000	R/W	P3DDR	-	-	-	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR

3.4.3　Related Registers

3.4.3.1　Port 3 data latch (P3)

1)　This data latch is a 5-bit register for controlling the port 3 output data and its pull-up resistors.

2)　When this register is read with an instruction, the data at pins P30 to P34 is read in. If P3 (FE4C) is manipulated using the NOT1, CLR1, SET1, DBZ, DBNZ, INC or DEC instruction, the contents of the register is referenced instead of the data at the pins.

3)　Port 3 data can always be read regardless of the I/O state of the port.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE4C	HHH0 0000	R/W	P3	-	-	-	P34	P33	P32	P31	P30

3.4.3.2　Port 3 data direction register (P3DDR)

1)　This register is a 5-bit register for controlling the I/O direction of the port 3 data in 1-bit units. Port P3n is placed in output mode when bit P3nDDR is set to 1 and in input mode when bit P3nDDR is set to 0.

2)　Port P3n is designated as an input with a pull-up resistor when bit P3nDDR is set to 0 and bit P3n of port 3 data latch is set to 1.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE4D	HHH0 0000	R/W	P3DDR	-	-	-	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR

Register Data		Port P3n State		Internal Pull-up Resistor
P3n	P3nDDR	Input	Output	
0	0	Enabled	Open	OFF
1	0	Enabled	Internal pull-up resistor	ON
0	1	Enabled	Low	OFF
1	1	Enabled	High/open (CMOS/N-channel open drain)	OFF

3.4.4 Options

Two user options are available.

1) CMOS output   (with a programmable pull-up resistor)
2) N-channel open drain output   (with a programmable pull-up resistor)

3.4.5 HALT and HOLD Mode Operation

When in HALT or HOLD mode, port 3 retains the state that is established when HALT or HOLD mode is entered.

3.5   Port 7

3.5.1   Overview

Port 7 is a 4-bit I/O port with programmable pull-up resistors. It is made up of a data control latch and a control circuit. The input/output direction of port data can be controlled in 1-bit units.

Port 7 can be used as an input port for external interrupts. It can also be used as an input port for the timer 0 count clock input, capture signal input, and HOLD mode release signal input.

There is no user option for this port.

3.5.2   Functions

1)   Input/output port (4 bits: P70 to P73)

-    The lower-order 4 bits of the port 7 control register (P7: FE5C) are used to control the port output data, and the higher-order 4 bits to control the I/O direction of port data.

-    P70 is of the N-channel open drain output type and P71 to P73 are of CMOS output type.

-    Each port is equipped with a programmable pull-up resistor.

2)   Interrupt input pin function

-    P70 and P71 are assigned to INT0 and INT1, respectively, and used to detect a low or high level, or a low or high edge and set the interrupt flag.

-    P72 and P73 are assigned to INT2 and INT3, respectively, and used to detect a low or high edge, or both edges and set the interrupt flag.

3)   Timer 0 count input function

A count signal is sent to timer 0 each time a signal change that sets the interrupt flag is supplied to the port selected from P72 and P73.

4)   Timer 0L capture input function

A timer 0L capture signal is generated each time a signal change that sets the interrupt flag is supplied to the port selected from P70 and P72.

When a selected level of signal is input to P70 that is specified for level-triggered interrupts, a timer 0L capture signal is generated at 1 cycle interval for the duration of the signal.

5)   Timer 0H capture input function

A timer 0H capture signal is generated each time a signal change that sets the interrupt flag is supplied to the port selected from P71 and P73.

When a selected level of signal is input to P71 that is specified for level-triggered interrupts, a timer 0H capture signal is generated at 1 cycle interval for the duration of the signal.

6)   HOLD mode release function

-    When the interrupt flag and interrupt enable flag are set by INT0, INT1, or INT2, a HOLD mode release signal is generated, releasing HOLD mode. The CPU then enters HALT mode (main oscillation by CR). When the interrupt is accepted, the CPU switches from HALT mode to normal operating mode.

- When a signal change that sets the interrupt flag is input to P70 or P71 that is specified for level-triggered interrupt in HOLD mode, the interrupt flag is set. In this case, HOLD mode is released if the corresponding interrupt enable flag is set.

- When a signal change that sets the interrupt flag is input to P72 in HOLD mode, the interrupt flag is set. In this case, HOLD mode is released if the corresponding interrupt enable flag is set. The interrupt flag, however, cannot be set by a rising edge occurring when P72 data which is established when HOLD mode is entered is in the high state, or by a falling edge occurring when P72 data which is established when HOLD mode is entered is in the low state. Consequently, to release HOLD mode with P72, it is recommended that P72 be used in the double edge interrupt mode.

	Input	Output	Interrupt Input Signal Detection	Timer 0 Count Input	Capture Input	Hold Mode Release
P70	With programmable pull-up resistor	N-channel open drain	L level, H level, L edge, H edge	—	Timer 0L	Enabled (Note)
P71		CMOS		—	Timer 0H	Enabled (Note)
P72			L edge, H edge, both edges	Available	Timer 0L	Enabled
P73				Available	Timer 0H	-

Note: P70 and P71 HOLD mode release is available only when level detection is set.

7) Analog voltage input function
- P70 and P71 are used to receive the analog voltage inputs to the AD converter.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5C	0000 0000	R/W	P7	P73DDR	P72DDR	P71DDR	P70DDR	P73DT	P72DT	P71DT	P70DT
FE5D	0000 0000	R/W	I01CR	INT1LH	INT1LV	INT1IF	INT1IE	INT0LH	INT0LV	INT0IF	INT0IE
FE5E	0000 0000	R/W	I23CR	INT3HEG	INT3LEG	INT3IF	INT3IE	INT2HEG	INT2LEG	INT2IF	INT2IE
FE5F	0000 0000	R/W	ISL	ST0HCP	ST0LCP	BTIMC1	BTIMC0	BUZON	NFSEL	NFON	ST0IN

3.5.3 Related Registers

3.5.3.1 Port 7 control register (P7)

1) This register is an 8-bit register for controlling the I/O of port 7 data and pull-up resistors.

2) When this register is read with an instruction, data at pins P70 to P73 is read into bits 0 to 3. Bits 4 to 7 are loaded with bits 4 to 7 of register P7. If P7 (FE5C) is manipulated using the NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced as bits 0 to 3 instead of the data at port pins.

3) Port 7 data can always be read regardless of the I/O state of the port

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5C	0000 0000	R/W	P7	P73DDR	P72DDR	P71DDR	P70DDR	P73DT	P72DT	P71DT	P70DT

Register Data		Port P7n State		Internal Pull-up Resistor
P7n	P7nDDR	Input	Output	
0	0	Enabled	Open	OFF
1	0	Enabled	Internal pull-up resistor	ON
0	1	Enabled	CMOS-Low	OFF
1	1	Enabled	CMOS-High (P70 is open)	ON

P73DDR (bit 7): P73 I/O control

      A 1 or 0 in this bit controls the output (CMOS) or input of pin P73.

P72DDR (bit 6): P72 I/O control

      A 1 or 0 in this bit controls the output (CMOS) or input of pin P72.

P71DDR (bit 5): P71 I/O control

      A 1 or 0 in this bit controls the output (CMOS) or input of pin P71.

P70DDR (bit 4): P70 I/O control

      A 1 or 0 in this bit controls the output (N-channel open drain) or input of pin P70.

P73DT (bit 3): P73 data

      The value of this bit is output from pin P73 when P73DDR is set to 1.

      A 1 or 0 in this bit turns on or off the internal pull-up resistor for pin P73.

P72DT (bit 2): P72 data

      The value of this bit is output from pin P72 when P72DDR is set to 1.

      A 1 or 0 in this bit turns on or off the internal pull-up resistor for pin P72.

P71DT (bit 1): P71 data

      The value of this bit is output from pin P71 when P71DDR is set to 1.

      A 1 or 0 in this bit turns on or off the internal pull-up resistor for pin P71.

P70DT (bit 0): P70 data

      The value of this bit is output from pin P70 when P70DDR is set to 1. Since this bit is of N-channel open drain output type, however, it is placed in the high-impedance state when P70 is set to 1.

      A 1 or 0 in this bit turns on or off the internal pull-up resistor for pin P70.

3.5.3.2 External interrupt 0/1 control register (I01CR)

1)   This register is an 8-bit register for controlling external interrupts 0 and 1.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5D	0000 0000	R/W	I01CR	INT1LH	INT1LV	INT1IF	INT1IE	INT0LH	INT0LV	INT0IF	INT0IE

INT1LH (bit 7): INT1 detection polarity select

INT1LV (bit 6): INT1 detection level/edge select

INT1LH	INT1LV	INT1 Interrupt Conditions (P71 Pin Data)
0	0	Falling edge detected
0	1	Low level detected
1	0	Rising edge detected
1	1	High level detected

INT1IF (bit 5): INT1 interrupt source flag

> This bit is set when the conditions specified by INT1LH and INT1LV are satisfied. When this bit and the INT1 interrupt request enable bit (INT1IE) are set to 1, a HOLD mode release signal and an interrupt request to vector address 000BH are generated.
>
> This bit must be cleared with an instruction as it is not cleared automatically.

INT1IE (bit 4): INT1 interrupt request enable

> When this bit and INT1IF are set to 1, a HOLD mode release signal and an interrupt request to vector address 000BH are generated.

INT0LH (bit 3): INT0 detection polarity select
INT0LV (bit 2): INT0 detection level/edge select

INT0LH	INT0LV	INT0 Interrupt Conditions (P70 Pin Data)
0	0	Falling edge detected
0	1	Low level detected
1	0	Rising edge detected
1	1	High level detected

INT0IF (bit 1): INT0 interrupt source flag

> This bit is set when the conditions specified by INT0LH and INT0LV are satisfied. When this bit and the INT0 interrupt request enable bit (INT0IE) are set to 1, a HOLD mode release signal and an interrupt request to vector address 0003H are generated.
>
> This bit must be cleared with an instruction as it is not cleared automatically.

INT0IE (bit 0): INT0 interrupt request enable

> When this bit and INT0IF are set to 1, a HOLD mode release signal and an interrupt request to vector address 0003H are generated.

3.5.3.3   External interrupt 2/3 control register (I23CR)

1)   This register is an 8 bit register for controlling external interrupts 2 and 3.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5E	0000 0000	R/W	I23CR	INT3HEG	INT3LEG	INT3IF	INT3IE	INT2HEG	INT2LEG	INT2IF	INT2IE

INT3HEG (bit 7): INT3 rising edge detection control
INT3LEG (bit 6): INT3 falling edge detection control

INT3HEG	INT3LEG	INT3 Interrupt Conditions (P73 Pin Data)
0	0	No edge detected
0	1	Falling edge detected
1	0	Rising edge detected
1	1	Both edges detected

INT3IF (bit 5): INT3 interrupt source flag

> This bit is set when the conditions specified by INT3HEG and INT3LEG are satisfied. When this bit and the INT3 interrupt request enable bit (INT3IE) are set to 1, an interrupt request to vector address 001BH is generated.
>
> This bit must be cleared with an instruction as it is not cleared automatically.

INT3IE (bit 4): INT3 interrupt request enable

When this bit and INT3IF are set to 1, an interrupt request to vector address 001BH is generated.

INT2HEG (bit 3): INT2 rising edge detection control
INT2LEG (bit 2): INT2 falling edge detection control

INT2HEG	INT2LEG	INT2 Interrupt Conditions (P72 Pin Data)
0	0	No edge detected
0	1	Falling edge detected
1	0	Rising edge detected
1	1	Both edges detected

INT2IF (bit 1): INT2 interrupt source flag

This bit is set when the conditions specified by INT2HEG and INT2LEG are satisfied.

When this bit and the INT2 interrupt request enable bit (INT2IE) are set to 1, a HOLD mode release signal and an interrupt request to vector address 0013H are generated.

The interrupt flag, however, cannot be set by a rising edge occurring when P72 data which is established when HOLD mode is entered is in the high state, or by a falling edge occurring when P72 data which is established when HOLD mode is entered is in the low state. Consequently, to release HOLD mode with P72, it is recommended that P72 be used in the double edge interrupt mode.

This bit must be cleared with an instruction as it is not cleared automatically.

INT2IE (bit 0): INT2 interrupt request enable

When this bit and INT2IF are set to 1, a HOLD mode release signal and an interrupt request to vector address 0013H are generated.

3.5.3.4 Input signal select register (ISL)

1) This register is an 8-bit register for controlling the timer 0 input, noise filter time constant, buzzer output, and base timer clock.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5F	0000 0000	R/W	ISL	ST0HCP	ST0LCP	BTIMC1	BTIMC0	BUZON	NFSEL	NFON	ST0IN

ST0HCP (bit 7): Timer 0H capture signal input port select

This bit selects the timer 0H capture signal input port.

When set to 1, a timer 0H capture signal is generated when an input that satisfies the INT1 interrupt detection conditions is supplied to P71. If the INT1 interrupt detection mode is set to level detection, capture signals are generated at an interval of 1 Tcyc as long as the detection level is present at P71.

When this bit is set to 0, a timer 0H capture signal is generated when an input that satisfies the INT3 interrupt detection conditions is supplied to P73.

ST0LCP (bit 6): Timer 0L capture signal input port select

This bit selects the timer 0L capture signal input port.

When set to 1, a timer 0L capture signal is generated when an input that satisfies the INT0 interrupt detection conditions is supplied to P70. If the INT0 interrupt detection mode is set to level detection, capture signals are generated at an interval of 1 Tcyc as long as the detection level is present at P70.

When this bit is set to 0, a timer 0L capture signal is generated when an input that satisfies the INT2 interrupt detection conditions is supplied to P72.

BTIMC1 (bit 5): Base timer clock select
BTIMC0 (bit 4): Base timer clock select

BTIMC1	BTIMC0	Base Timer Input Clock
0	0	Subclock
0	1	Cycle clock
1	0	Subclock
1	1	Timer/counter 0 prescaler output

BUZON (bit 3): Buzzer output select

This bit enables the buzzer output (fBST/16).

When set to 1, a signal that is obtained by dividing the base timer clock by 16 is sent to port P17 as buzzer output.

When this bit is set to 0, the buzzer output is fixed at the high level.

fBST: The frequency of the input clock to the base timer that is selected through the input signal select register (ISL), bits 5 and 4.

NFSEL (bit 2): Noise filter time constant select
NFON (bit 1): Noise filter time constant select

NFSEL	NFON	Noise Filter Time Constant
0	0	1 Tcyc
0	1	128 Tcyc
1	0	1 Tcyc
1	1	32 Tcyc

ST0IN (bit 0): Timer 0 count clock input port select

This bit selects the timer 0 count clock signal input port.

When set to 1, a timer 0 count clock is generated when an input that satisfies the INT3 interrupt detection conditions is supplied to P73.

When this bit is set to 0, a timer 0 count clock is generated when an input that satisfies the INT2 interrupt detection conditions is supplied to P72.

Note: When timer 0L capture signal input or timer 0H capture signal input is specified for INT4 or INT5 together with port 7, the signal from port 7 is ignored.

3.5.4   Options

There is no user option for port 7.

3.5.5   HALT and HOLD Mode Operation

The pull-up resistor to P70 is turned off.

P71 to P73 retain their state that is established when HALT or HOLD mode is entered.

3.6 Timer/Counter 0 (T0)

3.6.1 Overview

The timer/counter 0 (T0) incorporated in this series of microcontrollers is a 16-bit timer/counter that provides the following four functions:

1) Mode 0: 8-bit programmable timer with a programmable prescaler (with an 8-bit capture register) × 2 channels

2) Mode 1: 8-bit programmable timer with a programmable prescaler (with an 8-bit capture register) + 8-bit programmable counter (with an 8-bit capture register)

3) Mode 2: 16-bit programmable timer with a programmable prescaler (with a 16-bit capture register)

4) Mode 3: 16-bit programmable counter (with a 16-bit capture register)

3.6.2 Functions

1) Mode 0: 8-bit programmable timer with a programmable prescaler (with an 8-bit capture register) × 2 channels

- Two independent 8-bit programmable timers (T0L and T0H) run on the clock (with a period of 1 to 256 Tcyc) from an 8-bit programmable prescaler.

- The contents of T0L are captured into the capture register T0CAL on external input detection signals from the P70/INT0/T0LCP, P72/INT2/T0IN, and P20 to P27 timer 0L capture input pins.

- The contents of T0H are captured into the capture register T0CAH on external input detection signals from the P71/INT1/T0HCP, P73/INT3/T0IN, and P20 to P27 timer 0H capture input pins.

$$\text{T0L period} = (\text{T0LR} + 1) \times (\text{T0PRR} + 1) \times \text{Tcyc}$$
$$\text{T0H period} = (\text{T0HR} + 1) \times (\text{T0PRR} + 1) \times \text{Tcyc}$$
$$\text{Tcyc} = \text{Period of cycle clock}$$

2) Mode 1: 8-bit programmable timer with a programmable prescaler (with an 8-bit capture register) + 8-bit programmable counter (with an 8-bit capture register)

- T0L serves as an 8-bit programmable counter that counts the number of external input detection signals from the P72/INT2/T0IN and P73/INT3/T0IN pins.

- T0H serves as an 8-bit programmable timer that runs on the clock (with a period of 1 to 256 Tcyc) from an 8-bit programmable prescaler.

- The contents of T0L are captured into the capture register T0CAL on external input detection signals from the P70/INT0/T0LCP, P72/INT2/T0IN, and P20 to P27 timer 0L capture input pins.

- The contents of T0H are captured into the capture register T0CAH on external input detection signals from the P71/INT1/T0HCP, P73/INT3/T0IN, and P20 to P27 timer 0H capture input pins.

$$\text{T0L period} = (\text{T0LR} + 1)$$
$$\text{T0H period} = (\text{T0HR} + 1) \times (\text{T0PRR} + 1) \times \text{Tcyc}$$

3) Mode 2: 16-bit programmable timer with a programmable prescaler (with a 16-bit capture register)

- Timer/counter 0 serves as a 16-bit programmable timer that runs on the clock (with a period of 1 to 256 Tcyc) from an 8-bit programmable prescaler.

- The contents of T0L and T0H are captured into the capture registers T0CAL and T0CAH at the same time on external input detection signals from the P71/INT1/T0HCP, P73/INT3/T0IN, and P20 to P27 timer 0H capture input pins.

    T0 period = ([T0HR, T0LR] + 1) × (T0PRR +1) × Tcyc

    16 bits

4)   Mode 3:   16-bit programmable counter (with a 16-bit capture register)

- Timer/counter 0 serves as a 16-bit programmable counter that counts the number of external input detection signals from the P72/INT2/T0IN and P73/INT3/T0IN pins.

- The contents of T0L and T0H are captured into the capture registers T0CAL and T0CAH at the same time on external input detection signals from the P71/INT1/T0HCP, P73/INT3/T0IN, and P20 to P27 timer 0H capture input pins.

    T0 period = [T0HR, T0LR] + 1

    16 bits

5)   Interrupt generation

T0L or T0H interrupt request is generated at the counter interval for timer/counter T0L or T0H if the interrupt request enable bit is set.

6)   To control timer/counter 0 (T0), it is necessary to manipulate the following special function registers.
- T0CNT, T0PRR, T0L, T0H, T0LR, T0HR
- P7, ISL, I01CR, I23CR
- P2, P2DDR, I45CR, I45SL

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE10	0000 0000	R/W	T0CNT	T0HRUN	T0LRUN	T0LONG	T0LEXT	T0HCMP	T0HIE	T0LCMP	T0LIE
FE11	0000 0000	R/W	T0PRR	T0PRR7	T0PRR6	T0PRR5	T0PRR4	T0PRR3	T0PRR2	T0PRR1	T0PRR0
FE12	0000 0000	R	T0L	T0L7	T0L6	T0L5	T0L4	T0L3	T0L2	T0L1	T0L0
FE13	0000 0000	R	T0H	T0H7	T0H6	T0H5	T0H4	T0H3	T0H2	T0H1	T0H0
FE14	0000 0000	R/W	T0LR	T0LR7	T0LR6	T0LR5	T0LR4	T0LR3	T0LR2	T0LR1	T0LR0
FE15	0000 0000	R/W	T0HR	T0HR7	T0HR6	T0HR5	T0HR4	T0HR3	T0HR2	T0HR1	T0HR0
FE16	XXXX XXXX	R	T0CAL	T0CAL7	T0CAL6	T0CAL5	T0CAL4	T0CAL3	T0CAL2	T0CAL1	T0CAL0
FE17	XXXX XXXX	R	T0CAH	T0CAH7	T0CAH6	T0CAH5	T0CAH4	T0CAH3	T0CAH2	T0CAH1	T0CAH0
FE1E	XXXX XXXX	R	T0CA1L	T0CA1L7	T0CA1L6	T0CA1L5	T0CA1L4	T0CA1L3	T0CA1L2	T0CA1L1	T0CA1L0
FE1F	XXXX XXXX	R	T0CA1H	T0CA1H7	T0CA1H6	T0CA1H5	T0CA1H4	T0CA1H3	T0CA1H2	T0CA1H1	T0CA1H0

3.6.3   Circuit Configuration

3.6.3.1   Timer/counter 0 control register (T0CNT)   (8-bit register)
1)   This register controls the operation and interrupts of T0L and T0H.

3.6.3.2   Programmable prescaler match register (T0PRR)   (8-bit register)
1)   This register stores the match data for the programmable prescaler.

3.6.3.3   Programmable prescaler   (8-bit counter)

1)   Start/stop: This register runs in modes other than HOLD mode.

2)   Count clock: Cycle clock (period = 1 Tcyc).

3)   Match signal: A match signal is generated when the count value matches the value of register T0PRR (period: 1 to 256 Tcyc)

4)   Reset: The counter starts counting from 0 when a match signal occurs or when data is written into T0PRR.

3.6.3.4   Timer/counter 0 low byte (T0L)   (8-bit counter)

1)   Start/stop: This counter is stopped and started by the 0/1 value of T0LRUN (timer 0 control register, bit 6).

2)   Count clock: Either a prescaler match signal or an external signal must be selected through the 0/1 value of T0LEXT (timer 0 control register, bit 4).

3)   Match signal: A match signal is generated when the count value matches the value of the match buffer register (16 bits of data must match in the 16-bit mode).

4)   Reset: When the counter stops operation or a match signal is generated.

3.6.3.5   Timer/counter 0 high byte (T0H)   (8-bit counter)

1)   Start/stop: This counter is stopped and started by the 0/1 value of T0HRUN (timer 0 control register, bit 7).

2)   Count clock: Either prescaler match signal or T0L match signal must be selected through the 0/1 value of T0LONG (timer 0 control register, bit 5).

3)   Match signal: A match signal is generated when the count value matches the value of the match buffer register (16 bits of data needs to match in the 16-bit mode).

4)   Reset: When the counter stops operation or a match signal is generated.

3.6.3.6   Timer/counter 0 match data register low byte (T0LR)   (8-bit register with a match buffer register)

1)   This register is used to store the match data for T0L. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register matches the value of the lower-order byte of timer/counter 0 (16 bits of data needs to match in the 16-bit mode).

2)   The match buffer register is updated as follows:

   • When it is inactive (T0LRUN=0), the match register matches T0LR.

   • When it is active (T0LRUN=1), the match buffer register is loaded with the contents of T0LR when a match signal is generated.

3.6.3.7   Timer/counter 0 match data register high byte (T0HR)   (8-bit register with a match buffer register)

1)   This register is used to store the match data for T0H. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register matches the value of the higher-order byte of timer/counter 0 (16 bits of data needs to match in the 16-bit mode).

2)   The match buffer register is updated as follows:

   • When it is inactive (T0HRUN=0), the match register matches T0HR.

   • When it is active (T0HRUN=1), the match buffer register is loaded with the contents of T0HR when a match signal is generated.

3.6.3.8   Timer/counter 0 capture register low byte (T0CAL)   (8-bit register)

1)   Capture clock: External input detection signals from the P70/INT0/T0LCP, P72/INT2/T0IN, and P20 to P27 timer 0L capture input pins when T0LONG (timer 0 control register, bit 5) is set to 0.

External input detection signals from the P71/INT1/T0HCP, P73/INT3/T0IN, and P20 to P27 timer 0L capture input pins when T0LONG (timer 0 control register, bit 5) is set to 1.

2)   Capture data: Contents of timer/counter 0 low byte (T0L).

3.6.3.9   Timer/counter 0 capture register high byte (T0CAH)   (8-bit register)

1)   Capture clock: External input detection signals from the P71/INT1/T0HCP, P73/INT3/T0IN, and P20 to P27 timer 0H capture input pins.

2)   Capture data: Contents of timer/counter 0 high byte (T0H)

Table 3.6.1   Timer 0 (T0H, T0L) Count Clocks

Mode	T0LONG	T0LEXT	T0H Count Clock	T0L Count Clock	[T0H, T0L] Count Clock
0	0	0	T0PRR match signal	T0PRR match signal	−
1	0	1	T0PRR match signal	External signal	−
2	1	0	−	−	T0PRR match signal
3	1	1	−	−	External signal
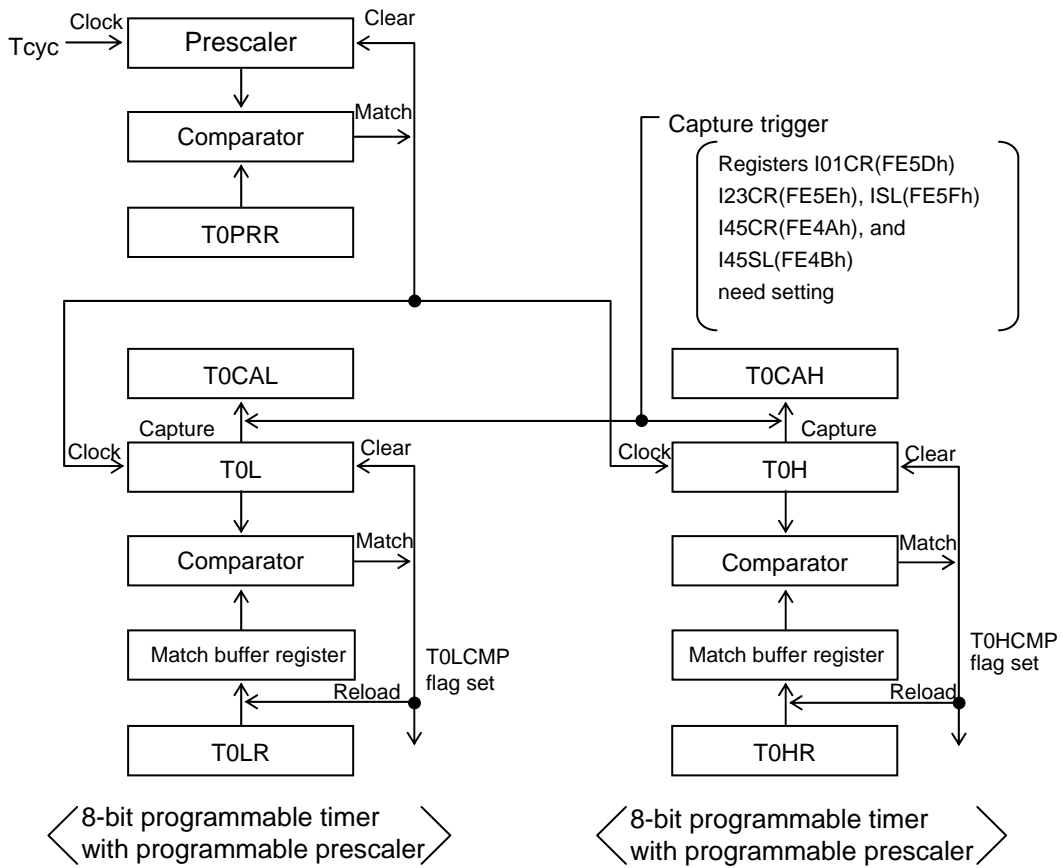
Figure 3.6.1   Mode 0 Block Diagram (T0LONG =0 , T0LEXT = 0)
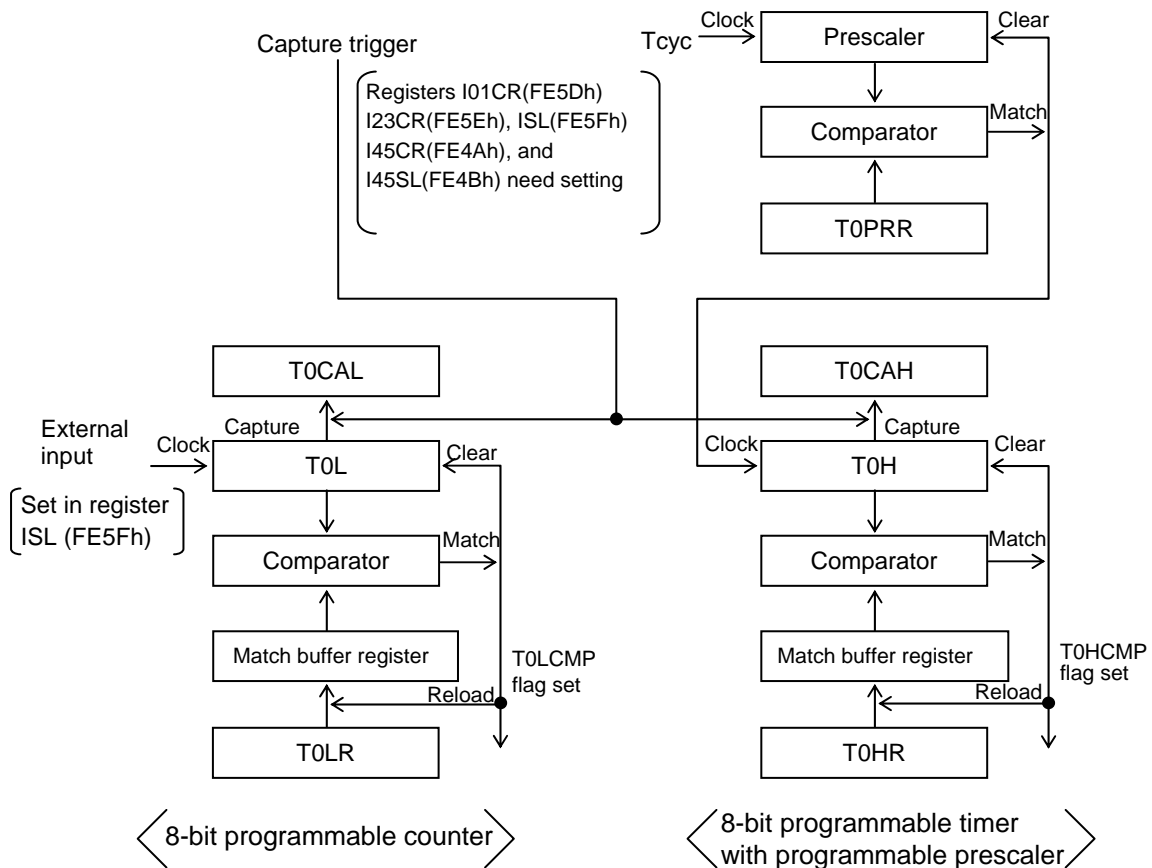
Figure 3.6.2   Mode 1 Block Diagram (T0LONG = 0, T0LEXT = 1)
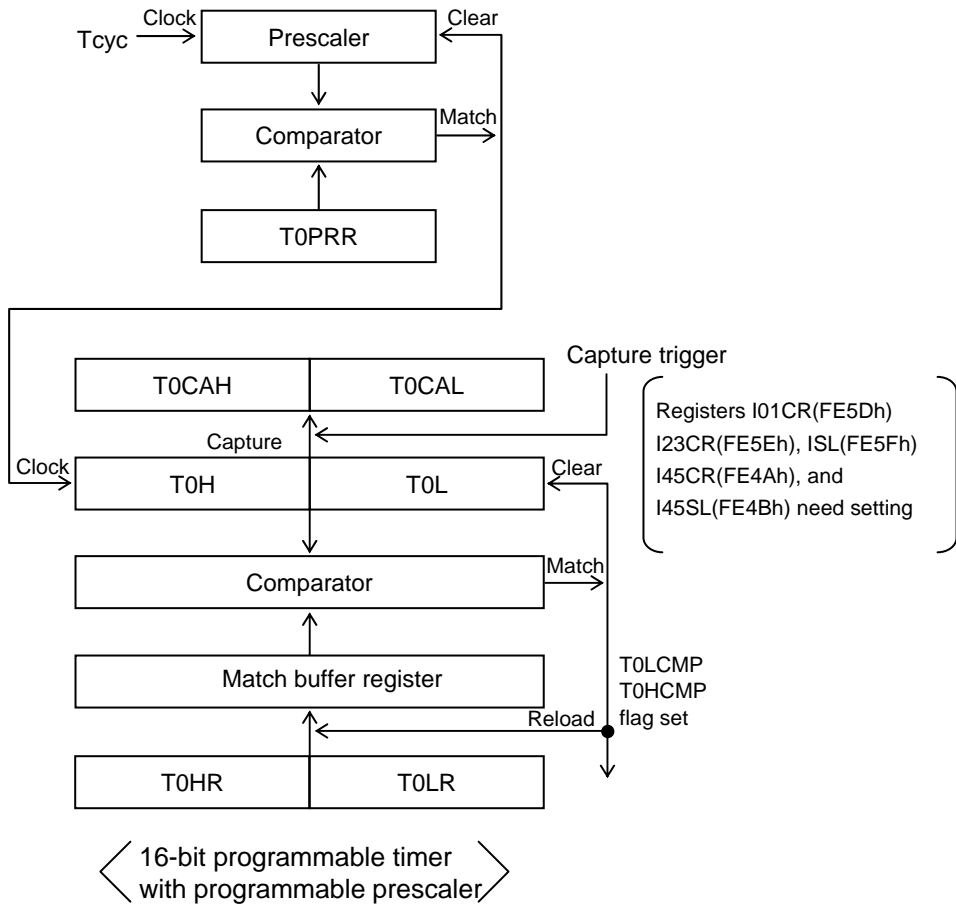
Figure 3.6.3   Mode 2 Block Diagram (T0LONG = 1, T0LEXT = 0)
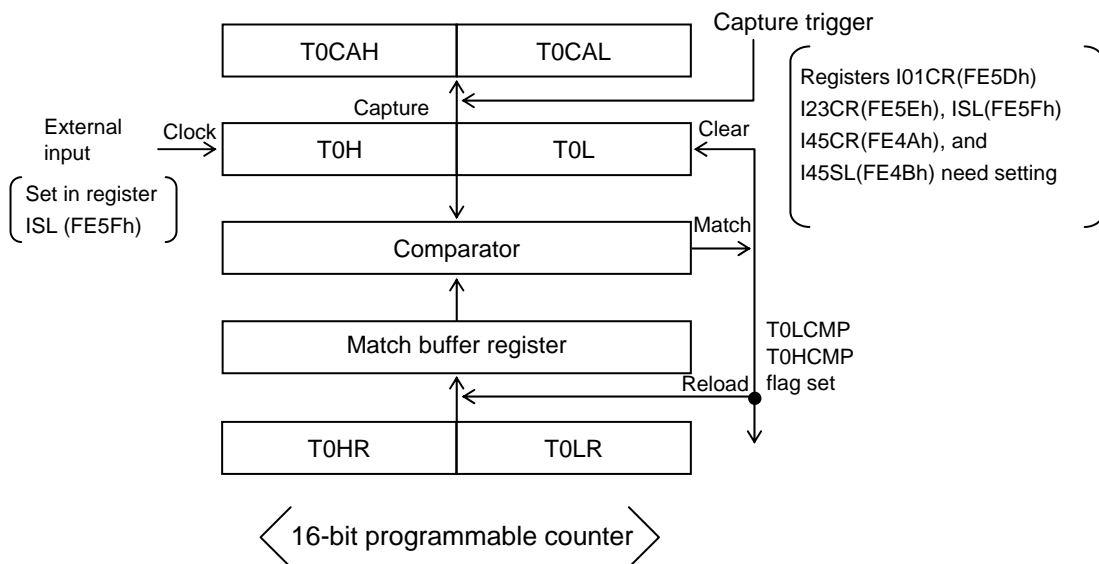
Figure 3.6.4   Mode 3 Block Diagram (T0LONG = 1, T0LEXT = 1)

3.6.4   Related Registers

3.6.4.1   Timer/counter 0 control register (T0CNT)

1)   This register is an 8-bit register that controls the operation and interrupts of T0L and T0H.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE10	0000 0000	R/W	T0CNT	T0HRUN	T0LRUN	T0LONG	T0LEXT	T0HCMP	T0HIE	T0LCMP	T0LIE

T0HRUN (bit 7): T0H count control

When this bit is set to 0, timer/counter 0 high byte (T0H) stops on a count value of 0. The match buffer register of T0H has the same value as T0HR.

When this bit is set to 1, timer/counter 0 high byte (T0H) performs the required counting operation. The match buffer register of T0H is loaded with the contents of T0HR when a match signal is generated.

T0LRUN (bit 6): T0L count control

When this bit is set to 0, timer/counter 0 low byte (T0L) stops on a count value of 0. The match buffer register of T0L has the same value as that of T0LR.

When this bit is set to 1, timer/counter 0 low byte (T0L) performs the required counting operation. The match buffer register of T0L is loaded with the contents of T0LR when a match signal is generated.

T0LONG (bit 5): Timer/counter 0 bit length select

When this bit is set to 0, timer/counter 0 higher- and lower-order bytes serve as independent 8-bit timers/counters.

When this bit is set to 1, timer/counter 0 functions as a 16-bit timer/counter. A match signal is generated when the count value of the 16-bit counter comprising T0H and T0L matches the contents of the match buffer register of T0H and T0L.

T0LEXT (bit 4): T0L input clock select

When this bit is set to 0, the count clock for T0L is the match signal for the prescaler.

When this bit is set to 1, the count clock for T0L is an external input signal.

T0HCMP (bit 3): T0H match flag

This bit is set when the value of T0H matches the value of the match buffer register for T0H and a match signal is generated while T0H is running (T0HRUN=1). Its state does not change if no match signal is generated. Consequently, this flag must be cleared with an instruction.

In the 16-bit mode (T0LONG=1), a match needs to occur in all 16 bits of data for a match signal to occur.

T0HIE (bit 2): T0H interrupt request enable control

When this bit and T0HCMP are set to 1, an interrupt request to vector address 0023H is generated.

T0LCMP (bit 1): T0L match flag

This bit is set when the value of T0L matches the value of the match buffer register for T0L and a match signal is generated while T0L is running (T0LRUN=1). Its state does not change if no match signal is generated. Consequently, this flag must be cleared with an instruction.

In the 16-bit mode (T0LONG=1), a match needs to occur in all 16 bits of data for a match signal to occur.

T0LIE (bit 0): T0L interrupt request enable control

When this bit and T0LCMP are set to 1, an interrupt request to vector address 0013H is generated.

Notes:

- *T0HCMP and T0LCMP must be cleared to 0 with an instruction.*
- *When the 16-bit mode is to be used, T0LRUN and T0HRUN must be set to the same value to control operation.*
- *T0LCMP and T0HCMP are set at the same time in the 16-bit mode.*

3.6.4.2 Timer 0 programmable prescaler match register (T0PRR)

1) Timer 0 programmable prescaler match register is an 8-bit register that is used to define the clock period (Tpr) of timer/counter 0.

2) The count value of the prescaler starts at 0 when T0PRR is loaded with data.

3) $Tpr = (T0PRR+1) \times Tcyc$ Tcyc = Period of cycle clock

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE11	0000 0000	R/W	T0PRR	T0PRR7	T0PRR6	T0PRR5	T0PRR4	T0PRR3	T0PRR2	T0PRR1	T0PRR0

3.6.4.3 Timer/counter 0 low byte (T0L)

1) This is a read-only 8-bit timer/counter. It counts the number of match signals from the prescaler or external signals.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE12	0000 0000	R	T0L	T0L7	T0L6	T0L5	T0L4	T0L3	T0L2	T0L1	T0L0

3.6.4.4 Timer/counter 0 high byte (T0H)

1) This is a read-only 8-bit timer/counter. It counts the number of match signals from the prescaler or overflow occurring in T0L.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE13	0000 0000	R	T0H	T0H7	T0H6	T0H5	T0H4	T0H3	T0H2	T0H1	T0H0

3.6.4.5 Timer/counter 0 match data register low byte (T0LR)

1) This register is used to store the match data for T0L. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register matches the value of the lower-order byte of timer/counter 0 (16 bits of data needs to match in the 16-bit mode).

2) The match buffer register is updated as follows:

• When it is inactive (T0LRUN=0), the match register matches T0LR.

• When it is active (T0LRUN=1), the match buffer register is loaded with the contents of T0LR when a match signal is generated.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE14	0000 0000	R/W	T0LR	T0LR7	T0LR6	T0LR5	T0LR4	T0LR3	T0LR2	T0LR1	T0LR0

3.6.4.6 Timer/counter 0 match data register high byte (T0HR)

1) This register is used to store the match data for T0H. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register matches the value of the higher-order byte of timer/counter 0 (16 bits of data needs to match in the 16-bit mode).

2) The match buffer register is updated as follows:

• When it is inactive (T0HRUN=0), the match register matches T0HR.

• When it is active (T0HRUN=1), the match buffer register is loaded with the contents of T0HR when a match signal is generated.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE15	0000 0000	R/W	T0HR	T0HR7	T0HR6	T0HR5	T0HR4	T0HR3	T0HR2	T0HR1	T0HR0

3.6.4.7 Timer/counter 0 capture register low byte (T0CAL)

1) This register is a read-only 8-bit register used to capture the contents of timer/counter 0 low byte (T0L) on an external input detection signal.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE16	XXXX XXXX	R	T0CAL	T0CAL7	T0CAL6	T0CAL5	T0CAL4	T0CAL3	T0CAL2	T0CAL1	T0CAL0

3.6.4.8 Timer/counter 0 capture register high byte (T0CAH)

1) This register is a read-only 8-bit register used to capture the contents of timer/counter 0 high byte (T0H) on an external input detection signal.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE17	XXXX XXXX	R	T0CAH	T0CAH7	T0CAH6	T0CAH5	T0CAH4	T0CAH3	T0CAH2	T0CAH1	T0CAH0

3.7 High-speed Clock Counter

3.7.1 Overview

The high-speed clock counter is a 3-bit counter that has a real-time output capability. It is coupled with timer/counter 0 to form an 11- or 19-bit high-speed counter. It can accept clocks with periods of as short as $\frac{1}{6}$ the cycle time. The high-speed clock counter is also equipped with a 4-bit capture register incorporating a carry bit.

3.7.2 Functions

1) 11-bit or 19-bit programmable high-speed counter

 • The 11-bit or 19-bit timer/counter, coupling the timer/counter 0 low byte (T0L) and timer/counter 0 high byte (T0H), functions as an 11- or 19-bit programmable high-speed counter that counts the external input signals from the P72/INT2/T0IN/NKIN pin. The coupled timer/counter 0 counts the number of overflows occurring in the 3-bit counter. In this case, timer 0 functions as a free-running counter.

2) Real-time output

 • A real-time output is placed at pin P17. Real-time output is a function to change the state of output at a port in real-time when the count value of a counter reaches the required value. This output change occurs asynchronously to the microcontroller clock.

3) Capture operation

 • The value of the high-speed clock counter is captured into NKCOV and NKCAP2 to NKCAP0 in synchronization with the capture operation of T0L (timer 0 low byte). NKCOV is a carry into timer/counter 0. When this bit is set to 1, the capture value of timer/counter 0 must be corrected by +1.

 NKCAP2 to NKCAP0 carry the capture value of the high-speed clock counter.

4) Interrupt generation

 • The required timer/counter 0 flag is set when the high-speed clock counter and timer/counter 0 keep counting and their count value reaches "(timer 0 match register value+1) × 8+ NKCMP2 to NKCMP0." In this case, a T0L or T0H interrupt request is generated if the interrupt request enable bit is set.

5) To control the high-speed clock counter, it is necessary to manipulate the following special function registers.

- NKREG, P1TST, T0CNT, T0L, T0H, T0LR, T0HR
- P7, ISL, I01CR, I23CR
- P2, P2DDR, I45CR, I45SL

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE7D	0000 0000	R/W	NKREG	NKEN	NKCMP2	NKCMP1	NKCMP0	NKCOV	NKCAP2	NKCAP1	NKCAP0
FE47	0HHH H0H0	R/W	P1TST	FIX0	-	-	-	-	DSNKOT	-	FIX0
FE10	0000 0000	R/W	T0CNT	T0HRUN	T0LRUN	T0LONG	T0LEXT	T0HCMP	T0HIE	T0LCNP	T0LIE
FE12	0000 0000	R	T0L	T0L7	T0L6	T0L5	T0L4	T0L3	T0L2	T0L1	T0L0
FE13	0000 0000	R	T0H	T0H7	T0H6	T0H5	T0H4	T0H3	T0H2	T0H1	T0H0
FE14	0000 0000	R/W	T0LR	T0LR7	T0LR6	T0LR5	T0LR4	T0LR3	T0LR2	T0LR1	T0LR0
FE15	0000 0000	R/W	T0HR	T0HR7	T0HR6	T0HR5	T0HR4	T0HR3	T0HR2	T0HR1	T0HR0
FE16	XXXX XXXX	R	T0CAL	T0CAL7	T0CAL6	T0CAL5	T0CAL4	T0CAL3	T0CAL2	T0CAL1	T0CAL0
FE17	XXXX XXXX	R	T0CAH	T0CAH7	T0CAH6	T0CAH5	T0CAH4	T0CAH3	T0CAH2	T0CAH1	T0CAH0
FE5D	0000 0000	R/W	I01CR	INT1LH	INT1LV	INT1IF	INT1IE	INT0LH	INT0LV	INT0IF	INT0IE
FE5E	0000 0000	R/W	I23CR	INT3HEG	INT3LEG	INT3IF	INT3IE	INT2HEG	INT2LEG	INT2IF	INT2IE
FE5F	0000 0000	R/W	ISL	ST0HCP	ST0LCP	BTIMC1	BTIMC0	BUZON	NFSEL	NFON	ST0IN

3.7.3   Circuit Configuration

3.7.3.1   High-speed clock counter control register (NKREG)   (8-bit register)

1) The high-speed clock counter control register controls the high-speed clock counter. It contains the start bit, count value setting bit, and counter value capture bit.

2) Start/stop: When NKEN=1and timer/counter 0 low byte (T0L) is started.

3) Count clock: External input signals from pins P72/INT2/T0IN/NKIN.

4) Real-time output: The real-time output port must be set to the output mode.

When NKEN (bit 7) is set to 0, the real-time output port relinquishes its real-time output capability and synchronizes itself with the data in the port latch.

When the value that will result in NKEN=1 is written into NKREG, the real-time output port restores its real-time output capability and holds the output data. In this state, the contents of the port latch must be replaced by the next real-time output value.

When the high-speed clock counter keeps counting and reaches the count value "(T0LR+1) × 8 + value of NKCMP2 to NKCMP0," the real-time output turns to the required value. Subsequently, the real-time output port relinquishes the real-time output capability and output changes in synchronization with the data in the port latch. To restore the real-time output capability, a value that will result in NKEN=1 must be written into NKREG.

5) Capture clock: Generated in synchronization with the capture clock for T0L (timer 0 low byte).

3.7.3.2   P1TST register

1) The real-time output function is enabled when DSNKOT (P1TST register, bit 2) is set to 1.

2) The real-time output function is disabled when DSNKOT (P1TST register, bit 2) is set to 0. In this case, the real-time output pin functions as an ordinary port pin.

3.7.3.3 Timer/counter 0 operation

T0LEXT (T0CNT, bit4) must be set to 1 when a high-speed clock counter is to be used.

When NKEN=1 and T0LONG (T0CNT, bit5)=0, timer 0H runs in the normal mode and timer 0L is coupled with the high-speed clock counter to form an 11-bit free-running counter. When NKEN=1 and T0LONG (T0CNT, bit5)=1, timer 0 is coupled with the NK counter to form a 19-bit free-running counter.

When a free-running counter reaches the count value "(timer 0 match register value + 1) × 8 + value of NKCMP2 to NKCMP0," a match detection signal occurs, generating the real-time output of the required value and setting the match flag of timer 0. No new match signal is detected until the next NKREG write operation is performed.

The match data for these free-running counters must always be greater than the current counter value. When updating the match data, the match register for timer 0 must be set up before loading the match register for NKREG (NKCMP2 to NKCMP0) with data. Even if the same value is loaded, it must be written into NKREG to start a search for a match.

3.7.4 Related registers

3.7.4.1 High-speed clock counter control register (NKREG)

1) This register is an 8-bit register that controls the operation of the high-speed clock counter.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE7D	0000 0000	R/W	NKREG	NKEN	NKCMP2	NKCMP1	NKCMP0	NKCOV	NKCAP2	NKCAP1	NKCAP0

NKEN (bit 7): Counter control

When this bit is set to 0, the NK control circuit is inactive.

When this bit is set to 1, the NK control circuit is active. The timer 0 operation is switched to make an asynchronous high-speed counter with timer 0 being the higher-order counter. Counting is started by setting this bit to 1 and starting timer 0 in the external clock mode.

NKCMP2-NKCMP0 (bits 6-4): Match register

As soon as the counter reaches the value equivalent to "(timer 0 match register value+1) × 8 + value of NKCMP2 to NKCMP0 + 8," a match detection signal occurs, generating the real-time output of the required value and setting the timer 0 match flag. Subsequently, the real-time output port relinquishes the real-time output capability and changes its state in synchronization with the data in the port latch. The real-time output function and match detection function will not be resumed until the next NKREG write operation is performed.

NKCOV, NKCAP2-NKCAP0 (bits 3-0): Capture register

The NK counter value is captured into these bits in synchronization with the timer 0L capture operation.

NKCOV is a carry into timer 0. When this bit is set to 1, the capture value of timer 0 must be corrected by +1.

NKCAP2 to NKCAP0 carry the capture value of the NK counter. These bits are read only.
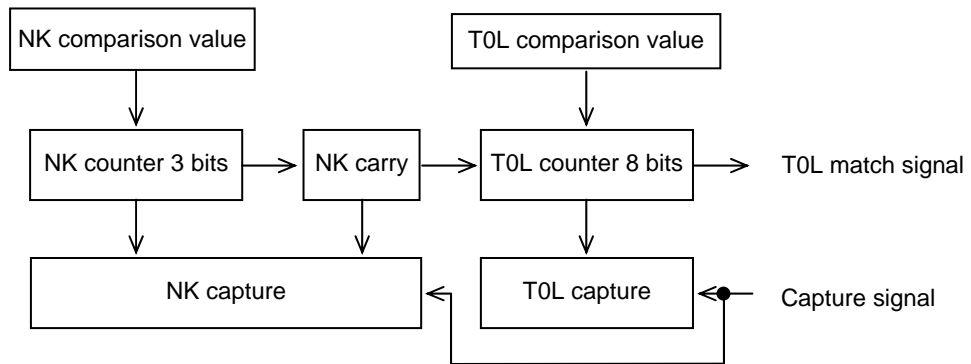
Figure 3.7.13   T0LONG = 0 (Timer 0: 8-bit mode) Block Diagram
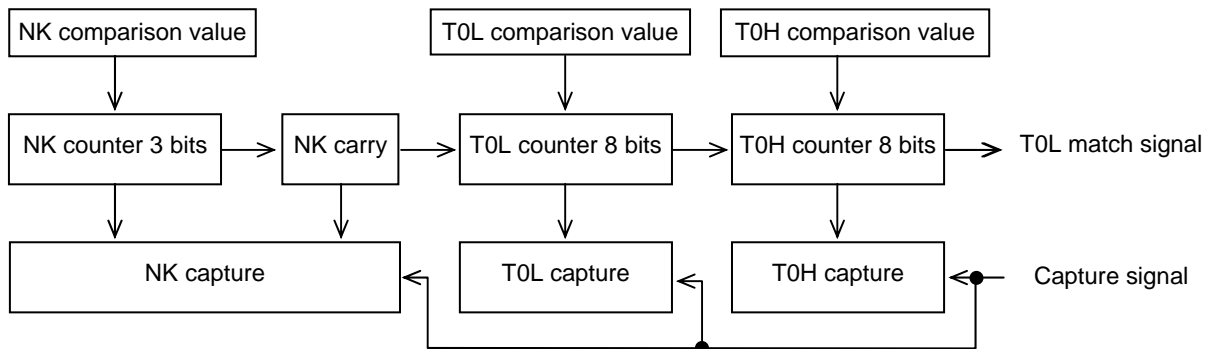
Figure 3.7.2   T0LONG = 1 (Timer 0: 16-bit mode) Block Diagram

3.8   Timer/Counter 1 (T1)

3.8.1   Overview

The timer/counter 1 (T1) incorporated in this series of microcontrollers is a 16-bit timer/counter that provides the following four functions:

1)   Mode 0: 8-bit programmable timer with an 8-bit prescaler (with toggle output) + 8-bit programmable timer/counter with an 8-bit prescaler (with toggle output)

2)   Mode 1: 8-bit PWM with an 8-bit prescaler × 2 channels

3)   Mode 2: 16-bit programmable timer/counter with an 8-bit prescaler (with toggle output) (the lower-order 8 bits may be used as a timer/counter with toggle output.)

4)   Mode 3: 16-bit programmable timer with an 8-bit prescaler (with toggle output) (the lower-order 8 bits may be used as a PWM.)

3.8.2   Functions

1)   Mode 0: 8-bit programmable timer with an 8-bit prescaler (with toggle output) + 8-bit programmable timer/counter with an 8-bit prescaler (with toggle output)

- T1L functions as an 8-bit programmable timer/counter that counts the number of signals obtained by dividing the cycle clock by 2 or external events, while T1H functions as an 8-bit programmable timer that counts the number of signals obtained by dividing the cycle clock by 2.

- T1PWML and T1PWMH generate a signal that toggles at the interval of T1L and T1H periods, respectively. (Note 1)

$$T1L \text{ period} = (T1LR + 1) \times (T1LPRC \text{ count}) \times 2Tcyc \quad \text{or}$$
$$(T1LR + 1) \times (T1LPRC \text{ count}) \text{ events detected}$$
$$T1PWML \text{ period} = T1L \text{ period} \times 2$$
$$T1H \text{ period} = (T1HR + 1) \times (T1HPRC \text{ count}) \times 2Tcyc$$
$$T1PWMH \text{ period} = T1H \text{ period} \times 2$$

$$Tcyc = \text{Period of cycle clock}$$

2)   Mode 1: 8-bit PWM with an 8-bit prescaler × 2 channels

- Two independent 8-bit PWMs (T1PWML and T1PWMH) run on the cycle clock.

$$T1PWML \text{ period} = 256 \times (T1LPRC \text{ count}) \times Tcyc$$
$$T1PWML \text{ low period} = (T1LR + 1) \times (T1LPRC \text{ count}) \times Tcyc$$
$$T1PWMH \text{ period} = 256 \times (T1HPRC \text{ count}) \times Tcyc$$
$$T1PWMH \text{ low period} = (T1HR + 1) \times (T1HPRC \text{ count}) \times Tcyc$$

3)   Mode 2: 16-bit programmable timer/counter with an 8-bit prescaler (with toggle output)
            (the lower-order 8 bits may be used as a timer/counter with toggle output.)

- A 16-bit programmable timer/counter runs that counts the number of signals whose frequency is equal to that of the cycle clock divided by 2 or the number of external events. Since interrupts can occur from the lower-order 8-bit timer (T1L) at the interval of T1L period, the lower-order 8 bits of this 16-bit programmable timer/counter can be used as the reference timer.

- T1PWML and T1PWMH generate a signal that toggles at the interval of T1L and T1 periods, respectively. (Note 1)

$$\text{T1L period} = (\text{T1LR} + 1) \times (\text{T1LPRC count}) \times 2\text{Tcyc}\quad\text{or}$$

$$(\text{T1LR} + 1) \times (\text{T1LPRC count})\text{ events detected}$$

$$\text{T1PWML period} = \text{T1L period} \times 2$$

$$\text{T1 period} = (\text{T1HR} + 1) \times (\text{T1HPRC count}) \times \text{T1L period}$$

$$\text{T1PWMH period} = \text{T1 period} \times 2$$

4)   Mode 3: 16-bit programmable timer with an 8-bit prescaler (with toggle output)
            (the lower-order 8 bits may be used as a PWM)

- A 16-bit programmable timer runs on the cycle clock.
- The lower-order 8 bits run as a PWM (T1PWML) having a period of 256 Tcyc.
- T1PWMH generates a signal that toggles at the interval of T1 period. (Note 1)

$$\text{T1PWML period} = 256 \times (\text{T1LPRC count}) \times \text{Tcyc}$$

$$\text{T1PWML low period} = (\text{T1LR} + 1) \times (\text{T1LPRC count}) \times \text{Tcyc}$$

$$\text{T1 period} = (\text{T1HR} + 1) \times (\text{T1HPRC count}) \times \text{T1PWML period}$$

$$\text{T1PWMH period} = \text{T1 period} \times 2$$

5)   Interrupt generation

T1L or T1H interrupt request is generated at the counter period of the T1L or T1H timer if the interrupt request enable bit is set.

6)   To control timer 1 (T1), it is necessary to manipulate the following special function registers.

- T1CNT, T1PRR, T1L, T1H, T1LR, T1HR
- P1, P1DDR, P1FCR
- P2, P2DDR, I45CR, I45SL

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE18	0000 0000	R/W	T1CNT	T1HRUN	T1LRUN	T1LONG	T1PWM	T1HCMP	T1HIE	T1LCMP	T1LIE
FE19	0000 0000	R/W	T1PRR	T1HPRE	T1HPRC2	T1HPRC1	T1HPRC0	T1LPRE	T1PRC2	T1LPRC1	T1LPRC0
FE1A	0000 0000	R	T1L	T1L7	T1L6	T1L5	T1L4	T1L3	T1L2	T1L1	T1L0
FE1B	0000 0000	R	T1H	T1H7	T1H6	T1H5	T1H4	T1H3	T1H2	T1H1	T1H0
FE1C	0000 0000	R/W	T1LR	T1LR7	T1LR6	T1LR5	T1LR4	T1LR3	T1LR2	T1LR1	T1LR0
FE1D	0000 0000	R/W	T1HR	T1HR7	T1HR6	T1HR5	T1HR4	T1HR3	T1HR2	T1HR1	T1HR0

Note 1: The output of the T1PWML is fixed at the high level if the T1L is stopped. If the T1L is running, the output of the T1PWML is fixed at the low level when T1LR = FFH. The output of T1PWMH is fixed at the high level if the T1H is stopped. If the T1H is running, the output of the T1PWMH is fixed at the low level when T1HR = FFH.

3.8.3   Circuit Configuration

3.8.3.1   Timer 1 control register (T1CNT)   (8-bit register)
1)   This register controls the operation and interrupts of the T1L and T1H.

3.8.3.2   Timer 1 prescaler control register (T1PRR)   (8-bit counter)
1)   This register sets the clocks for T1L and T1H.

3.8.3.3   Timer 1 prescaler low byte   (8-bit counter)
1)   Start/stop: The stop/start of timer 1 prescaler low byte is controlled by the 0/1 value of T1LRUN (timer 1 control register, bit 6).
2)   Count clock: Depends on the operation mode.

Mode	T1LONG	T1PWM	T1L Prescaler Count Clock
0	0	0	2 Tcyc/events (Note 2)
1	0	1	1 Tcyc (Note 3)
2	1	0	2 Tcyc/events (Note 2)
3	1	1	1 Tcyc (Note 3)

Note 2: T1L serves as an event counter when INT4 or INT5 is specified as the timer 1 count clock input in the external interrupt 4/5 pin select register (I45SL). It serves as a timer that runs using 2Tcyc as its count clock if neither INT4 nor INT5 is specified as the timer 1 count clock input.

Note 3: T1L will not run normally if INT4 or INT5 is specified as the timer 1 count clock input when T1PWM = 1. When T1PWM = 1, do not specify INT4 or INT5 as the timer 1 count clock input.

3)   Prescaler count: Determined by the T1PRR value.

The count clock for T1L is output at the intervals determined by the prescaler count.

T1LPRE	T1LPRC2	T1LPRC1	T1LPRC0	T1L Prescaler Count
0	–	–	–	1
1	0	0	0	2
1	0	0	1	4
1	0	1	0	8
1	0	1	1	16
1	1	0	0	32
1	1	0	1	64
1	1	1	0	128
1	1	1	1	256

4)   Reset: When the timer 1 stops operation or a T1L reset signal is generated.

3.8.3.4   Timer 1 prescaler high byte   (8-bit counter)

1) Start/stop: The stop/start of timer 1 prescaler high byte is controlled by the 0/1 value of T1HRUN (timer 1 control register, bit 7).

2) Count clock: Depends on the mode.

Mode	T1LONG	T1PWM	T1H Prescaler Count Clock
0	0	0	2 Tcyc
1	0	1	1 Tcyc
2	1	0	T1L match signal
3	1	1	$256 \times$ (T1LPRC count) $\times$ Tcyc

3) Prescaler count: Determined by the T1PRR value.

The count clock for T1H is output at the intervals determined by the prescaler count.

T1HPRE	T1HPRC2	T1HPRC1	T1HPRC0	T1H Prescaler Count
0	–	–	–	1
1	0	0	0	2
1	0	0	1	4
1	0	1	0	8
1	0	1	1	16
1	1	0	0	32
1	1	0	1	64
1	1	1	0	128
1	1	1	1	256

4) Reset: When the timer 1 stops operation or a T1H reset signal is generated.

3.8.3.5   Timer 1 low byte (T1L)   (8-bit counter)

1) Start/stop: The stop/start of the timer 1 low byte is controlled by the 0/1 value of T1LRUN (timer 1 control register, bit 6).

2) Count clock: T1L prescaler output clock.

3) Match signal: A match signal is generated when the count value matches the value of the match buffer register.

4) Reset: When the counter stops operation or a match signal occurs on the mode 0, or 2 condition.

3.8.3.6   Timer 1 high byte (T1H)   (8-bit counter)

1) Start/stop: The stop/start of the timer 1 high byte is controlled by the 0/1 value of T1HRUN (timer 1 control register, bit 7).

2) Count clock: T1H prescaler output clock

3) Match signal: A match signal is generated when the count value matches the value of the match buffer register.

4) Reset: When the counter stops operation or a match signal occurs on the mode 0, 2, or 3 condition.

3.8.3.7 Timer 1 match data register low byte (T1LR)   (8-bit register with a match buffer register)

1) This register is used to store the match data for T1L. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register coincides with that of timer 1 low byte (T1L)

2) The match buffer register is updated as follows:

   • When it is inactive (T1LRUN=0), the match register matches T1LR.
   • When it is active (T1LRUN=1), the match buffer register is loaded with the contents of T1LR when the value of T1L reaches 0.

3.8.3.8 Timer 1 match data register high byte (T1HR)   (8-bit register with a match buffer register)

1) This register is used to store the match data for T1H. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register coincides with that of timer 1 high byte (T1H).

2) The match buffer register is updated as follows:

   • When it is inactive (T1HRUN=0), the match register matches T1HR.
   • When it is active (T1HRUN=1), the match buffer register is loaded with the contents of T1HR when the value of T1H reaches 0.

3.8.3.9 Timer 1 low byte output (T1PWML)

1) The T1PWML output is fixed at the high level when T1L is inactive. If T1L is active, the T1PWML output is fixed at the low level when T1LR = FFH.

2) Timer 1 low byte output is a toggle output whose state changes on a T1L match signal when T1PWM (timer 1 control register, bit 4) is set to 0.

3) When T1PWM (timer 1 control register, bit 4) is set to 1, this PWM output is cleared on a T1L overflow and set on a T1L match signal.

3.8.3.10 Timer 1 high byte output (T1PWMH)

1) The T1PWMH output is fixed at the high level when T1H is inactive. If T1H is active, the T1PWMH output is fixed at the low level when T1HR = FFH.

2) The timer 1 high byte output is a toggle output whose state changes on a T1H match signal when T1PWM = 0 or T1LONG = 1.

3) When T1PWM = 1 and T1LONG = 0, this PWM output is cleared on a T1H overflow and set on a T1H match signal.
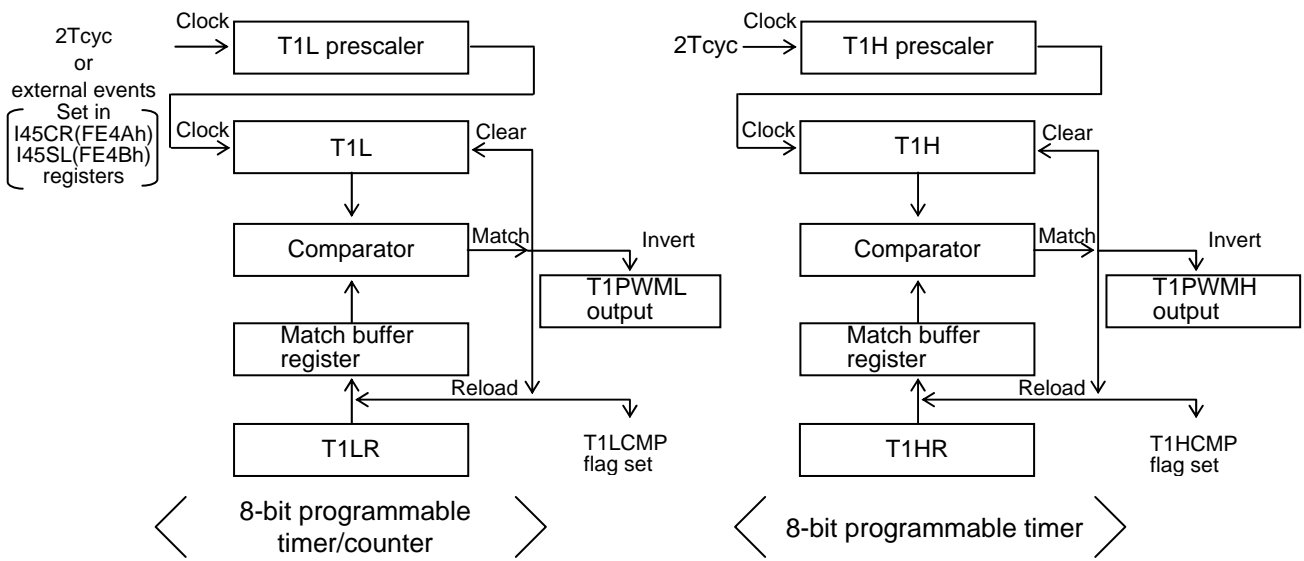
Figure. 3.8.1   Mode 0 Block Diagram (T1LONG = 0, T1PWM = 0)
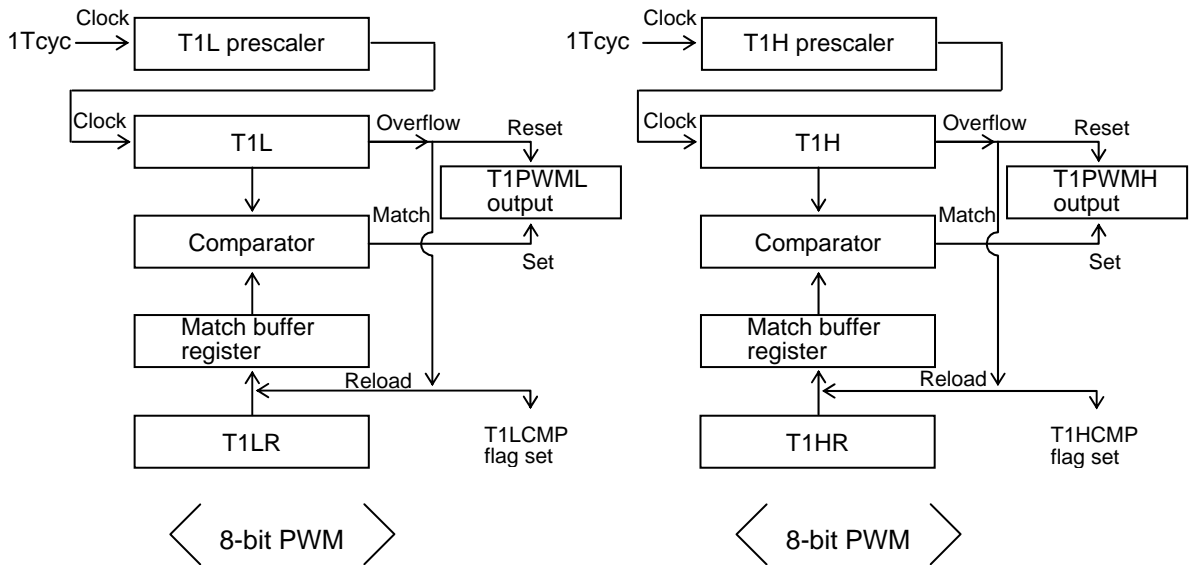
Figure. 3.8.2   Mode 1 Block Diagram (T1LONG = 0, T1PWM = 1)
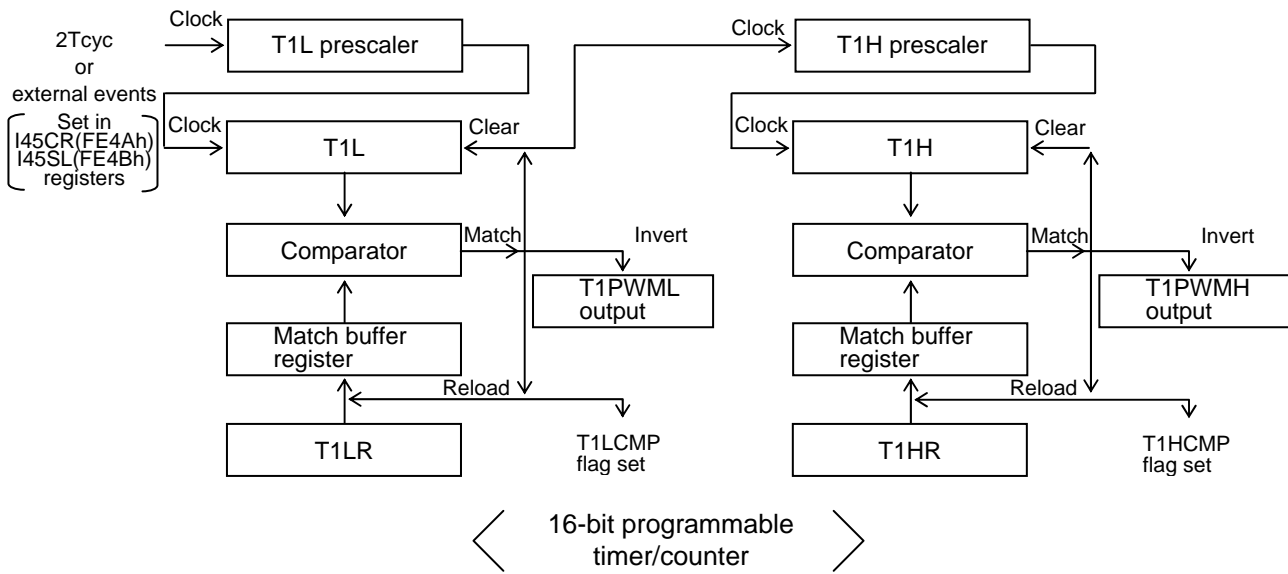
Fig. 3.8.3   Mode 2 Block Diagram (T1LONG = 1, T1PWM = 0)
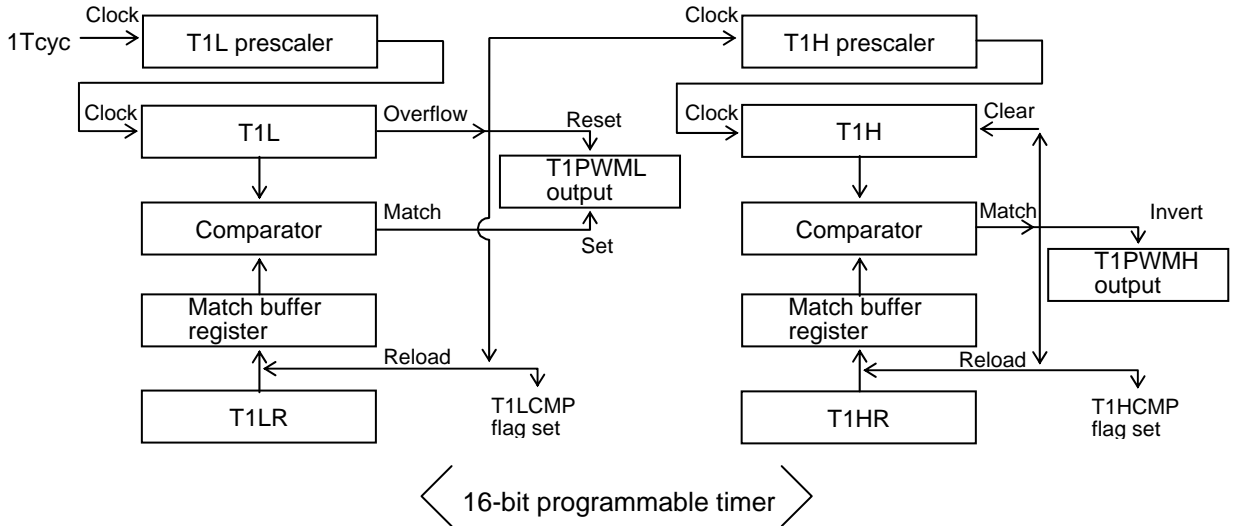
Fig. 3.8.4   Mode 3 Block Diagram (T1LONG = 1, T1PWM = 1)

3.8.4 Related Registers

3.8.4.1 Timer 1 control register (T1CNT)

1) Timer 1 control register is an 8-bit register that controls the operation and interrupts of T1L and T1H.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE18	0000 0000	R/W	T1CNT	T1HRUN	T1LRUN	T1LONG	T1PWM	T1HCMP	T1HIE	T1LCMP	T1LIE

T1HRUN (bit 7): T1H count control

When this bit is set to 0, timer 1 high byte (T1H) stops on a count value of 0. The match buffer register of T1H has the same value as that of T1HR.

When this bit is set to 1, timer 1 high byte (T1H) performs the required counting operation.

T1LRUN (bit 6): T1L count control

When this bit is set to 0, timer 1 low byte (T1L) stops on a count value of 0. The match buffer register of T1L has the same value as that of T1LR.

When this bit is set to 1, timer 1 low byte (T1L) performs the required counting operation.

T1LONG (bit 5): Timer 1 bit length select

When this bit is set to 0, timer 1 higher- and lower-order bytes serve as independent 8-bit timers.

When this bit is set to 1, timer 1 serves as a 16-bit timer since the timer 1 high byte (T1H) counts up at the interval of the timer 1 low byte (T1L).

Independent match signals are generated from T1H and T1L when their count value matches the contents of the corresponding match buffer register, regardless of the value of this bit.

T1PWM (bit 4): T1 output mode select

This bit and T1LONG (bit 5) determine the output mode of T1 (T1PWMH and T1PWML) as summarized in Table 3.8.1.

Table 3.8.1 Timer 1 Output (T1PWMH, T1PWML)

Mode	T1LONG	T1PWM	T1PWMH		T1PWML	
0	0	0	Toggle output	Period: $\{(T1HR+1) \times (T1HPRC\ count) \times 2Tcyc\} \times 2$	Toggle output	Period: $\{(T1LR+1) \times (T1LPRC\ count) \times 2Tcyc\} \times 2$
					or	Period: $\{(T1LR+1) \times (T1LPRC\ count) \times events\} \times 2$
1	0	1	PWM output	Period: $256 \times (T1HPRC\ count) \times Tcyc$	PWM output	Period: $256 \times (T1LPRC\ count) \times Tcyc$
2	1	0	Toggle output	Period: $\{(T1HR+1) \times (T1HPRC\ count) \times (T1LR+1) \times (T1LPRC\ count) \times 2\ Tcyc\} \times 2$	Toggle output	Period: $\{(T1LR+1) \times (T1LPRC\ count) \times 2Tcyc\} \times 2$
			or	Period: $\{(T1HR+1) \times (T1HPRC\ count) \times (T1LR+1) \times (T1LPRC\ count) \times events\} \times 2$	or	Period: $\{(T1LR+1) \times (T1LPRC\ count) \times events\} \times 2$
3	1	1	Toggle output	Period: $\{(T1HR+1) \times (T1HPRC\ count) \times 256 \times (T1LPRC\ count) \times Tcyc\} \times 2$	PWM output	Period: $256 \times (T1LPRC\ count) \times Tcyc$

T1HCMP (bit 3): T1H match flag

This flag is set if T1H reaches 0 when T1H is active (T1HRUN = 1).

This flag must be cleared with an instruction.

T1HIE (bit 2): T1H interrupt request enable control

An interrupt request is generated to vector address 002BH when this bit and T1HCMP are set to 1.

T1LCMP (bit 1): T1L match flag

This flag is set if T1L reaches 0 when T1L is active (T1LRUN = 1).

This flag must be cleared with an instruction.

T1LIE (bit 0): T1L interrupt request enable control

An interrupt request is generated to vector address 002BH when this bit and T1LCMP are set to 1.

Note:

• *T1HCMP and T1LCMP must be cleared to 0 with an instruction.*

3.8.4.2   Timer 1 prescaler control register (T1PRR)

1)   This register sets up the count values for the timer 1 prescaler.

2)   When the register value is changed while the timer is running, the change is reflected in the prescaler operation at the same timing when the match buffer register for the timer (T1L, T1H) is updated.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE19	0000 0000	R/W	T1PRR	T1HPRE	T1HPRC2	T1HPRC1	T1HPRC0	T1LPRE	T1LPRC2	T1LPRC1	T1LPRC0

T1HPRE (bit 7): This bit controls the timer 1 prescaler high byte.

T1HPRC2 (bit 6): This bit controls the timer 1 prescaler high byte.

T1HPRC1 (bit 5): This bit controls the timer 1 prescaler high byte.

T1HPRC0 (bit 4): This bit controls the timer 1 prescaler high byte.

T1HPRE	T1HPRC2	T1HPRC1	T1HPRC0	T1H Prescaler Count
0	–	–	–	1
1	0	0	0	2
1	0	0	1	4
1	0	1	0	8
1	0	1	1	16
1	1	0	0	32
1	1	0	1	64
1	1	1	0	128
1	1	1	1	256

T1LPRE (bit 3): This bit controls the timer 1 prescaler low byte.

T1LPRC2 (bit 2): This bit controls the timer 1 prescaler low byte.

T1LPRC1 (bit 1): This bit controls the timer 1 prescaler low byte.

T1LPRC0 (bit 0): This bit controls the timer 1 prescaler low byte.

T1LPRE	T1LPRC2	T1LPRC1	T1LPRC0	T1L Prescaler Count
0	–	–	–	1
1	0	0	0	2
1	0	0	1	4
1	0	1	0	8
1	0	1	1	16
1	1	0	0	32
1	1	0	1	64
1	1	1	0	128
1	1	1	1	256

3.8.4.3   Timer 1 low byte (T1L)

1)   This is a read-only 8-bit timer. It counts up on T1L prescaler output clock.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE1A	0000 0000	R	T1L	T1L7	T1L6	T1L5	T1L4	T1L3	T1L2	T1L1	T1L0

3.8.4.4   Timer 1 high byte (T1H)

1)   This is a read-only 8-bit timer. It counts up on T1H prescaler output clock.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE1B	0000 0000	R	T1H	T1H7	T1H6	T1H5	T1H4	T1H3	T1H2	T1H1	T1H0

3.8.4.5   Timer 1 match data register low byte (T1LR)

1)   This register is used to store the match data for T1L. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register coincides with the value of timer 1 low byte.

2)   Match buffer register is updated as follows:
   • When it is inactive (T1LRUN=0), the match register matches T1LR.
   • When it is active (T1LRUN=1), the match buffer register is loaded with the contents of T1LR when the value of T1L reaches 0.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE1C	0000 0000	R/W	T1LR	T1LR7	T1LR6	T1LR5	T1LR4	T1LR3	T1LR2	T1LR1	T1LR0

3.8.4.6   Timer 1 match data register high byte (T1HR)

1)   This register is used to store the match data for T1H. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register coincides with the value of timer 1 high byte.

2)   The match buffer register is updated as follows:
   • When it is inactive (T1HRUN=0), the match register matches T1HR.

   • When it is active (T1HRUN=1), the match buffer register is loaded with the contents of T1HR when the value of T1H reaches 0.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE1D	0000 0000	R/W	T1HR	T1HR7	T1HR6	T1HR5	T1HR4	T1HR3	T1HR2	T1HR1	T1HR0

T1

Modes 0, 2

Match buffer register value

T1L,T1H

Match signal

Interrupt flag set

T1PWML,T1PWMH

Mode 1

Counter value=FFH

Match

T1L、T1H

Match signal

Counter value=FFH

Interrupt flag set
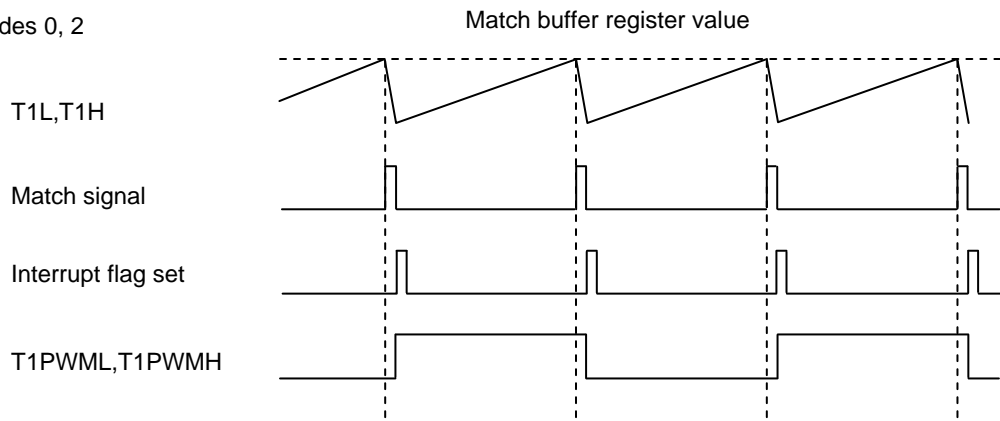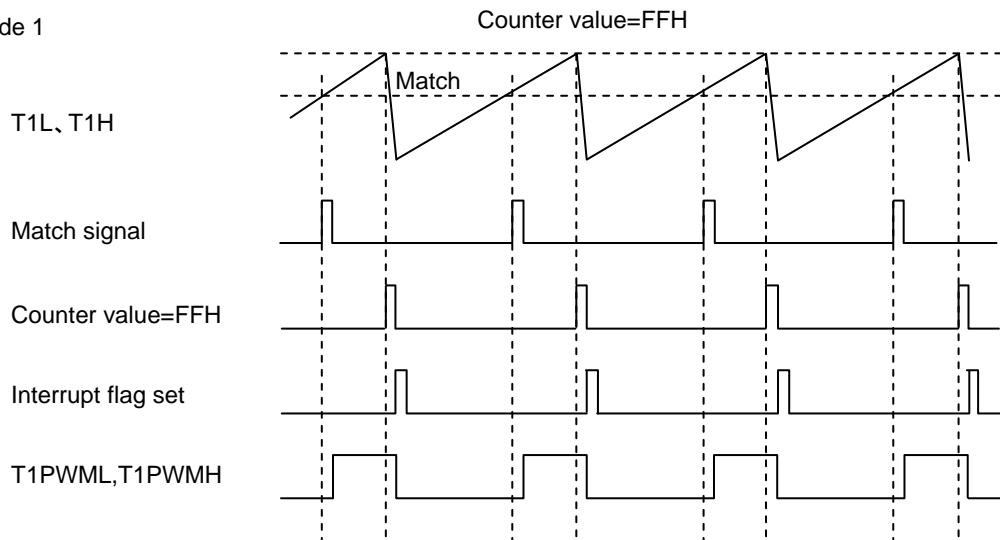
T1PWML,T1PWMH

Mode 3

Match buffer register value

T1H

Match signal

Interrupt flag set

T1PWMH
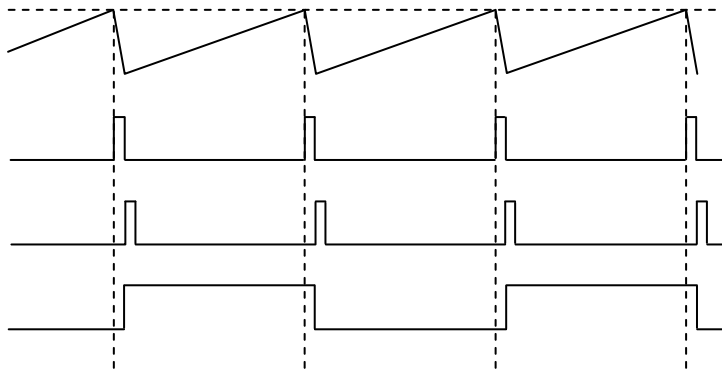
Counter value=FFH

T1L

Match

Match signal

Counter value=FFH

Interrupt flag set

T1PWML
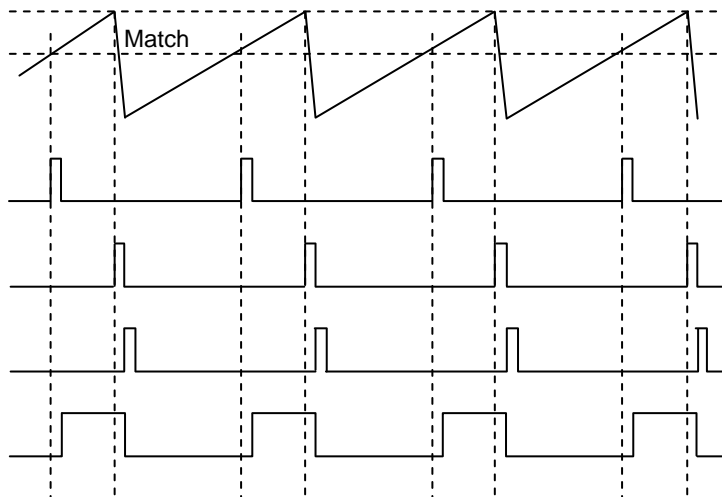
3.9 Timers 6 and 7 (T6, T7)

3.9.1 Overview

The timer 6 (T6) and timer 7 (T7) incorporated in this series of microcontrollers are 8-bit timers with two independently controlled 6-bit prescalers.

3.9.2 Functions

1) Timer 6 (T6)

Timer 6 is an 8-bit timer that runs on either 4Tcyc, 16Tcyc, or 64Tcyc clock. It can generate, at pin P06, a toggle waveform whose frequency is equal to the period of timer 6.

$$\text{T6 period} = (\text{T6R}+1) \times 4^{n}\text{Tcyc} \quad (n=1, 2, 3)$$

Tcyc = Period of cycle clock

2) Timer 7 (T7)

Timer 7 is an 8-bit timer that runs on either 4Tcyc, 16Tcyc, or 64Tcyc clock. It can generate, at pin P07, a toggle waveform whose frequency is equal to the period of timer 7.

$$\text{T7 period} = (\text{T7R}+1) \times 4^{n}\text{Tcyc} \quad (n=1, 2, 3)$$

Tcyc = Period of cycle clock

3) Interrupt generation

An interrupt requests to vector address 0043H is generated when the overflow flag is set at the interval of timer 6 or timer 7 period and the corresponding interrupt request enable bit is set.

4) To control the timer 6 (T6) and timer 7 (T7), it is necessary to manipulate the following special function registers.

- T67CNT, T6R, T7R
- P0, P0DDR, P0FCRU

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE78	0000 0000	R/W	T67CNT	T7C1	T7C0	T6C1	T6C0	T7OV	T7IE	T6OV	T6IE
FE7A	0000 0000	R/W	T6R	T6R7	T6R6	T6R5	T6R4	T6R3	T6R2	T6R1	T6R0
FE7B	0000 0000	R/W	T7R	T7R7	T7R6	T7R5	T7R4	T7R3	T7R2	T7R1	T7R0
FE42	0000 0000	R/W	P0FCRU	T7OE	T6OE	SCKOSL5	SCKOSL4	CLKOEN	CKODV2	CKODV1	CKODV0

3.9.3 Circuit Configuration

3.9.3.1 Timer 6/7 control register (T67CNT)   (8-bit register)

1) This register controls the operation and interrupts of T6 and T7.

3.9.3.2   Timer 6 counter (T6CTR)   (8-bit counter)

1)   This counter counts the number of clocks from the timer 6 prescaler (T6PR). The value of timer 6 counter (T6CTR) reaches 0 on the clock following the clock that brought about the value specified in the timer 6 period setting register (T6R), when the interrupt flag (T6OV) is set.

2)   When T6C0 and T6C1 (T67CNT:FE78, bits 4 and 5) are set to 0, the timer 6 counter stops at a count value of 0. In the other cases, the timer 6 counter continues operation.

3)   When data is written into T6R while timer 6 is running, both the timer 6 prescaler and counter are cleared and start counting again.

3.9.3.3   Timer 6 prescaler (T6PR)   (6-bit counter)

1)   This prescaler is used to define the clock period for the timer 6 determined by T6C0 and T6C1 (T67CNT: FE78, bits 4 and 5).

Table 3.9.1   Timer 6 Count Clocks

T6C1	T6C0	T6 Count Clock
0	0	Timer 6 prescaler and timer/counter are in the reset state.
0	1	4 Tcyc
1	0	16 Tcyc
1	1	64 Tcyc

3.9.3.4   Timer 6 period setting register (T6R)   (8-bit register)

1)   This register defines the period of timer 6.

2)   When data is written into T6R while timer 6 is running, both the timer 6 prescaler and counter are cleared and start counting again.

3.9.3.5   Timer 7 counter (T7CTR)   (8-bit counter)

1)   This counter counts the number of clocks from the timer 7 prescaler (T7PR). The value of timer 7 counter (T7CTR) reaches 0 on the clock following the clock that brought about the value specified in the timer 7 period setting register (T7R), when the interrupt flag (T7OV) is set.

2)   When T7C0 and T7C1 (T67CNT:FE78, bits 6 and 7) are set to 0, the timer 7 counter stops at a count value of 0. In the other cases, the timer 7 counter continues operation.

3)   When data is written into T7R while timer 7 is running, both the timer 7 prescaler and counter are cleared and start counting again.

3.9.3.6   Timer 7 prescaler (T7PR)   (6-bit counter)

1)   This prescaler is used to define the clock period for the timer 7 determined by T7C0 and T7C1 (T67CNT:FE78, bits 6 and 7).

Table 3.9.2   Timer 7 Count Clocks

T7C1	T7C0	T7 Count Clock
0	0	Timer 7 prescaler and timer/counter are in the reset state.
0	1	4 Tcyc
1	0	16 Tcyc
1	1	64 Tcyc

3.9.3.7   Timer 7 period setting register (T7R)   (8-bit register)

1)   This register defines the period of timer 7.

2)   When data is written into T7R while timer 7 is running, both the timer 7 prescaler and counter are cleared and start counting again.

Timer 6 period setting
register T6R (FE7Ah)

Timer 6/7 control register
T67CNT (FE78h)

bit5	bit4

Comparator

T6 overflow
(T6R + 1) × count clock

Set prescaler
count

1 Tcyc — Clock → Timer 6 prescaler
(T6PR) — Clock → Timer 6 counter
(T6CTR)

Clear

Clear

Timer 6/7 control register
T67CNT (FE78h)

Set

bit3	bit2	bit1	bit0

Set

T6 interrupt

T7 interrupt

Clear

Clear

1 Tcyc — Clock → Timer 7 prescaler
(T7PR) — Clock → Timer 7 counter
(T7CTR)

Set prescaler
count

bit7	bit6

Timer 6/7 control register
T67CNT (FE78h)

Comparator

T7 overflow
(T7R +1) × count clock

Timer 7 period setting
register T7R (FE7Bh)

Figure 3.9.1   Timer 6/7 Block Diagram

3.9.4   Related Registers

3.9.4.1   Timer 6/7 control register (T67CNT)

1)   This register is an 8-bit register that controls the operation and interrupts of T6 and T7.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE78	0000 0000	R/W	T67CNT	T7C1	T7C0	T6C1	T6C0	T7OV	T7IE	T6OV	T6IE

T7C1 (bit 7): T7 count clock control

T7C0 (bit 6): T7 count clock control

T7C1	T7C0	T7 Count Clock
0	0	Timer 7 prescaler and timer/counter are stopped in the reset state.
0	1	4 Tcyc
1	0	16 Tcyc
1	1	64 Tcyc

T6C1 (bit 5): T6 count clock control

T6C0 (bit 4): T6 count clock control

T6C1	T6C0	T6 Count Clock
0	0	Timer 6 prescaler and timer/counter are stopped in the reset state.
0	1	4 Tcyc
1	0	16 Tcyc
1	1	64 Tcyc

T7OV (bit 3): T7 overflow flag

This flag is set at the interval of timer 7 period when timer 7 is running.

This flag must be cleared with an instruction.

T7IE (bit 2): T7 interrupt request enable control

An interrupt request to vector address 0043H is generated when this bit and T7OV are set to 1.

T6OV (bit 1): T6 overflow flag

This flag is set at the interval of timer 6 period when timer 6 is running.

This flag must be cleared with an instruction.

T6IE (bit 0): T6 interrupt request enable control

An interrupt request to vector address 0043H is generated when this bit and T6OV are set to 1.

3.9.4.2   Timer 6 period setting register (T6R)

1)   This register is an 8-bit register for defining the period of timer 6.

Timer 6 period = (T6R value+1) × Timer 6 prescaler value (4, 16 or 64 Tcyc)

2)   When data is written into T6R while timer 6 is running, both the timer 6 prescaler and counter are cleared and start counting again.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE7A	0000 0000	R/W	T6R	T6R7	T6R6	T6R5	T6R4	T6R3	T6R2	T6R1	T6R0

3.9.4.3   Timer 7 period setting register (T7R)

1)   This register is an 8-bit register for defining the period of timer 7.

Timer 7 period = (T7R value+1) × Timer 7 prescaler value (4, 16 or 64 Tcyc)

2)   When data is written into T7R while timer 7 is running, both the timer 7 prescaler and counter are cleared and start counting again.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE7B	0000 0000	R/W	T7R	T7R7	T7R6	T7R5	T7R4	T7R3	T7R2	T7R1	T7R0

3.9.4.4   Port 0 function control register (P0FCRU)

1)   This 8-bit register controls the port 0 multiplexed output pin. It controls the timer 6 and timer 7 toggle output.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE42	0000 0000	R/W	P0FCRU	T7OE	T6OE	SCKOSL5	SCKOSL4	CLKOEN	CKODV2	CKODV1	CKODV0

T7OE (bit 7):

This bit is used to control the timer 7 toggle output at pin P07.

This bit is disabled when pin P07 is set in the input mode.

When pin P07 is set in the output mode:

A 0 in this bit causes the value of port data latch to be presented at pin P07.

A 1 in this bit causes the OR of the value of the port data latch and the waveform which toggles at the interval equal to the timer 7 period at pin P07.

T6OE (bit 6):

This bit is used to control the timer 6 toggle output at pin P06.

This bit is disabled when pin P06 is set in the input mode.

When pin P06 is set in the output mode:

A 0 in this bit causes the value of port data latch to be presented at pin P06.

A 1 in this bit causes the OR of the value of the port data latch and the waveform which toggles at the interval equal to the timer 6 period at pin P06.

SCKOSL5 (bit 5):
SCKOSL4 (bit 4):

The above two bits have no bearing on the control of timers 6 and 7. See the description of port 0 for details on these bits.

CLKOEN (bit 3):
CKODV2 (bit 2):

CKODV1 (bit 1):
CKODV0 (bit 0):

The above four bits have no bearing on the control of timers 6 and 7. See the description of port 0 for details on these bits.

3.10   Base Timer (BT)

3.10.1   Overview

The base timer (BT) incorporated in this series of microcontrollers is a 14-bit binary up-counter that provides the following five functions:

1)   Clock timer

2)   14-bit binary up-counter

3)   High-speed mode (when used as a 6-bit base timer)

4)   Buzzer output

5)   Hold mode release

3.10.2   Functions

1)   Clock timer

The base timer can count clocks at 0.5 second intervals when a 32.768 kHz subclock is used as the count clock for the base timer. In this case, one of the three clocks, namely, "cycle clock," "timer/counter 0 prescaler output," and "subclock," must be loaded in the input signal select register (ISL) as the base timer count clock.

2)   14-bit binary up-counter

A 14-bit binary up-counter can be constructed using an 8-bit binary up-counter and a 6-bit binary up-counter. These counters can be cleared under program control.

3)   High-speed mode (when used as a 6-bit base timer)

When the base timer is used as a 6-bit timer, it can clock at intervals of approximately 2 ms if the 32.768 kHz subclock is used as the count clock. The bit length of the base timer can be specified using the base timer control register (BTCR).

4)   Buzzer output function

The base timer can generate 2kHz beeps when the 32.768 kHz subclock is used as the count clock. The buzzer output can be controlled using the input signal select register (ISL). The buzzer output is ANDed with the PWMH output from timer 1 and can be transmitted via pin P17.

5)   Interrupt generation

An interrupt request to vector address 001BH is generated if an interrupt request is generated by the base timer when the interrupt request enable bit is set. The base timer can generate two types of interrupt requests: "base timer interrupt 0" and "base timer interrupt 1."

6)   HOLD mode operation and HOLD mode releasing

The base timer is enabled for operation in HOLD mode when bit 2 of the power control register (PCON) is set. HOLD mode can be released by an interrupt from the base timer. This function allows the microcontroller to perform low-current intermittent operations.

7) To control the base timer, it is necessary to manipulate the following special function registers.
   - BTCR, ISL
   - P1, P1DDR, P1FCR

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE7F	0000 0000	R/W	BTCR	BTFST	BTON	BTC11	BTC10	BTIF1	BTIE1	BTIF0	BTIE0
FE5F	0000 0000	R/W	ISL	ST0HCP	ST0LCP	BTIMC1	BTIMC0	BUZON	NFSEL	NFON	ST0IN

3.10.3 Circuit Configuration

3.10.3.1 8-bit binary up-counter
1) This counter is an up-counter that receives, as its input, the signal selected by the input signal select register (ISL). It generates 2 kHz buzzer output and base timer interrupt 1 flag set signals. The overflow out of this counter serves as the clock to the 6-bit binary counter.

3.10.3.2 6-bit binary up-counter
1) This counter is a 6-bit up-counter that receives, as its input, the signal selected by the input signal select register (ISL) or the overflow signal from the 8-bit counter and generates set signals for base timer interrupts 0 and 1. The switching of the input clock is accomplished by the base timer control register (BTCR).

3.10.3.3 Base timer input clock source
1) The clock input to the base timer (fBST) can be selected from "cycle clock," "timer 0 prescaler," and "subclock" via the input signal select register (ISL).
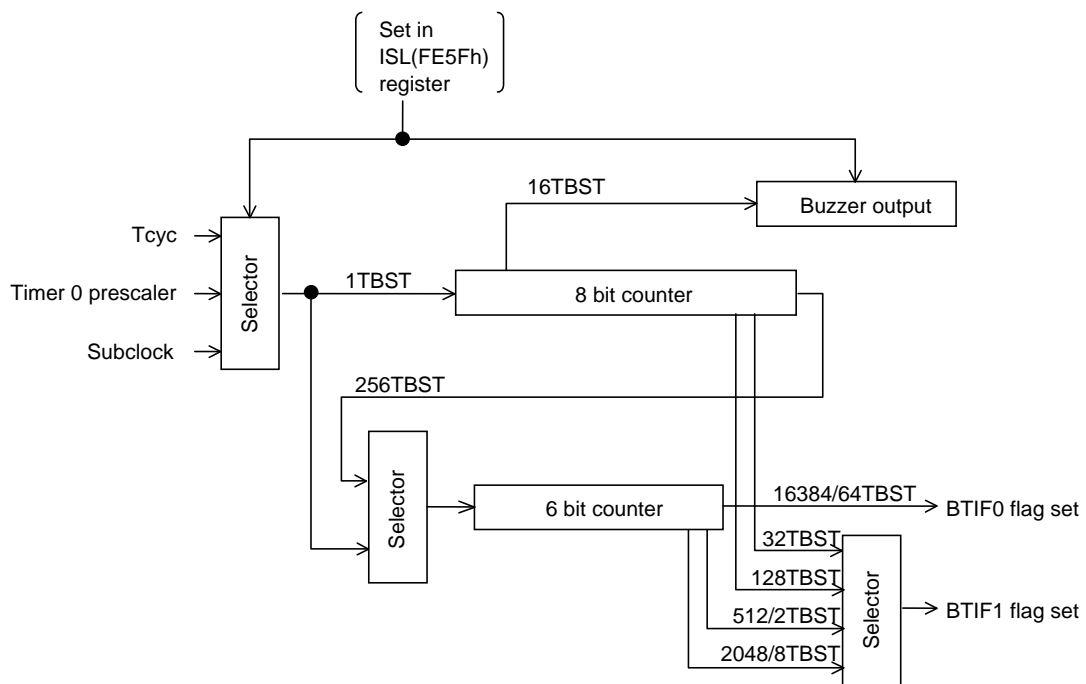
Figure 3.10.1   Base Timer Block Diagram

3.10.4   Related Registers

3.10.4.1   Base timer control register (BTCR)

1)   The register is an 8-bit register that controls the operation of the base timer.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE7F	0000 0000	R/W	BTCR	BTFST	BTON	BTC11	BTC10	BTIF1	BTIE1	BTIF0	BTIE0

BTFST (bit 7): Base timer interrupt 0 period control

This bit is used to select the interval at which base timer interrupt 0 is to occur. If it is set to 1, the base timer interrupt 0 flag is set when an overflow occurs in the 6-bit counter. The interval at which overflows occur is 64TBST.

If this bit is set to 0, the base timer interrupt 0 flag is set when an overflow occurs in the 14-bit counter. The interval at which overflows occur is 16384TBST.

This bit must be set to 1 when the high-speed mode is to be used.

TBST: Is the period of the input clock to the base timer that is selected by the input signal select register (ISL), bits 4 and 5.

BTON (bit 6): Base timer operation control

When this bit is set to 0, the base timer stops when a count value of 0 is reached.

When this bit is set to 1, the base timer continues operation.

BTC11 (bit 5): Base timer interrupt 1 period control
BTC10 (bit 4): Base timer interrupt 1 period control

BTFST	BTC11	BTC10	Base Timer Interrupt 0 Period	Base Timer Interrupt 1 Period
0	0	0	16384TBST	32TBST
1	0	0	64TBST	32TBST
0	0	1	16384TBST	128TBST
1	0	1	64TBST	128TBST
0	1	0	16384TBST	512TBST
0	1	1	16384TBST	2048TBST
1	1	0	64TBST	2TBST
1	1	1	64TBST	8TBST

BTIF1 (bit 3): Base timer interrupt 1 flag

This flag is set at the interval equal to the base timer interrupt 1 period that is defined by BTFST, BTC11, and BTC10.

This flag must be cleared with an instruction.

BTIE1 (bit 2): Base timer interrupt 1 request enable control

Setting this bit and BTIF1 to 1 generates "X'tal HOLD mode release signal" and "interrupt request to vector address 001BH".

BTIF0 (bit 1): Base timer interrupt 0 flag

This flag is set at the interval equal to the base timer interrupt 0 period that is defined by BTFST, BTC11, and BTC10.

This flag must be cleared with an instruction.

BTIE0 (bit 0): Base timer interrupt 0 request enable control

Setting this bit and BTIF0 to 1 generates "X'tal HOLD mode release signal" and "interrupt request to vector address 001BH".

Notes:

- *Set up the conditions under which the flags (BTIF1 and BTIF0) are set at intervals of the base timer interrupt cycle such that the period of the cycle clock (Tcyc) and the base timer interrupt period satisfy the following relationship:*

  Period of cycle clock (Tcyc) $<$ Base timer interrupt period$\div 2$

  In fact, since program processing (e.g. interrupt processing routine) intervenes in execution, however, the time taken by such processing should be taken into consideration when configuring the interrupt period.

- *Note that there are cases in which BTIF1 is set to 1 if BTC11 or BTC10 is rewritten when the base timer is active.*

- *If HOLD mode is entered while running the base timer when the cycle clock or subclock is selected as the base timer clock source, the base timer is subject to the influence of unstable oscillations caused by the main clock and subclock when they are started following the releasing of HOLD mode, resulting in an erroneous count from the base timer. When entering HOLD mode, therefore, it is recommended that the base timer be stopped.*

- *This series of microcontrollers supports X'tal HOLD mode that enables low-current intermittent operation. In this mode, only the base timer is allowed for operation.*

3.10.4.2 Input signal select register (ISL)

1) This register is an 8-bit register that controls the timer 0 input, noise filter time constant, buzzer output, and base timer clock.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5F	0000 0000	R/W	ISL	ST0HCP	ST0LCP	BTIMC1	BTIMC0	BUZON	NFSEL	NFON	ST0IN

ST0HCP (bit 7): Timer 0H capture signal input port select

ST0LCP (bit 6): Timer 0L capture signal input port select

These 2 bits have nothing to do with the control function on the base timer.

BTIMC1 (bit 5): Base timer clock select

BTIMC0 (bit 4): Base timer clock select

BTIMC1	BTIMC0	Base Timer Input Clock
0	0	Subclock
0	1	Cycle clock
1	0	Subclock
1	1	Timer/counter 0 prescaler output

BUZON (bit 3): Buzzer output select

This bit enables the buzzer output ($\frac{fBST}{16}$).

When this bit is set to 1, a signal that is obtained by dividing the base timer clock by 16 is sent to port P17 as buzzer output.

When this bit is set to 0, the buzzer output is held high.

fBST: Is the period of the input clock to the base timer that is selected by the input signal select register (ISL), bits 4 and 5.

NFSEL (bit 2): Noise filter time constant select

NFON (bit 1): Noise filter time constant select

ST0IN (bit 0): Timer 0 count clock input port select

These 3 bits have nothing to do with the control function of the base timer.

3.11 Serial Interface 0 (SIO0)

3.11.1 Overview

The serial interface SIO0 incorporated in this series of microcontrollers has the following two major functions:

1) Synchronous 8-bit serial I/O (2- or 3-wire system, transfer rates of $\frac{4}{3}$ to $\frac{512}{3}$ Tcyc)
2) Continuous data transmission/reception (transfer of data whose length varies between 1 and 256 bits in 1-bit units, transfer rates of $\frac{4}{3}$ to $\frac{512}{3}$ Tcyc)

3.11.2 Functions

1) Synchronous 8-bit serial I/O

   - Performs 2- or 3-wire synchronous serial communication. The clock may be an internal or external clock.

   - The clock rate of the internal clock is programmable within the range of $(n+1) \times \frac{2}{3}$ Tcyc (n = 1 to 255; Note: n = 0 is inhibited).

2) Continuous data transmission/reception

   - Transmits and receives bit streams whose length is variable in 1-bit units between 1 and 256 bits. Transfer is carried out in the clock synchronization mode. Either internal or external clock can be used.
     It allows suspension and resumption of data transfer in byte units.

   - The clock rate of the internal clock is programmable within the range of $(n+1) \times \frac{2}{3}$ Tcyc (n= 1 to 255; Note: n = 0 is inhibited).

   - 1 to 256 bits of send data is automatically transferred from RAM to the data shift register (SBUF0) and receive data is automatically transferred from the data shift register (SBUF0) to RAM.

3) Interrupt generation

   An interrupt request is generated at the end of communication when the interrupt request enable bit is set.

4) To control serial interface 0 (SIO0), it is necessary to manipulate the following special function registers.

   - SCON0, SBUF0, SBR0, SCTR0, SWCON0
   - P1, P1DDR, P1FCR

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE30	0000 0000	R/W	SCON0	SI0BNK	SI0WRT	SI0RUN	SI0CTR	SI0DIR	SI0OVR	SI0END	SI0IE
FE31	0000 0000	R/W	SBUF0	SBUF07	SBUF06	SBUF05	SBUF04	SBUF03	SBUF02	SBUF01	SBUF00
FE32	0000 0000	R/W	SBR0	SBRG07	SBRG06	SBRG05	SBRG04	SBRG03	SBRG02	SBRG01	SBRG00
FE33	0000 0000	R/W	SCTR0	SCTR07	SCTR06	SCTR05	SCTR04	SCTR03	SCTR02	SCTR01	SCTR00
FE37	0000 0000	R/W	SWCON0	S0WSTP	SWCONB6	SWCONB5	S0XBYT4	S0XBYT3	S0XBYT2	S0XBYT1	S0XBYT0

3.11.3   Circuit Configuration

3.11.3.1   SIO0 control register (SCON0)   (8-bit register)

   1)   This register controls the operation and interrupts of SIO0.

3.11.3.2   SIO0 data shift register (SBUF0)   (8-bit reload counter)

   1)   This register is an 8-bit shift register that performs data input and output operations at the same time.

3.11.3.3   SIO0 baudrate generator (SBR0)   (8-bit register)

   1)   This register is an 8-bit register that defines the transfer rate for SIO0 serial transfer.

   2)   It can generate clocks at intervals of $(n+1) \times \frac{2}{3}$ Tcyc (n = 1 to 255; Note: n = 0 is inhibited).

3.11.3.4   Continuous data bit register (SCTR0)   (8-bit register)

   1)   This register controls the bit length of data to be transmitted or received in the continuous data transmission/reception mode.

3.11.3.5   Continuous data transfer control register (SWCON0)   (8-bit register)

   1)   This register controls the suspension and resumption of serial transfer in byte units in the continuous data transmission /reception mode.

   2)   It allows the application program to read the number of bytes transferred in the continuous data transmission/reception mode.
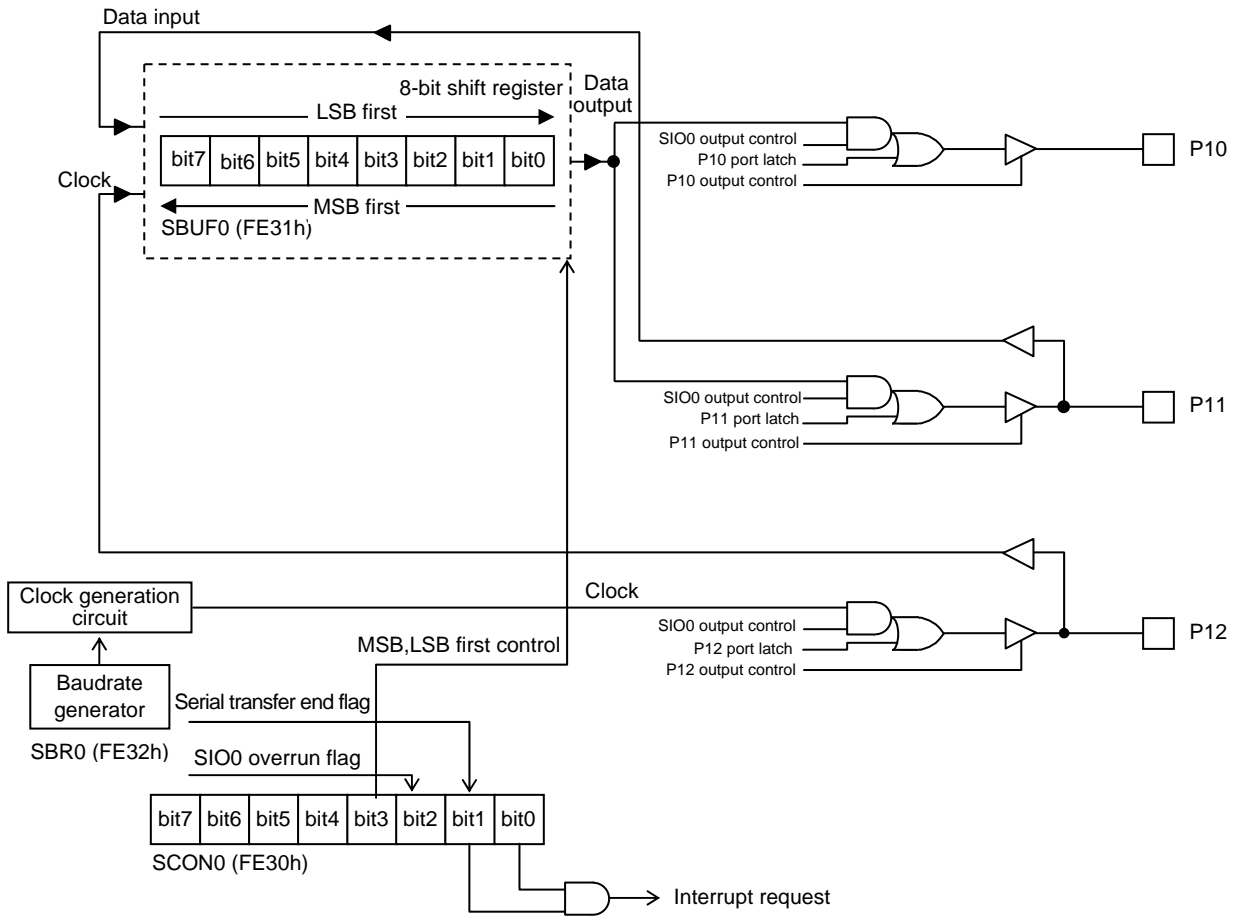
SIO0

Figure 3.11.1   SIO0 Synchronous 8-bit Serial I/O Block Diagram (SI0CTR=0)
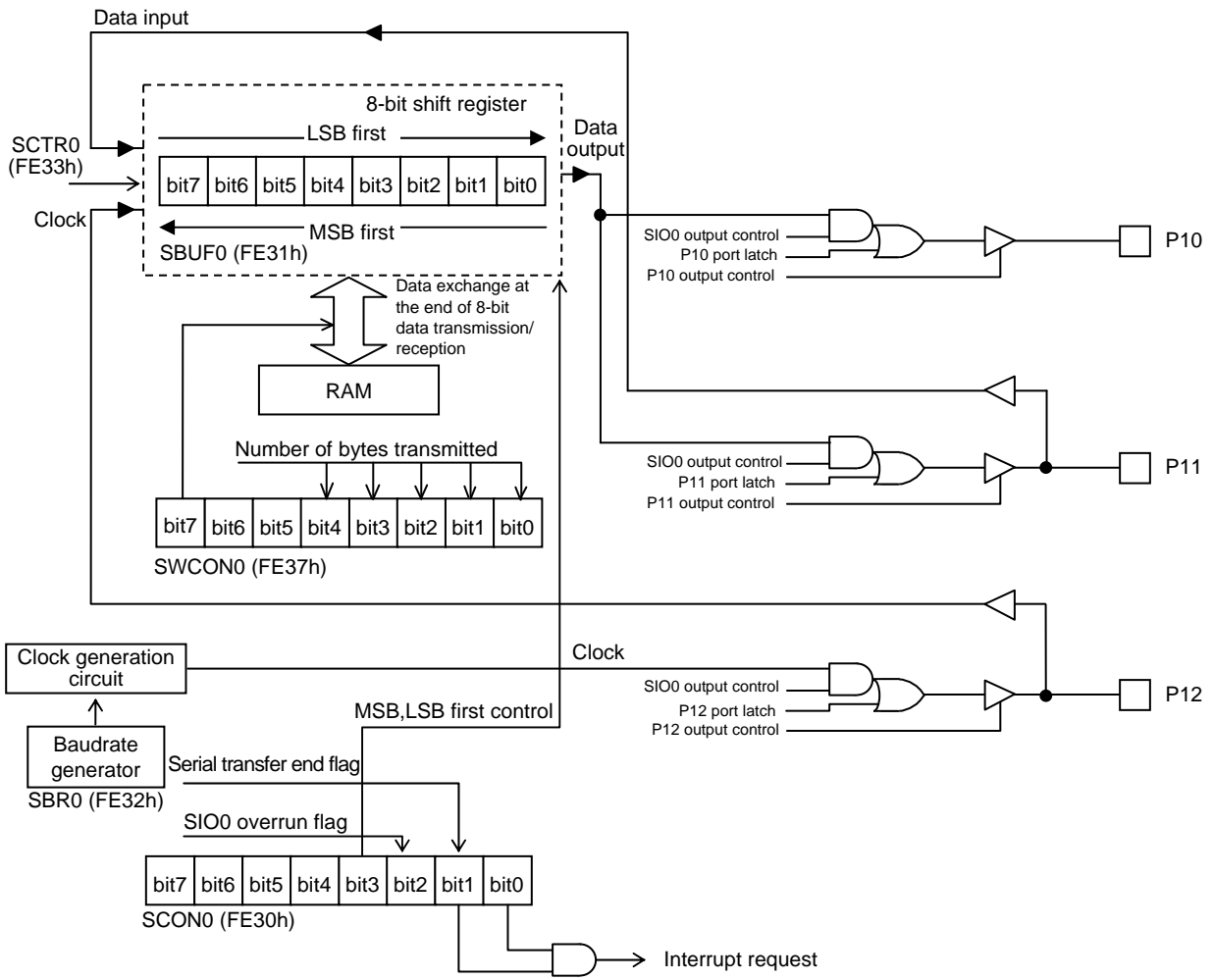
Figure 3.11.2   SIO0 Continuous Data Transmission/Reception Mode Block Diagram (SI0CTR=1)

3.11.4   Related Registers

3.11.4.1   SIO0 control register (SCON0)
1)   This register is an 8-bit register that controls the operation and interrupts of SIO0.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE30	0000 0000	R/W	SCON0	SI0BNK	SI0WRT	SI0RUN	SI0CTR	SI0DIR	SI0OVR	SI0END	SI0IE

SI0BNK (bit 7): Transfer RAM address control during continuous data transmission/reception
1)   When this bit is set to 1, transfer of continuous transmission/reception data is carried out between RAM addresses (01E0[H] to 01FF[H]) and SBUF0.
2)   When this bit is set to 0, transfer of continuous transmission/reception data is carried out between RAM addresses (01C0[H] to 01DF[H]) and SBUF0.

SI0WRT (bit 6): RAM write control during continuous data transmission/reception
1)   When this bit is set to 1, the contents of data RAM and SBUF0 are automatically exchanged during continuous data transmission/reception.
2)   When this bit is set to 0, the contents of data RAM are automatically transferred to SBUF0 during continuous data transmission/reception, but the contents of data RAM remain unchanged.

SI0RUN (bit 5): SIO0 operation flag
1)   A 1 in this bit indicates that SIO0 is running.
2)   This bit must be set with an instruction.
3)   This bit is automatically cleared at the end of serial transfer (on the rising edge of the last clock involved in the transfer).

SI0CTR (bit 4): SIO0 continuous data transmission-reception/synchronous 8-bit control
1)   A 1 in this bit places SIO0 into the continuous data transmission/reception mode.
2)   A 0 in this bit places SIO0 into the synchronous 8-bit mode.
3)   This bit is automatically cleared at the end of serial transfer (on the rising edge of the last clock involved in the transfer).

SI0DIR (bit 3): MSB/LSB first select
1)   A 1 in this bit places SIO0 into the MSB first mode.
2)   A 0 in this bit places SIO0 into the LSB first mode.

SI0OVR (bit 2): SIO0 overrun flag
1)   This bit is set when a falling edge of the input clock is detected with SI0RUN=0.
2)   This bit is set when a falling edge of the input clock is detected during internal data communication between SBUF0 and RAM with each 8-bit transfer in continuous data transmission/reception.
3)   Read this bit and judge if the communication is performed normally at the end of the communication.
4)   This bit must be cleared with an instruction.

SI0END (bit 1): End of serial transfer flag
1)   This bit is set at the end of serial transfer (on the rising edge of the last clock involved in the transfer).
2)   This bit must be cleared with an instruction.

SI0IE (bit 0): SI00 interrupt request enable control
1)   When this bit and SI0END are set to 1, an interrupt request to vector address 0033H is generated.

3.11.4.2 SIO0 data shift register (SBUF0)

1) This register is an 8-bit shift register for serial transfer.

2) Data to be transmitted/received is written to and read from this shift register directly.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE31	0000 0000	R/W	SBUF0	SBUF07	SBUF06	SBUF05	SBUF04	SBUF03	SBUF02	SBUF01	SBUF00

3.11.4.3 Baudrate generator register (SBR0)

1) This register is an 8-bit register that defines the transfer rate of SIO0 serial transfer.

2) The transfer rate is computed as follows:

$$\text{TSBR0} = (\text{SBR0 value} + 1) \times \frac{2}{3} \text{Tcyc}$$

SBR0 can take a value from 1 to 255 and the valid value range of TSBR0 is from $\frac{4}{3}$ to $\frac{512}{3}$ Tcyc.

* The SBR0 value of 00[H] is prohibited.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE32	0000 0000	R/W	SBR0	SBRG07	SBRG06	SBRG05	SBRG04	SBRG03	SBRG02	SBRG01	SBRG00

3.11.4.4 Continuous data bit register (SCTR0)

1) This register is used to specify the bit length of serial data to be transmitted/received through SIO0 in the continuous data transmission/reception mode.

2) The valid value range is from 00[H] to FF[H].

3) When continuous data transmission/reception is started with this register set to 00[H], 1 bit of data transmission/reception is carried out after the contents of data RAM is transferred to SBUF0 (after the contents of RAM and SBUF0 are exchanged when SI0WRT = 1) (Number of bits transferred = SCTR0 value + 1).

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE33	0000 0000	R/W	SCTR0	SCTR07	SCTR06	SCTR05	SCTR04	SCTR03	SCTR02	SCTR01	SCTR00

3.11.4.5 Continuous data transfer control register (SWCON0)

1) This register is used to suspend or resume the operation of SIO0 in byte units in the continuous data transmission/reception mode and to read the number of transferred bytes (bits 4 to 0 are read only).

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE37	0000 0000	R/W	SWCON0	S0WSTP	SWCONB6	SWCONB5	S0XBYT4	S0XBYT3	S0XBYT2	S0XBYT1	S0XBYT0

S0WSTP (bit 7):

When this bit is set to 1, SIO0 stops operation after completing the transfer of 1 byte data in the continuous transfer mode (1 byte of serial data separated at the beginning of serial transfer). Serial transfer resumes when this bit is subsequently set to 0.

SWCONB6, SWCONB5 (bits 6 and 5):

These bits can be read and written with instructions. The user can use these bits freely.

S0XBYT4-S0XBYT0 (bits 4 to 0):

These bits can be read to determine the number of bytes transferred in the continuous data transfer mode.

3.11.4.6   RAM used in continuous data transmission/reception mode

SIO0 can transmit and receive 1 to 256 bits of serial data in the continuous data transmission/reception mode, using the RAM area from 01C0[H] to 01FF[H].

1)   The RAM area from 01C0[H] to 01DF[H] is used when SI0BNK=0.

2)   The RAM area from 01E0[H] to 01FF[H] is used when SI0BNK=1.

3)   In the continuous data transmission/reception mode, data transmission/reception is started after the operation flag is set and RAM data at the lowest address is transferred to SBUF0 (after the contents of RAM and SBUF0 are exchanged when SI0WRT=1). After 8 bits of data is transmitted and received, the RAM data from the next RAM address is transferred to SBUF0 (the contents of RAM and SBUF0 are exchanged when SI0WRT=1) and data transmission/reception processing is continued. The last 8 bits or less of received data are left in SBUF0 and not exchanged with data in RAM. If the volume of data to transmit/receive is set to 8 bits or less, after the operation flag is set and RAM data is transferred to SBUF0 (after the contents of RAM and SBUF0 are exchanged when SI0WRT=1), data transmission and reception are carried out. Any data received after the transmission/reception processing terminated is left in SBUF0 and not exchanged with data in RAM.

3.11.5   SIO0 Communication Examples

3.11.5.1   Synchronous 8-bit mode

1)   Setting the clock
   - Set up SBR0 when using an internal clock.

2)   Setting the mode
   - Set as follows:
     SI0CTR = 0, SI0DIR = ?, SI0IE = 1

3)   Setting up the ports

	Clock Port (P12)
Internal clock	Output
External clock	Input

	Data Output Port (P10)	Data I/O Port (P11)
Data transmission only	Output	—
Data reception only	—	Input
Data transmission/reception (3-wire)	Output	Input
Data transmission/reception (2-wire)	—	N-channel open drain output

4)   Setting up output data
   - Write the output data into SBUF0 in the data transmission or data transmission/reception mode.

5)   Starting operation
   - Set SI0RUN.

6)   Reading data (after an interrupt)
   - Read SBUF0 (SBUF0 has been loaded with serial data from the data I/O port even in the transmission mode).
   - Clear SI0END.
   - Return to step 4) when repeating transmission/reception processing.

3.11.5.2   Continuous data transmission/reception mode

1) Setting the clock
   - Set up SBR0 when using an internal clock

2) Setting the mode
   - Set as follows:
     SI0BNK = ?, SI0WRT = 1, SI0DIR = ?, SI0IE = 1

3) Setting up the ports

	Clock Port (P12)
Internal clock	Output
External clock	Input

	Data Output Port (P10)	Data I/O Port (P11)
Data transmission only	Output	—
Data reception only	—	Input
Data transmission/reception (3-wire)	Output	Input
Data transmission/reception (2-wire)	—	N-channel open drain output

4) Setting up the continuous data bit register
   - Specify the number of bits to be subject to continuous transmission/reception processing.

5) Setting up output data
   - Transfer the output data of the specified bit length to data RAM at the specified address in the data transmission or data transmission/reception mode.
   - Transferred to
     RAM area from (01C0[H] to 01DF[H]) when SI0BNK = 0
     RAM area from (01E0[H] to 01FF[H]) when SI0BNK = 1.
   - Data transmission and reception processing is started after the operation flag is set and the contents of RAM and SBUF0 are exchanged. Consequently, there is no need to transfer data to SBUF0.

6) Starting operation
   - Set SI0CTR.
   - Set SI0RUN.
   * Suspending continuous data transfer processing
   - Set S0WSTP.

   ⇒ Resuming continuous data transfer processing
   - Clear S0WSTP.

   * Checking the number of bytes transferred during continuous data transfer processing
   - Read S0XBYT4 to S0XBYT0.

7) Reading data (after an interrupt)
   - Received data has been stored in data RAM at the specified address and SBUF0.
     RAM addresses (01C1[H] to 01DF[H]) when SI0BNK = 0
     RAM addresses (01E1[H] to 01FF[H]) when SI0BNK = 1
   - The last 8 bits or less of received data is left in SBUF0 and not present in RAM.
   - Clear SI0END.
   - Return to step 5) when repeating transmission/reception processing.

3.11.6   SIO0 HALT Mode Operation

3.11.6.1   Synchronous 8-bit mode

1) SIO0 synchronous 8-bit mode processing is enabled in HALT mode.

2) HALT mode can be released by an interrupt that is generated during SIO0 synchronous 8-bit mode processing.

3.11.6.2   Continuous data transmission/reception mode

1) SIO0 suspends processing immediately before the contents of RAM and SBUF0 are exchanged when HALT mode is entered in the continuous data transmission/reception mode. After HALT mode is entered, SIO0 continues processing until immediately before the contents of first RAM address and SBUF0 are exchanged. After HALT mode is released, SIO0 resumes the suspended processing.

2) Since SIO0 processing is suspended by HALT mode, it is impossible to release HALT mode using a continuous data transmission/reception mode SIO0 interrupt.

3.12 Serial Interface 1 (SIO1)

3.12.1 Overview

The serial interface SIO1 incorporated in this series of microcontrollers provides the following four functions:

1) Mode 0: Synchronous 8-bit serial I/O (2- or 3-wire system, transfer rates of 2 to 512 Tcyc)
2) Mode 1: Asynchronous serial (Half-duplex, 8 data bits, 1 stop bit, baudrates of 8 to 2048 Tcyc)
3) Mode 2: Bus-master (start bit, 8 data bits, transfer rates of 2 to 512 Tcyc)
4) Mode 3: Bus-slave (start detection, 8 data bits, stop detection)

3.12.2 Functions

1) Mode 0: Synchronous 8-bit serial I/O
   - Performs 2- or 3-wire synchronous serial communication. The clock may be an internal or external clock.
   - The clock rate of the internal clock is programmable within the range of 2 to 512 Tcyc.
2) Mode 1: Asynchronous serial (UART)
   - Performs half-duplex, 8 data bits, 1 stop bit asynchronous serial communication.
   - The baudrate is programmable within the range of 8 to 2048 Tcyc.
3) Mode 2: Bus-master
   - SIO1 is used as a bus master controller.
   - The start conditions are automatically generated but the stop conditions must be generated by manipulating ports.
   - Clock synchronization is used. Since it is possible to verify the transfer-time bus data at the end of transfer, this mode can be combined with mode 3 to provide support for multi-master configurations.
   - The period of the output clock is programmable within the range of 2 to 512 Tcyc.
4) Mode 3: Bus-slave
   - SIO1 is used as a slave device of the bus.
   - Start/stop condition detection processing is performed but the detection of an address match condition and the generation of an acknowledge require program intervention.
   - SIO1 can generate an interrupt after automatically placing the clock line at the low level on the falling edge of the eighth clock for recognition by a program.
5) Interrupt generation

   An interrupt request is generated at the end of communication if the interrupt request enable flag is set.
6) To control serial interface 1 (SIO1), it is necessary to manipulate the following special function registers.
   - SCON1, SBUF1, SBR1
   - P1, P1DDR, P1FCR

Address	Initial Value	R/W	Name	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE34	0000 0000	R/W	SCON1	-	SI1M1	SI1M0	SI1RUN	SI1REC	SI1DIR	SI1OVR	SI1END	SI1IE
FE35	00000 0000	R/W	SBUF1	SBUF18	SBUF17	SBUF16	SBUF15	SBUF14	SBUF13	SBUF12	SBUF11	SBUF10
FE36	0000 0000	R/W	SBR1	-	SBRG17	SBRG16	SBRG15	SBRG14	SBRG13	SBRG12	SBRG11	SBRG10

3.12.3　Circuit Configuration

3.12.3.1　SIO1 control register (SCON1)　(8-bit register)

1)　This register controls the operation and interrupts of SIO1.

3.12.3.2　SIO1 shift register (SIOSF1)　(8-bit shift register)

1)　This register is a shift register used to transfer and receive SIO1 data.

2)　This register cannot be accessed with an instruction. It is accessed via SBUF1.

3.12.3.3　SIO1 data register (SBUF1)　(9-bit register)

1)　The lower-order 8 bits of SBUF1 are transferred to SIOSF1 at the beginning of data transfer.

2)　At the end of data transfer, the contents of SIOSF1 are placed in the lower-order 8 bits of SBUF1. In modes 1, 2, and 3, since the 9th input data is placed in bit 8 of SBUF1, it is possible to check for a stop bit.

3.12.3.4　SIO1 baudrate generator (SBR1)　(8-bit reload counter)

1)　This is a reload counter for generating internal clocks.

2)　The generator can generate clocks of 2 to 512 Tcyc in modes 0 and 2 and clocks of 8 to 2048 Tcyc in mode 1.

Table 3.12.1   SIO1 Operations and Operating Modes

		Synchronous (Mode 0)		UART (Mode 1)		Bus Master (Mode 2)		Bus Slave (Mode 3)	
		Transfer SI1REC=0	Receive SI1REC=1	Transfer SI1REC=0	Receive SI1REC=1	Transfer SI1REC=0	Receive SI1REC=1	Transfer SI1REC=0	Receive SI1REC=1
Start bit		None	None	Output (Low)	Input (Low)	See 1) and 2) below	Not required	Not required	See 2) below
Data output		8 (Shift data)	8 (All 1s)	8 (Shift data)	8 (All 1s)	8 (Shift data)	8 (All 1s)	8 (Shift data)	8 (All 1s)
Data input		8 (Input pin)	←	8 (Input pin)	←	8 (Input pin)	←	8 (Input pin)	←
Stop bit		None	←	Output (High)	Input (H/L)	Input (H/L)	Output (SBUF1,bit8)	Input (H/L)	Output (L)
Clock		8	←	9 (Internal)	←	9	←	Low output on falling edge of 8th clock	←
Operation start		SI1RUN ↑	←	1) SI1RUN ↑ 2) Start bit detected	Start bit detected	1) No start bit on falling edge of SI1END when SI1RUN=1 2) With start bit on rising edge of SI1RUN when SI1END=0	1) on left side	1) on right side	1) Clock released on falling edge of SI1END when SI1RUN=1 2) Start bit detected when SI1RUN=0 and SI1END=0
Period		2 to 512 Tcyc	←	8 to 2048 Tcyc	←	2 to 512Tcyc	←	2 to 512Tcyc	←
SI1RUN (bit 5)	Set	Instruction	←	1) Instruction 2) Start bit detected	Start bit detected	Instruction	Already set	Already set	Start bit detected
	Clear	End of processing	←	End of stop bit	←	1) Stop condition detected 2) When arbitration lost (Note 1)	←	1) Stop condition detected 2) Ack=1 detected	←
SI1END (bit 1)	Set	End of processing	←	End of stop bit	←	1) Rising edge of 9th clock 2) Stop condition detected	←	1) Falling edge of 8th clock 2) Stop condition detected	←
	Clear	Instruction	←	Instruction	←	Instruction	←	Instruction	←

Note 1: If internal data output state="H" and data port state= "L" conditions are detected on the rising edges of the first to 8th clocks, the microcontroller recognizes a bus contention loss and clears SI1RUN (and also stops the generation of the clock at the same time).

Table 3.12.1   SIO1 Operations and Operating Modes (cont.)

		Synchronous (Mode 0)		UART (Mode 1)		Bus Master (Mode 2)		Bus Slave (Mode 3)	
		Transfer SI1REC=0	Receive SI1REC=1	Transfer SI1REC=0	Receive SI1REC=1	Transfer SI1REC=0	Receive SI1REC=1	Transfer SI1REC=0	Receive SI1REC=1
SI1OVR (bit 2)	Set	1) Falling edge of clock detected when SI1RUN=0 2) SI1END set conditions met when SI1END=1	←	1) Falling edge of clock detected when SI1RUN=0 2) SI1END set conditions met when SI1END=1	←	1) SI1END set conditions met when SI1END=1	←	1) Falling edge of clock detected when SI1RUN=0 2) SI1END set conditions met when SI1END=1 3) Start bit detected	←
	Clear	Instruction	←	Instruction	←	Instruction	←	Instruction	←
Shifter data update		SBUF1→ Shifter at beginning of operation	←	SBUF1→ Shifter at beginning of operation	←	SBUF1→ Shifter at beginning of operation	←	SBUF1→ Shifter at beginning of operation	←
Shifter→ SBUF1 (bits 0 to 7)		Rising edge of 8th clock	←	When 8-bit data transferred	When 8-bit data received	Rising edge of 8th clock	←	Rising edge of 8th clock	←
Automatic update of SBUF1, bit 8		None	←	Input data read in on stop bit	←	Input data read in on rising edge of 9th clock	←	Input data read in on rising edge of 9th clock	←
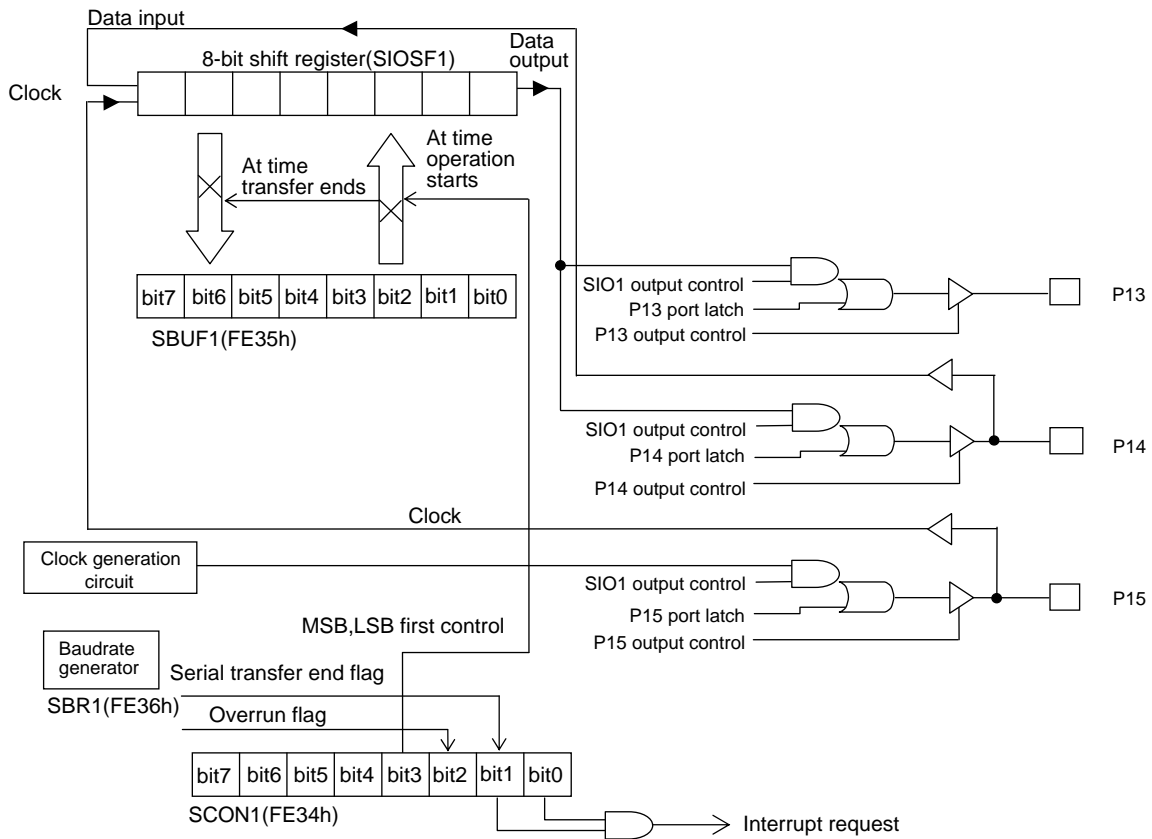
Figure 3.12.1   SIO1 Mode 0: Synchronous 8-bit Serial I/O Block Diagram (SI1M1=0, SI1M0=0)
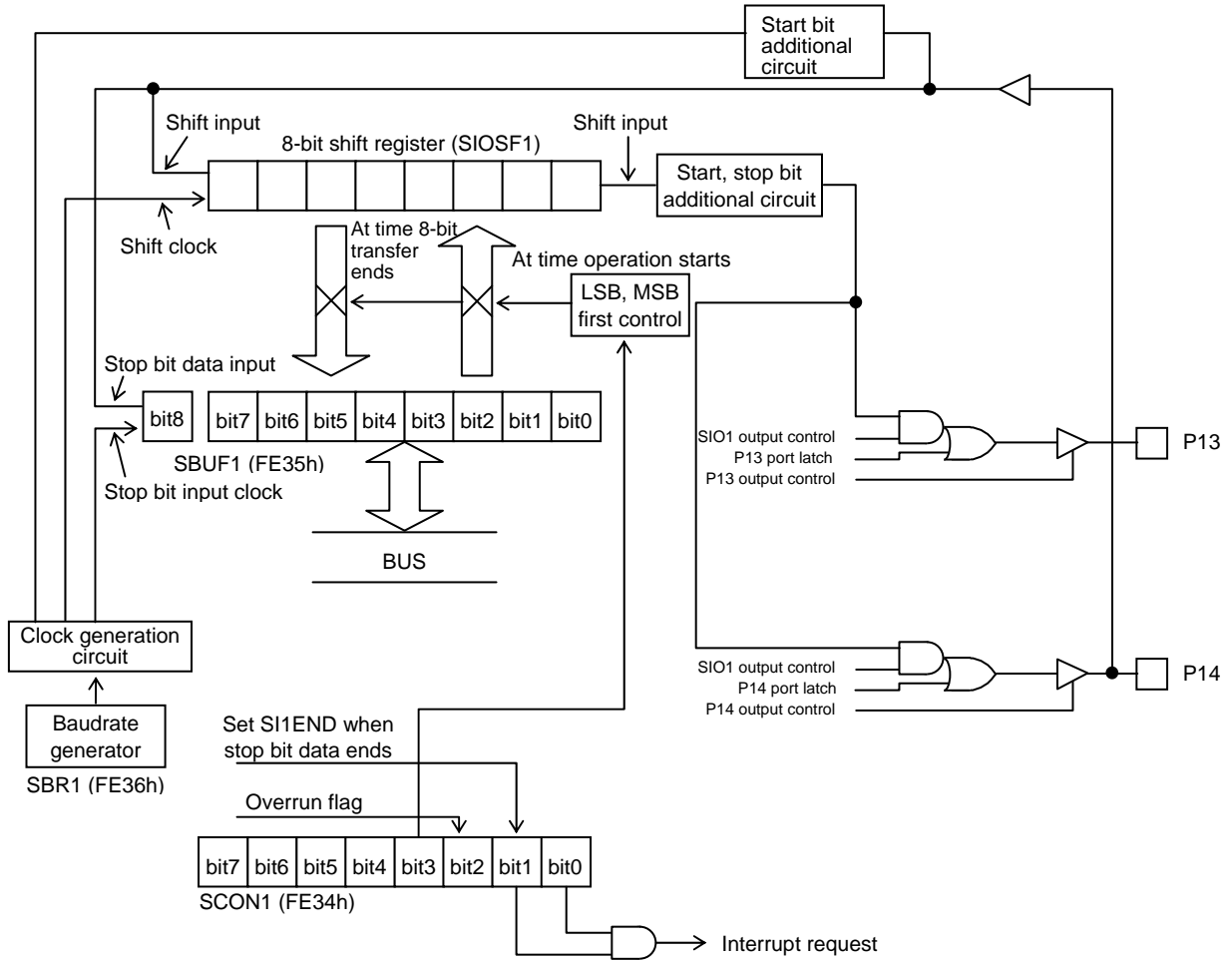
Figure 3.12.2   SIO1 Mode 1: Asynchronous Serial [UART] Block Diagram (SI1M1=0, SI1M0=1)

3.12.4 SIO1 Communication Examples

3.12.4.1 Synchronous serial communication (mode 0)

1) Setting the clock
   - Set up SBR1 when using an internal clock.

2) Setting the mode
   - Set as follows:
     SI1M0 = 0, SI1M1= 0, SI1DIR, SI1IE = 1

3) Setting up the ports and SI1REC (bit 4)

	Clock Port P15
Internal clock	Output
External clock	Input

	Data Output Port P13	**Data I/O Port P14**	**SI1REC**
Data transmission only	Output	–	0
Data reception only	–	Input	1
Data transmission/reception (3-wire)	Output	Input	0
Data transmission/reception (2-wire)	–	N-channel open drain output	0

4) Setting up output data
   - Write output data into SBUF1 in the data transmission mode (SI1REC=0).

5) Starting operation
   - Set SI1RUN.

6) Reading data (after an interrupt)
   - Read SBUF1 (SBUF1 has been loaded with serial data from the data I/O port even in the transmission mode).
   - Clear SI1END and exit interrupt processing.
   - Return to step 4) when repeating processing.

3.12.4.2 Asynchronous serial communication (mode 1)

1) Setting the baudrate
   - Set up SBR1.

2) Setting the mode
   - Set as follows:
     SI1M0 = 1, SI1M1 = 0, SI1DIR, SI1IE = 1

3) Setting up the ports.

	Data Output Port P13	**Data I/O Port P14**
Data transmission/reception (2-wire)	Output	Input
Data transmission/reception (1-wire)	–	N-channel open drain output

4) Starting transmission
   - Set SI1REC to 0 and write output data into SBUF1.
   - Set SI1RUN.

Note: Use the SIO1 data I/O port(P14) when using the SIO1 transmission only in mode 1.

In mode 1, transmission is automatically started when a falling edge of receive data is detected. While mode 1 is on, the falling edge of data is always sensed at the data I/O port (P14). Consequently, if the transmit port is assigned to the data output port (P13), it is likely that data transmissions are started unexpectedly according to the changes in the state of P14.

5) Starting receive operation
  - Set SI1REC to 1. (Once SI1REC is set to 1, do not attempt to write data to the SCON1 register until the SI1END flag is set.)
  - Detect the falling edge of receive data.

6) Reading data (after an interrupt)
  - Read SBUF1. (SBUF1 has been loaded with serial data read from the data I/O port even in the transmission mode. When SBUF1 is read in, the data about the position of the stop bit is read into bit 1 of the PSW.)
  - Clear SI1END and exit interrupt processing.
  - Return to step 4) when repeating processing.

Note: Make sure that the following conditions are met when performing continuous reception processing with SIO1 in mode 1 (UART):
  - The number of stop bits is set to 2 or greater.
  - Clearing of SI1END during interrupt processing terminates before the next start bit arrives.

3.12.4.3   Bus-master mode (mode 2)
1) Setting the clock
  - Set up SBR1.
2) Setting the mode.
  - Set as follows:
    SI1M0=0, SI1M1=1, SI1DIR, SI1IE=1, SI1REC=0
3) Setting up the ports
  - Designate the clock and data ports as N-channel open drain output ports.
4) Starting communication (sending an address)
  - Load SBUF1 with address data.
  - Set SI1RUN (transfer a start bit + SBUF1 (8 bits) + stop bit (H)).
5) Checking address data (after an interrupt)
  - Read SBUF1. (SBUF1 has been loaded with serial data from the data I/O port even in the transmission mode. When SBUF1 is read in, the data about the position of the stop bit is read into bit 1 of the PSW.)
  - Check for an acknowledge by reading bit 1 of the PSW.
  - If a condition for losing the bus contention occurs (see Note 1 in Table 3.12.1), no interrupt will be generated as SI1RUN is cleared in that case. If there is a possibility of a condition for losing the bus contention such as the presence of a separate master mode device, find out such condition by, for example, performing timeout processing using a timer module.
6) Sending data
  - Load SBUF1 with output data.
  - Clear SI1END and exit interrupt processing (transfer SBUF1 (8 bits) + stop bit (H)).

7) Checking sent data (after an interrupt)

- Read SBUF1. (SBUF1 has been loaded with serial data from the data I/O port even in the transmission mode. When SBUF1 is read in, the data about the position of the stop bit is read into bit 1 of the PSW.)
- Check for an acknowledge by reading bit 1 of the PSW.
- If a condition for losing the bus contention occurs (see Note 1 in Table 3.12.1), no interrupt will be generated as SI1RUN is cleared in that case. If there is a possibility of a condition for losing the bus contention such as the presence of a separate master mode device, find out such condition by, for example, performing timeout processing using a timer module.
- Return to step 6) when continuing data transmission.
- Go to step 10) to terminate communication.

8) Receiving data

- Set SI1REC to 1.
- Clear SI1END and exit interrupt processing (receive (8 bits) + SBUF1 bit 8 (acknowledge) output).

9) Reading received data (after an interrupt)

- Read SBUF1.
- Return to step 8) to continue reception of data.
- Go to * in step 10) to terminate processing. At this moment, SBUF1 bit 8 data has already been presented as acknowledge data and the clock for the master side has been released.

10) Terminating communication

- Manipulate the clock output port (P15FCR=0, P15DDR=1, P15=0) and set the clock output to 0.
- Manipulate the data output port (P14FCR=0, P14DDR=1, P14=0) and set the data output to 0.
- Restore the clock output port into the original state (P15FCR=1, P15DDR=1, P15=0) and release the clock output.
- * Wait for all slaves to release the clock and the clock to be set to 1.
- Allow for a data setup time, then manipulate the data output port (P14FCR=0, P14DDR=1, P14=1) and set the data output to 1. In this case, the SIO1 overrun flag SI1OVR (SCON1:FE34, bit 2) is set but this will exert no influence on the operation of SIO1.
- Restore the data output port into the original state (set P14FCR to 1, then P14DDR to 1 and P14 to 0).
- Clear SI1END and SI1OVR, then exit interrupt processing.
- Return to step 4) to repeat processing.

3.12.4.4 Bus-slave mode (mode 3)

1) Setting the clock

- Set up SBR1 (to set the acknowledge data setup time).

2) Setting the mode

- Set as follows:
  SI1M0 = 1, SI1M1 = 1, SI1DIR, SI1IE = 1, SI1REC = 0

3) Setting up ports

- Designate the clock and data ports as N-channel open drain output ports.

4) Starting communication (waiting for an address)

*1 • Set SI1REC.

*2 • SI1RUN is automatically set on detection of a start bit.

• Perform receive processing (8 bits) and set the clock output to 0 on the falling edge of the 8th clock, which generates an interrupt.

5) Checking for address data (after an interrupt)

• Detecting a start condition sets SI1OVR. Check SI1RUN=1 and SI1OVR=1 to determine if the address has been received.

(SI1OVR is not automatically cleared. Clear it by instruction.)

• Read SBUF1 and check the address.

• If no address match occurs, clear SI1RUN and SI1END and exit interrupt processing, then wait for a stop condition detection at * of step 8).

6) Receiving data

* • Clear SI1END and exit interrupt processing. (If a receive sequence has been performed, send an acknowledge and release the clock port after the lapse of (SBR1 value + 1) × Tcyc.)

• When a stop condition is detected, SI1RUN is automatically cleared and an interrupt is generated. Then, clear SI1END to exit interrupt processing and return to *2 in step 4).

• Perform a receive operation (8 bits), then set the clock output to 0 on the falling edge of the 8th clock, after which an interrupt occurs. The clock counter will be cleared if a start condition is detected in the middle of receive processing. In such a case, another 8 clocks are required to generate an interrupt.

• Read SBUF1 and store the read data.

Note: Bit 8 of SBUF1 is not yet updated because the rising edge of 9th clock has not yet occurred.

• Return to * in step 6) to continue receive processing.

7) Sending data

• Clear SI1REC.

• Load SBUF1 with output data.

• Clear SI1END and exit interrupt processing. (Send an acknowledge for the preceding reception operation and release the clock port after the lapse of (SBR1 value + 1) × Tcyc.)

*1 • Perform a send operation (8 bits) and set the clock output to 0 on the falling edge of the 8th clock, after which an interrupt occurs.

*2 • Go to *3 in step 7) if SI1RUN is set to 1.

• If SI1RUN is set to 0, implying an interrupt from *4 in step 7), clear SI1END and SI1OVR and return to *1 in step 4).

*3 • Read SBUF1 and check send data as required.

Note: Bit 8 of SBUF1 is not yet updated because the rising edge of 9th clock has not yet occurred.

• Load SBUF1 with the next output data.

• Clear SI1END and exit interrupt processing. (Release the clock port after the lapse of (SBR1 value + 1) × Tcyc).

• Return to *1 in step7) if an acknowledge from the master is present (L).

- If there is no acknowledge presented from the master (H), SIO1, recognizing the end of data transmission, automatically clears SI1RUN and releases the data port.

  * However, in a case that restart condition comes just after the event, SI1REC must be set to 1 before exiting the interrupt (SI1REC is for detecting a start condition and is not set automatically). It may disturb the transmission of address from the master if there is an unexpected restart just after slave's transmission (when SI1REC is not set to 1 by instruction).

*4 • When a stop condition is detected, an interrupt is generated and processing returns to *2 in step 7).

8) Terminating communication

- Set SI1REC.
- Return to * in step 6) to cause communication to automatically terminate.
- To force communication to termination, clear SI1RUN and SI1END (release the clock port).

* • An interrupt occurs when a stop condition is detected. Then, clear SI1END and SI1OVR and return to *2 in step 4).

3.12.5 Related Registers

3.12.5.1 SIO1 control register (SCON1)

1) This register is an 8-bit register that controls the operation and interrupts of SIO1.

Address	Initial Value	R/W	Name	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE34	0000 0000	R/W	SCON1	-	SI1M1	SI1M0	SI1RUN	SI1REC	SI1DIR	SI1OVR	SI1END	SI1IE

SI1M1 (bit 7): SIO1 mode control

SI1M0 (bit 6): SIO1 mode control

Table 3.12.2   SIO1 Operation Modes

Mode	SI1M1	SI1M0	Operation Mode
0	0	0	Synchronous 8-bit SIO
1	0	1	UART (1 stop bit, no parity)
2	1	0	Bus master mode
3	1	1	Bus slave mode

SI1RUN (bit 5): SIO1 operation flag

1) A 1 in this bit indicates that SIO1 is running.
2) See Table 3.12.1 for the conditions for setting and clearing this bit.

SI1REC (bit 4): SIO1 receive/transmit control

1) Setting this bit to 1 places SIO1 into the receive mode.
2) Setting this bit to 0 places SIO1 into the transmit mode.

SI1DIR (bit 3): MSB/LSB first select

1) Setting this bit to 1 places SIO1 into the MSB first mode.
2) Setting this bit to 0 places SIO1 into the LSB first mode.

SI1OVR (bit 2): SIO1 overrun flag

    1)   In modes 0, 1, and 3, this bit is set when a falling edge of the input clock is detected when SI1RUN=0.

    2)   This bit is set if the conditions for setting SI1END are established when SI1END=1.

    3)   In mode 3 this bit is set when the start condition is detected.

    4)   This bit must be cleared with an instruction.

SI1END (bit 1): End of serial transfer flag

    1)   This bit is set when serial transfer terminates (see Table 3.12.1).

    2)   This bit must be cleared with an instruction.

SI1IE (bit 0): SIO1 interrupt request enable control

When this bit and SI1END are set to 1, an interrupt request to vector address 003BH is generated.

3.12.5.2   Serial buffer 1 (SBUF1)

    1)   Serial buffer 1 is a 9-bit register used to store data to be handled during SIO1 serial transfer.

    2)   The lower-order 8 bits of SBUF1 are transferred to the data shift register for data transmission/ reception at the beginning of transfer processing and the contents of the shift register are placed in the lower-order 8 bits of SBUF1 when 8-bit data is transferred.

    3)   In modes 1, 2, and 3, bit 8 of SBUF1 is loaded with the 9th data bit that is received (data about the position of the stop bit).

Address	Initial Value	R/W	Name	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE35	00000 0000	R/W	SBUF1	SBUF18	SBUF17	SBUF16	SBUF15	SBUF14	SBUF13	SBUF12	SBUF11	SBUF10

3.12.5.3   Baudrate generator register (SBR1)

    1)   This register is an 8-bit register that defines the transfer rate of SIO1 serial transfer.

    2)   Loading this register with data causes the baudrate generating counter to be initialized immediately.

    3)   The baudrate varies from mode to mode (the baudrate generator is disabled in mode 3).

    Modes 0 and 2: $TSBR1 = (SBR1\ value + 1) \times 2Tcyc$
                                 (Value range = 2 to 512 Tcyc)

    Mode 1: $TSBR1 = (SBR1\ value + 1) \times 8Tcyc$
                                   (Value range = 8 to 2048Tcyc)

Address	Initial Value	R/W	Name	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE36	0000 0000	R/W	SBR1	-	SBRG17	SBRG16	SBRG15	SBRG14	SBRG13	SBRG12	SBRG11	SBRG10

3.13   Serial Interface 4 (SIO4)

3.13.1   Overview

The serial interface 4 (SIO4) incorporated in this series of microcontrollers is a synchronous serial interface providing the following functions:

1)   Continuous synchronous data transfer

- Data transfer of arbitrary number of bytes between 1 and 4096 bytes
- Transfer rate: 4/3 to 1020/3 Tcyc

2)   16-bit CRC code calculation

3.13.2   Functions

1)   Continuous synchronous data transfer

- Performs 2- or 3-wire synchronous serial communication. The clock may be an internal clock or external clock.
- The clock period of the internal clock is 4n/3 Tcyc (n= 1 to 255; Note that n=0 is inhibited).
- Transmits and receives arbitrary number of data bytes between 1 to 4096 bytes automatically and continuously. Transmit data is automatically transferred from RAM to a shift register (SI4BUF) while receive data is automatically transferred from the shift register (SI4BUF) to RAM.
- The RAM area to be used for continuous transmission and reception can be allocated to any address in 1-byte units.
- When the internal clock is used, suspension and resumption of continuous data transfer can be controlled in 1- or 2-byte units.
- Data can be communicated either MSB or LSB first.
- 16-bit CRC code calculation can be performed on serial transfer data.
- Related ports

  Either port pin group P22 to P24 or P13 to P15 can be selected for serial communication.

Port		I/O	Description
P22	P13	I/O	Serial data input/output pin
P23	P14	I/O	Serial data input/output pin
P24	P15	I/O	Synchronous clock input/output pin

2)   Interrupt generation

An interrupt request is generated at the end of transfer when the interrupt request enable bit is set.

3)   To control serial interface 4 (SIO4), it is necessary to manipulate the following special function registers.

- S4ADRL, S4BYTH, CRCL, CRCH, CRCCNT, SI4CN0, SI4CN1, SI4BUF, S4BAUD, S4ADDR, S4BYTE
- P1, P1DDR, P2, P2DDR

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED6	0000 0000	R/W	S4ADRL	S4ADL7	S4ADL6	S4ADL5	S4ADL4	S4ADL3	S4ADL2	S4ADL1	S4ADL0
FED7	0000 0000	R/W	S4BYTH	S4STPWD	S4BYTRD	S4BYTH5	S4BYTH4	S4BYTH3	S4BYTH2	S4BYTH1	S4BYTH0
FED8	0000 0000	R/W	CRCL	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
FED9	0000 0000	R/W	CRCH	CRC15	CRC14	CRC13	CRC12	CRC11	CRC10	CRC9	CRC8
FEDA	0000 0000	R/W	CRCCNT	CRCON	CRCLRZ	CRCRD	1/0SEL	S4STPCEN	S4STPCHI	S4STPSL1	S4STPSL0
FEDB	0000 0000	R/W	SI4CN0	SI4RUN	SBITON	MSBSEL	S4RAM	S4CKPL	SI4WRT	SI4END	SI4IE
FEDC	0000 0000	R/W	SI4CN1	PARA	P1/P0	P22/P23	P24OUT	P23MOS	P23OUT	P22MOS	P22OUT
FEDD	0000 0000	R/W	SI4BUF	S4BUF7	S4BUF6	S4BUF5	S4BUF4	S4BUF3	S4BUF2	S4BUF1	S4BUF0
FEDE	0000 0000	R/W	S4BAUD	S4BAU7	S4BAU6	S4BAU5	S4BAU4	S4BAU3	S4BAU2	S4BAU1	S4BAU0
FEDF	0000 0000	R/W	S4ADDR	S4WSTP	S4PTSEL	S4ADR5	S4ADR4	S4ADR3	S4ADR2	S4ADR1	S4ADR0
FEE0	0000 0000	R/W	S4BYTE	S4BYT7	S4BYT6	S4BYT5	S4BYT4	S4BYT3	S4BYT2	S4BYT1	S4BYT0

3.13.3 Circuit Configuration

3.13.3.1 SIO4 transfer RAM address register low byte (S4ADRL) (8-bit register)

1) This register is used to define the starting address of the RAM area to be used for data transfer.

3.13.3.2 SIO4 transfer data byte register high byte (S4BYTH) (8-bit register)

1) This register is used to define the number of data bytes to be transferred via the SIO4 in the continuous data transfer mode.

3.13.3.3 CRC (Cyclic Redundancy Check) register (CRCL, CRCH) (8-bit register)

1) This register is used to set generator polynomials for cyclic redundancy check (CRC) encoding.

3.13.3.4 CRC calculation result register (CRC16) (16-bit register)

1) This register stores the calculation results of CRC encoding.

3.13.3.5 CRC control register (CRCCNT) (8-bit register)

1) This register controls the operation of CRC processing.
2) This register also controls the suspension of the serial interface port in continuous data transfer mode.

3.13.3.6 SIO4 control register 0 (SI4CN0) (8-bit register)

1) This register 0 controls the operation and interrupts of SIO4.

3.13.3.7 SIO4 control register 1 (SI4CN1) (8-bit register)

1) This register controls the SIO4 interface port.

3.13.3.8 SIO4 shift register (SI4BUF) (8-bit shift register)

1) This register is an 8-bit shift register used for SIO4 serial transfer.

3.13.3.9　SIO4 baudrate generator (S4BAUD)　(8-bit reload register)

1)　The SIO4 baudrate generator is a reload counter for generating internal clocks.

2)　It can generate clocks at intervals of 4n/3 Tcyc (n=1 to 255; Note: n=0 is inhibited).

3.13.3.10　SIO4 transfer RAM address register high byte (S4ADDR)　(8-bit register)

1)　This register is used to define the starting address of the RAM area to be used for data transfer.

3.13.3.11　SIO4 transfer data byte register low byte (S4BYTE)　(8-bit register)

1)　This register is used to define the number of data bytes to be transferred in the continuous data transfer mode.

3.13.4　Related Registers

3.13.4.1　CRC register (CRCL, CRCH)

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED8	0000 0000	R/W	CRCL	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
FED9	0000 0000	R/W	CRCH	CRC15	CRC14	CRC13	CRC12	CRC11	CRC10	CRC9	CRC8

1)　The CRC register for setting up the generator polynomial is 16 bits long and is made up of two registers, CRCL and CRCH.

2)　This register is loaded with the data for setting up the generator polynomial when the CRC control register (CRCCNT), bit 5 (CRCRD) is set to 0. This register must be set up only once at the beginning.

Example: The CRC encoding/decoding circuit for the 16-bit generator polynomial

$G(x) = X^{16} + X^{12} + X^5 + 1$ is shown below.
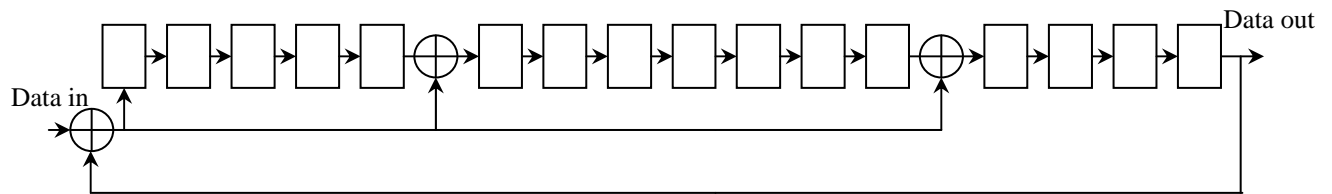
Figure 3.13.1　CRC Encoding/Decoding Circuit

In this example, the 16-bit long CRC register (CRCH and CRCL) must be set up as follows:

CRCH = 10[H],　CRCL = 21[H]

3)　The results of CRC calculation can be read from the CRC register (CRCH and CRCL) when the CRC control register (CRCCNT), bit 5 (CRCRD) is set to 1.

3.13.4.2 CRC control register (CRCCNT)

1) This register controls cyclic redundancy check (CRC) processing.

2) The register exercises suspension port control in the continuous data transfer mode.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEDA	0000 0000	R/W	CRCCNT	CRCON	CRCLRZ	CRCRD	1/0SEL	S4STPCEN	S4STPCHI	S4STPSL1	S4STPSL0

CRCON (bit 7): CRC calculation control flag

1: Starts calculation.

0: Stops calculation.

CRCLRZ (bit 6): CRC register control flag

1: The contents of the CRC calculation result register are preserved.

0: The CRC calculation result register is initialized.

CRCRD (bit 5): CRC results read control flag

1: The CRC calculation results are read from the CRC register.

0: The generator polynomial is read from the CRC register.

1/0SEL (bit 4): CRC calculation result register initialization control flag

1: All CRC calculation result register bits are initialized to 1.

0: All CRC calculation result register bits are initialized to 0.

S4STPCEN (bit 3): Suspension port control enable flag

1: Enables continuous transfer mode suspension control (on a 1- or 2-byte units) according to the level (H or L level selectable) of the port (either P70, P71, 72, or P73).

0: Disables suspension port control in continuous transfer mode.

S4STPCHI (bit 2): Suspension port control polarity select

1: Transfer is suspended when the port (P70 to P73) is set to the high level and resumed when the port is set to the low level.

0: Transfer is suspended when the port (P70 to P73) is set to the low level and resumed when the port is set to high level.

S4STPSL1 (bit 1): Suspension control port select

S4STPSL0 (bit 0): Suspension control port select

These bits are used to select the ports to be used to control continuous transfer suspension.

S4STPSL[1:0]	Suspension Control Port
00	P70
01	P71
10	P72
11	P73
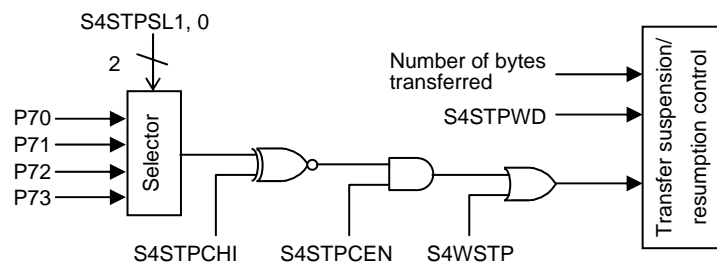
3.13.4.3 SIO4 control register 0 (SI4CN0)

1) This register is an 8-bit register that controls the operation and interrupts of SIO4.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEDB	0000 0000	R/W	SI4CN0	SI4RUN	SBITON	MSBSEL	S4RAM	S4CKPL	SI4WRT	SI4END	SI4IE

SI4RUN (bit 7): SIO4 operation control flag

> 1: Starts transfer. This bit is automatically cleared at the end of transfer.
>
> 0: Stops transfer.

SBITON (bit 6): Automatic transfer on detection of start bit control flag

> 1: Sets the serial data transfer control flag (SI4RUN) automatically on detection of the falling edge of the serial input data.
>
> 0: Does not set the automatic transfer.

MSBSEL (bit 5): MSB/LSB transfer direction control flag

> 1: MSB transfer
>
> 0: LSB transfer

S4RAM (bit 4): Start-time RAM address select

> 1: On output, the RAM address 00 is selected first, followed by RAM address 01, and so on.
>
> On input, the RAM address 01 is selected first, followed by RAM address 02, and so on.
>
> 0: On outout, the serial shift register is selected first, followed by RAM address 00, and so on.
>
> On input, the RAM address 00 is selected first, followed by RAM address 01, and so on.

S4CKPL (bit 3): SIO4 clock polarity switching flag

> 1: Data is output on detection of the rising edge of a clock and input on detection of the falling edge of a clock.
>
> 0: Data is output on detection of the falling edge of a clock and input on detection of the rising edge of a clock.

SI4WRT (bit 2): SIO4 transmission/reception mode setting flag

> 1: Transmission and reception (the contents of RAM and shift register are automatically exchanged when data is transferred continuously.)
>
> 0: Transmission only (the contents of RAM are automatically transferred to the shift register but the contents of RAM remain unchanged when data is transferred continuously.)

SI4END (bit 1): End of SIO4 transfer flag

This bit is automatically set at the end of SIO4 transfer. It must be cleared with an instruction.

SI4IE (bit 0): Interrupt enable flag

An interrupt request to vector address 003BH is generated when this bit and SI4END are set to 1.

3.13.4.4 SIO4 control register 1 (SI4CN1)

1)   This register is an 8-bit register that sets up the SIO4 communication ports.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEDC	0000 0000	R/W	SI4CN1	PARA	P1/P0	P22/P23	P24OUT	P23MOS	P23OUT	P22MOS	P22OUT

PARA (bit 7): Parallel mode select

> 1: Turns on the parallel mode.
>
> 0: Turns off the parallel mode (serial mode).

P1/P0 (bit 6):

Parallel mode (When PARA=1)・・・P1/P0 select flag

1: The data I/O port for the 8-bit parallel interface is assigned to P1.

0: The data I/O port for the 8-bit parallel interface is assigned to P0.

Serial mode (When PARA=0)・・・P24 (SIO4 clock) output type select flag

1: CMOS output

0: N-channel open drain output

P22/P23 (bit 5): SIO4 serial data input port select flag

1: Serial data to SIO4 is received via P22.

0: Serial data to SIO4 is received via P23.

P24OUT (bit 4): P24 (sync clock) I/O control flag

1: The SIO4 sync clock is transmitted from P24.

0: No SIO4 sync clock is transmitted from P24 (external clock mode).

P23MOS (bit 3): P23 (serial data) output type select flag

1: CMOS output

0: N-channel open drain output

P23OUT (bit 2): P23 (serial data) I/O control flag

1: SIO4 serial data is output from P23.

0: No SIO4 serial data is output from P23.

P22MOS (bit 1): P22 (serial data) output type select flag

1: CMOS output

0: N-channel open drain output

P22OUT (bit 0): P22 (serial data) I/O control flag

1: SIO4 serial data is output from P22.

0: No SIO4 serial data is output from P22.

Mode			SI4CN1 Register							
		Clock Internal/ External	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Transmit	Receive		PARA	P1/ P0	P22/P23	P24OUT	P23MOS	P23OUT	P22MOS	P22OUT
P22 data transmit	None	Internal	0	*1	1 (*2)	1	—	0	*1	1
		External		—		0				
P23 data transmit	None	Internal	0	*1	0 (*2)	1	*1	1	—	0
		External		—		0				
P23＆P22 data transmit	None	Internal	0	*1	0/1 (*2)	1	*1	1	*1	1
		External		—		0				
None	P22 data receive	Internal	0	*1	1	1	—	0	—	0
		External		—		0				
None	P23 data receive	Internal	0	*1	0	1	—	0	—	0
		External		—		0				
P22 data transmit	P23 data receive	Internal	0	*1	0	1	—	0	*1	1
		External		—		0				
P23 data transmit	P22 data receive	Internal	0	*1	1	1	*1	1	—	0
		External		—		0				

*1: Set according to the output type (CMOS/N-channel open drain).

*2: Since CRC encoding is performed on the input data, select the port (P22/P23) that is configured for output when performing calculation on the output data.

3.13.4.5 SIO4 shift register (SI4BUF)

1) This register is an 8-bit shift register for SIO4 serial data transfer.

2) Data to be transmitted or received is written to or read from this shift register directly.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEDD	0000 0000	R/W	SI4BUF	S4BUF7	S4BUF6	S4BUF5	S4BUF4	S4BUF3	S4BUF2	S4BUF1	S4BUF0

3.13.4.6 SIO4 baudrate register (S4BAUD)

1) This register is an 8-bit register that sets the transfer rate of SIO4 serial transfer.

2) The transfer rate is calculated as follows:

TS4BAUD = 4 × S4BAUD value × 1/3 Tcyc

S4BAUD can take a value from 1 to 255 and the valid value range of TS4BAUD is from 4/3 to 1020/3 Tcyc.

* The S4BAUD value of 00[H] is inhibited.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEDE	0000 0000	R/W	S4BAUD	S4BAU7	S4BAU6	S4BAU5	S4BAU4	S4BAU3	S4BAU2	S4BAU1	S4BAU0

3.13.4.7 SIO4 transfer RAM address register low byte (S4ADRL)

1) This register defines the lower-order 8 bits of the start address of the RAM area to be used for data transfer.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED6	0000 0000	R/W	S4ADRL	S4ADL7	S4ADL6	S4ADL5	S4ADL4	S4ADL3	S4ADL2	S4ADL1	S4ADL0

3.13.4.8 SIO4 transfer RAM address register high byte (S4ADDR)

1) This register is used to control the suspension of continuous data transfer processing.

2) The register is also used to select the ports for serial communication.

3) The register defines the higher-order 6 bits of the start address of the RAM area to be used for data transfer.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEDF	0000 0000	R/W	S4ADDR	S4WSTP	S4PTSEL	S4ADR5	S4ADR4	S4ADR3	S4ADR2	S4ADR1	S4ADR0

S4WSTP (bit 7): Continuous data transfer mode suspension control flag

1: Disables automatic data transfer between the RAM and shift register.
Continuous data transfer processing is suspended when the current transfer of the data to or from the shift register is finished. If S4STPWD (S4BYTH, bit 7) is set to 1, however, data transfer processing is suspended after the transfer of an even byte data is finished. Suspension of data transfer cannot be controlled when the SIO4 is running on an external clock.

0: The suspension mode is canceled.

S4PTSEL (bit 6): Serial communication port select

	S4PTSEL=0	S4PTSEL=1
Serial data I/O pin	P22	P13
Serial data I/O pin	P23	P14
Sync clock I/O pin	P24	P15

When S4PTSEL is set to 1, make settings explained in 3.13.4.4, "SIO4 control register 1 (SI4CN1)," after substituting P22, P23, and P24 for P13, P14, and P15, respectively.

S4ADR5 to 0 (bits 5 to 0): Transfer RAM start address higher-order 6 bits

The lower-order 6 bits of S4ADDR and 8 bits of S4ADRL are used to define the start address of the RAM area to be used for data transfer.

S4ADDR Lower-order 6 bits [H]	S4ADRL [H]	RAM Start Address [H]
00	00	0000
00	01	0001
〜	〜	〜
0F	FF	FFF

3.13.4.9 SIO4 transfer data byte register high byte (S4BYTH)

1) This register is used to specify continuous data transfer processing to be suspended in 1- or 2-byte units.

2) The register is also used to control how to read the number of transferred data bytes.

3) The register is used to define the higher-order 4 bits of the number of data bytes to be transferred in the continuous data transfer mode.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED7	0000 0000	R/W	S4BYTH	S4STPWD	S4BYTRD	S4BYTH5	S4BYTH4	S4BYTH3	S4BYTH2	S4BYTH1	S4BYTH0

S4STPWD (bit 7): 1-/2-byte unit suspension selection flag

> 1: Continuous mode data transfer is suspended in 2-byte units.
>
> 0: Continuous mode data transfer is suspended in 1-byte units.

S4BYTRD (bit 6): Transferred byte count read control flag

> 1: The number of transferred data bytes can be read from the lower-order 4 bits of S4BYTH and S4BYTE. If continuous mode data transfer is suspended with S4RAM (SI4CN0, bit 4) set to 0, however, the byte count that is read is (number of data bytes that are transferred minus 1). A 0 is read as the transferred byte count after the continuous mode data transfer processing is finished.
>
> 0: The readout of transferred byte count is disabled.

S4BYTH5 (bit 5): This bit is reserved and must always be set to 0.

S4BYTH4 (bit 4): This bit is reserved and must always be set to 0.

S4BYTH3 to 0 (bits 3 to 0): Higher-order 4 bits of transferred data byte count

See the next subsubsection.

3.13.4.10 SIO4 transfer data byte register low byte (S4BYTE)

1) This register is used to define the lower-order 8 bits of the number of data bytes to be transferred in the continuous data transfer mode.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEE0	0000 0000	R/W	S4BYTE	S4BYT7	S4BYT6	S4BYT5	S4BYT4	S4BYT3	S4BYT2	S4BYT1	S4BYT0

The lower-order 4 bits of S4BYTH and 8 bits of S4BYTE are used to define the number of data bytes to be transferred

S4BYTH Lower-order 4 bits [H]	S4BYTE [H]	Transfer Data Byte Count
0	00	1
0	01	2
〜	〜	〜
F	FF	4096

3.13.5   SIO4 Communication Examples

3.13.5.1   Synchronous serial interface

Example 1: Sending data bytes continuously (on the internal clock)

1)   Setting up the ports (P24, P23, and P22)

- • [P2DDR]   P24DDR=0, P23DDR=0, P22DDR=0
- • [P2]   P24=0, P23=0, P22=0
- • [S4ADDR]   S4PTSEL=0

2)   Setting up the mode

- • [SI4CN0] SBITON=0, MSBSEL=1/0, S4RAM=1, SI4WRT=0, SI4IE=1
- • [IE]   IE7=1

3)   Setting the clock

- • [SI4CN0] S4CKPL=1/0
- • [SI4CN1] PARA=0, P1/P0=1

4)   Setting up the ports [SI4CN1]

<For data transmission from P22>

- •   P22/P23 = 1, P24OUT = 1, P23OUT = 0, P22MOS = 1, P22OUT = 1

<For data transmission from P23>

- •   P22/P23 = 0, P24OUT = 1, P23MOS = 1, P23OUT = 1, P22OUT = 0

5)   Setting the baudrate [S4BAUD]

- •   Set the period of the SIO4 serial clock (internal clock) to a value between 4/3 to 1020/3 Tcyc.

6)   Setting the byte count [S4BYTH, S4BYTE]

- •   Specify the number of bytes to be transmitted continuously.

7)   Setting up the SIO4 data transfer RAM address offset register [S4ADDR, S4ADRL]

- •   Load the SIO4 RAM address register with the starting address of the RAM data area to be used for continuous serial data transmission.

8)   Setting up output data

- •   Write the number of data bytes specified in step 6) to the RAM area specified in step 7).

9)   Starting data transfer

- •   Set SI4RUN (SI4CN0, bit 7) to 1 to start data transfer.

10)   End of transfer processing

- •   When the number of data bytes specified in 6) have been output, SI4RUN (SI4CN0, bit 7) is automatically cleared, SI4END (SI4CN0, bit 1) is set, and an interrupt request to vector address 003B[H] is generated.

Example 2: Receiving data bytes continuously (on the external clock)

1) Setting up the ports (P15, P14, and P13)
   - [P1DDR]   P15DDR=0, P14DDR=0, P13DDR=0
   - [P1]   P15=0, P14=0, P13=0
   - [S4ADDR]   S4PTSEL=1

2) Setting up the mode
   - [SI4CN0]   SI4RUN=0, SBITON=1, MSBSEL=1/0,
                      S4RAM=0, SI4WRT=1, SI4IE=1
   - [IE]   IE7=1

   SBITON=1: The SIO4 detects the falling edge of the signal at the serial data input port and starts automatic SIO4 transfer.

3) Setting the clock
   - [SI4CN0]S4CKPL=1/0
   - [SI4CN1]PARA=0

4) Setting up the ports [SI4CN1]
   <For data reception from P13>
   ・P22/P23=1, P24OUT=0, P22OUT=0
   <For data reception from P14>
   ・P22/P23=0, P24OUT=0, P23OUT=0

5) Setting the byte count [S4BYTH, S4BYTE]
   - Specify the number of bytes to be received continuously.

6) Setting up the SIO4 data transfer RAM address offset register [S4ADDR, S4ADRL]
   - Load SIO4 RAM address register with the starting address of the RAM data area to be used for continuous serial data reception.

7) Starting data transfer
   - The SIO4 sets SI4RUN (SI4CN0, bit 7) and starts automatic SIO4 transfer when it detects the falling edge of the signal at the serial data input port.

8) End of transfer processing
   - SI4RUN (SI4CN0, bit 7) is automatically cleared, SI4END (SI4CN0, bit 1) is set, and an interrupt request to vector address 003B[H] is generated when the last data byte of the data whose byte count is specified in step 5) is transferred to the shift register.

9) Reading received data
   - Received data is stored in RAM sequentially starting at the RAM starting address specified in 6). The last data byte is held in the shift buffer and not transferred to RAM.
   - For example, when data is received with S4ADDR set to 04[H], S4ADRL to 00[H], S4BYTH to 00[H], and S4BYTE to 1F[H], 32 bytes of received data are stored in the RAM address area (0400[H] to 041E[H]) and the shift buffer [SI4BUF].

32	Shift buffer
	SI4BUF（FEDD［H］）

RAM
S4ADDR, S4ADRL ⟶ 0400[H]

1	2	3	4		15	16
17	18	19	20		31	041F$_H$
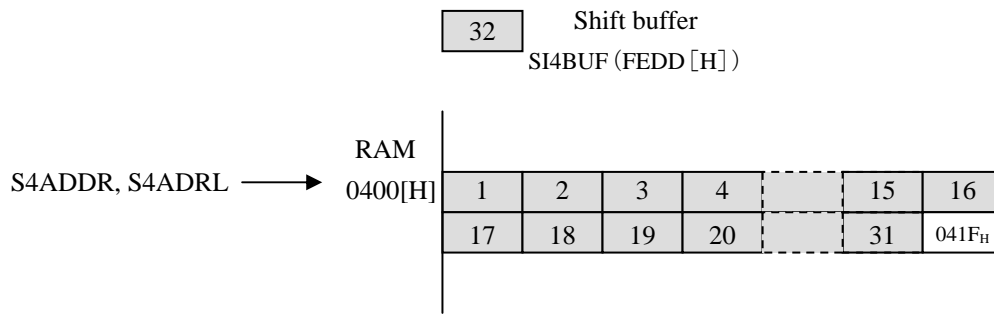
Example 3: CRC (cyclic redundancy checking) calculation

1) Setting up CRC-related registers

    Set the SIO4 registers as follows:

    (i) [CRCCNT] CRCON = 0, CRCLRZ = 0, CRCRD = 0, 1/0SEL = 0

    (ii) Define the generator polynomial [CRCH, CRCL]

    CRCH = 10[H], CRCL = 21 [H]

    (iii) [CRCCNT] CRCON = 0, CRCLRZ = 1, CRCRD = 1, 1/0SEL = 1

    * Step (ii) may be skipped if already defined.

    For 2) to 9) set the SIO4 registers in the same way as in 1) to 8) in example 2.

10) Reading received data
    • Received data is stored in RAM sequentially starting at the specified RAM starting address. The last data byte is held in the shift buffer and not transferred to RAM.
    • For example, when the byte data stream

    00_11_22_33_44_55_66_77_88_99_aa_bb_cc_dd_ee_ff[H]

    is received with S4ADDR set to 05[H], S4ADRL to 00[H], S4BYTH to 00[H], and S4BYTE to 0F[H], 16 bytes of received data are stored in the RAM address area (0500[H] to 050E[H]) and the shift buffer [SI4BUF].

11) Reading CRC results
    • The results of CRC calculation on the received data are as follows:
    CRCH = 12[H], CRCL = 48[H]

16	Shift buffer
	SI4BUF (FEDD[H])

RAM
S4ADDR, S4ADRL ⟶ 0500[H]

1	2	3	4		15	050F$_H$
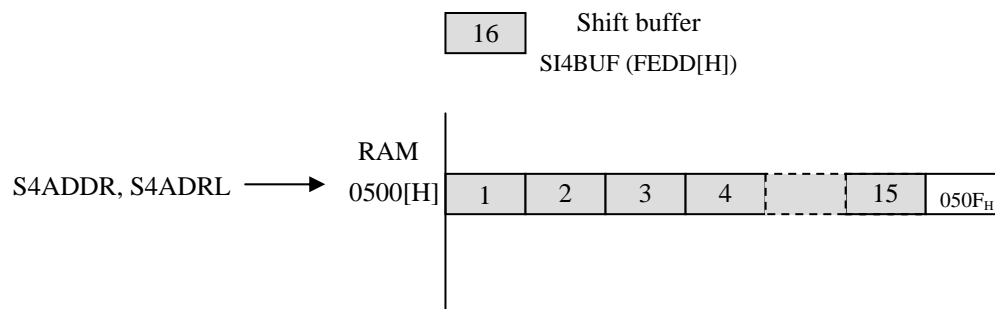
3.13.6 SIO4 HALT Mode Operation

1) The SIO4 suspends processing immediately before the contents of RAM and SI4BUF are exchanged after the microcontroller enters HALT mode. Even after the microcontroller enters HALT mode, the SIO4 continues processing immediately before the contents of the first RAM location and SI4BUF are exchanged. The SIO4 resumes and continues processing after the microcontroller exits HALT mode.

2) Since the SIO4 suspends processing on entry into HALT mode, SIO4 interrupt cannot be used to release HALT mode.

3.14 Parallel Interface

3.14.1 Overview

This series of microcontrollers can generate a read or write signal to external memory when an instruction accessing a port (P0 or P1) is executed. The generation of the address needs to be set up under software control.

3.14.2 Functions

1) External memory read mode

Execution of an instruction (PUSH, LD, etc.) for reading data from a port (P0 or P1) generates a read signal (RD#) from pin P22 or P13.

2) External memory write mode

Execution of an instruction (POP, ST, etc.) for writing data into a port (P0 or P1) generates a write signal (WR#) from pin P23 or P14.

The ports are assigned using S4PTSEL (S4ADDR, bit 6).

Port assignment		I/O	Description
P22	P13	Output	Read signal output pin
P23	P14	Output	Write signal output pin

3) To control the parallel interface, it is necessary to manipulate the following special function registers.

- SI4CN0, SI4CN1, SI4BUF
- P1, P1DDR, P2, P2DDR

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEDB	0000 0000	R/W	SI4CN0	SI4RUN	SBITON	MSBSEL	S4RAM	S4CKPL	SI4WRT	SI4END	SI4IE
FEDC	0000 0000	R/W	SI4CN1	PARA	P1/P0	P22/P23	P24OUT	P23MOS	P23OUT	P22MOS	P22OUT
FEDD	0000 0000	R/W	SI4BUF	S4BUF7	S4BUF6	S4BUF5	S4BUF4	S4BUF3	S4BUF2	S4BUF1	S4BUF0

3.14.3 Related Registers

See Subsection 3.13.4 for a description of the special function registers (SI4CN0, SI4CN1, and SI4BUF) for controlling the parallel interface.

3.14.4 Parallel Interface Programming Example

An example of configuring the special function registers for using the parallel interface is shown below, followed by related timing charts.

1) Initial setting

    Before using the parallel interface, it is necessary to perform the following sequence of initialization steps (shifting the SIO4 shift register by 1 bit and loading a 1 into the output data latch) once:

    - Set SI4CN1 to 00 [H].
    - Set SI4BUF to FF [H] (set FF [H] to the shift register).
    - Set SI4CN0 to 80 [H] (set SI4RUN to 1).
    - Set P2, bit 4 to 1 and P2DDR, bit 4 to 1 (Output "1" from P24).
    - Set P2, bit 4 to 0 (Output "0" from P24).
    - Set P2DDR, bit 4 to 0.
    - Set SI4CN0 to 00 [H].

2) Setting up the mode

    - [SI4CN1]PARA=1

3) Setting up the port [SI4CN1]

    - P22OUT=1, P22MOS=1 (output the read signal)
    - P23OUT=1, P23MOS=1 (output the write signal)

4) Selecting the parallel interface data I/O port [SI4CN1]

    - P1/P0=1 (port 1)

5) Accessing the port

    <1> External memory read mode

    - Execute an instruction (e.g., PUSH) for reading data from P1 to output a read signal to P22 at the timing of S2.

    <Read mode timing chart>
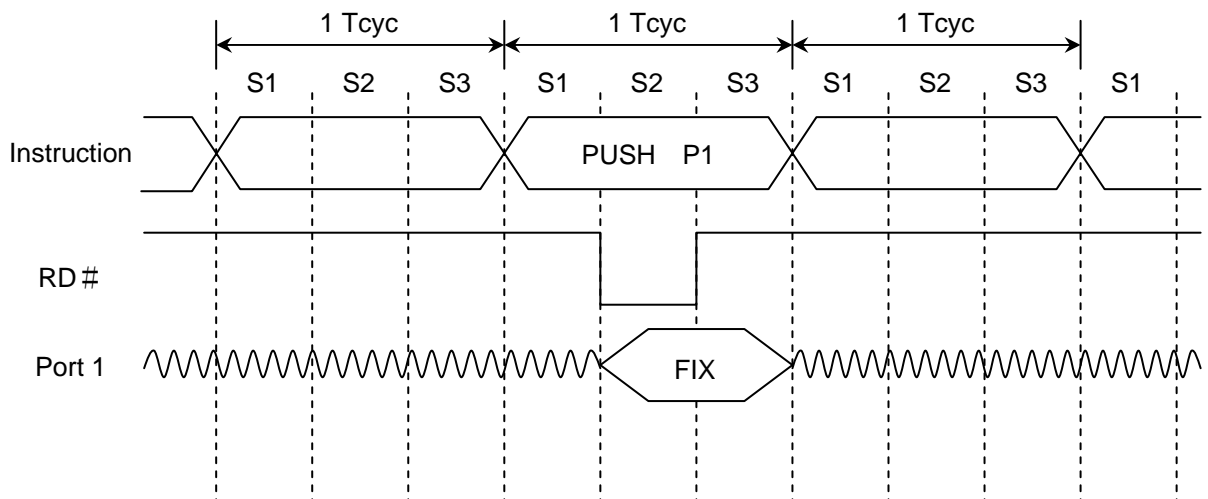
<2> External memory write mode

- Execute an instruction (e.g., POP) for writing data into P1 to output a write signal to P23 at the timing of S1.

<Write mode timing chart>
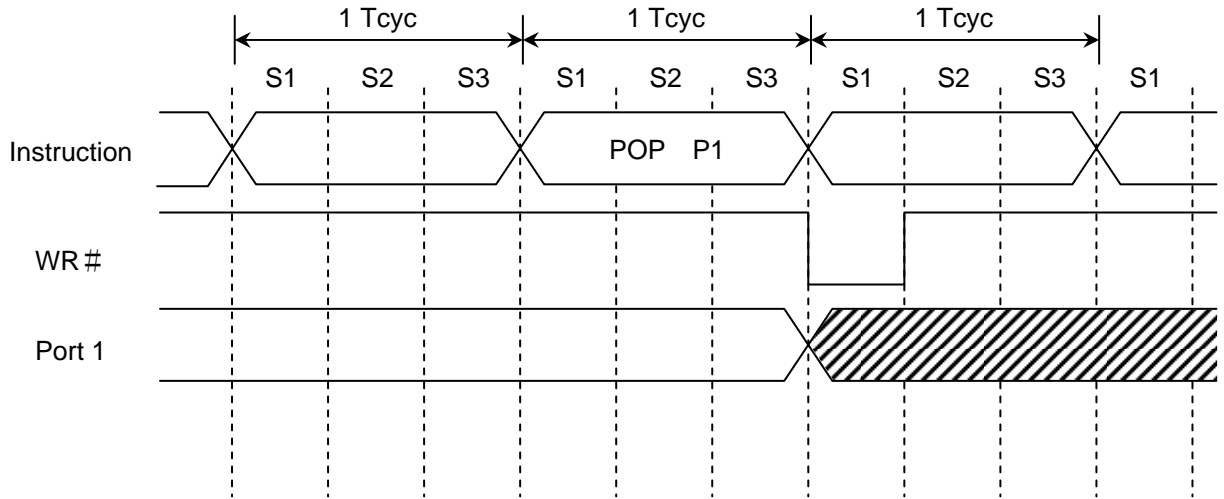
3.15 Asynchronous Serial Interface 1 (UART1)

3.15.1 Overview

This series of microcontrollers incorporates an asynchronous serial interface 1 (UART1) that has the following characteristics and functions:

1) Data length: 7/8/9 bits (LSB first)

2) Stop bits: 1 bit (2 bits in continuous transmit mode)

3) Parity bits: None

4) Transfer rate: $\frac{16}{3}$ to $\frac{8192}{3}$ Tcyc

5) Full duplex communication

The independent transmitter and receiver modules allow both transmit and receive operations to be performed at the same time. Both transmitter and receiver modules have a double buffer configuration, so that data can be transmitted and received continuously.

3.15.2 Functions

1) Asynchronous serial (UART1)

- Performs full duplex asynchronous serial communication using a data length of 7, 8, or 9 bits with 1 stop bit.

- The transfer rate of UART1 is programmable within the range of $\frac{16}{3}$ to $\frac{8192}{3}$ Tcyc.

2) Continuous data transmission/reception

- Performs continuous transmission of serial data whose data length and transfer rate are fixed (the data length and transfer rate that are identified at the beginning of transmit operation are used). The number of stop bits used in the continuous transmit mode is 2. (See Figure 3.15.4)

- Performs continuous reception of serial data whose data length and transfer rate vary on each receive operation.

- The transfer rate of UART1 is programmable within the range of $\frac{16}{3}$ to $\frac{8192}{3}$ Tcyc.

- The transmit data is read from the transmit data register (TBUF) and the received data is stored in the receive data register (RBUF).

3) Interrupt generation

Interrupt requests are generated at the beginning of transmit operation and at the end receive operation if the interrupt request enable bit is set.

4) To control the asynchronous serial interface 1 (UART1), it is necessary to manipulate the following special function registers.

- UCON0, UCON1, UBR, TBUF, RBUF
- P2, P2DDR

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED0	0000 0000	R/W	UCON0	UBRSEL	STRDET	RECRUN	STPERR	U0B3	RBIT8	RECEND	RECIE
FED1	0000 0000	R/W	UCON1	TRUN	8/9BIT	TDDR	TCMOS	8/7BIT	TBIT8	TEPTY	TRNSIE
FED2	0000 0000	R/W	UBR	UBRG7	UBRG6	UBRG5	UBRG4	UBRG3	UBRG2	UBRG1	UBRG0
FED3	0000 0000	R/W	TBUF	T1BUF7	T1BUF6	T1BUF5	T1BUF4	T1BUF3	T1BUF2	T1BUF1	T1BUF0
FED4	0000 0000	R/W	RBUF	R1BUF7	R1BUF6	R1BUF5	R1BUF4	R1BUF3	R1BUF2	R1BUF1	R1BUF0

3.15.3 Circuit Configuration

3.15.3.1 UART1 control register 0 (UCON0)   (8-bit register)
1) This register controls the receive operation and interrupts of UART1.

3.15.3.2 UART1 control register 1 (UCON1)   (8-bit register)
1) This register controls the transmit operation, data length, and interrupts of UART1.

3.15.3.3 UART1 baudrate generator (UBR)   (8-bit reload counter)
1) The UART1 baudrate generator is a reload counter for generating internal clocks.
2) It can generate clocks at intervals of $(n+1) \times \frac{8}{3}$ Tcyc, or $(n+1) \times \frac{32}{3}$ Tcyc (n = 1 to 255; Note: n = 0 is inhibited).

3.15.3.4 UART1 transmit data register (TBUF)   (8-bit register)
1) This register is an 8-bit register for storing the data to be transmitted.

3.15.3.5 UART1 transmit shift register (TSFT)   (11-bit shift register)
1) This register is used to send transmit data via UART1.
2) This register cannot be accessed directly with an instruction. It must be accessed through the transmit data register (TBUF).

3.15.3.6 UART1 receive data register (RBUF)   (8-bit register)
1) This is an 8-bit register for storing received data.

3.15.3.7 UART1 receive shift register (RSFT)   (11-bit shift register)
1) This register is used to receive serial data via UART1.
2) This register cannot be accessed directly with an instruction. It must be accessed through the receive data register (RBUF).
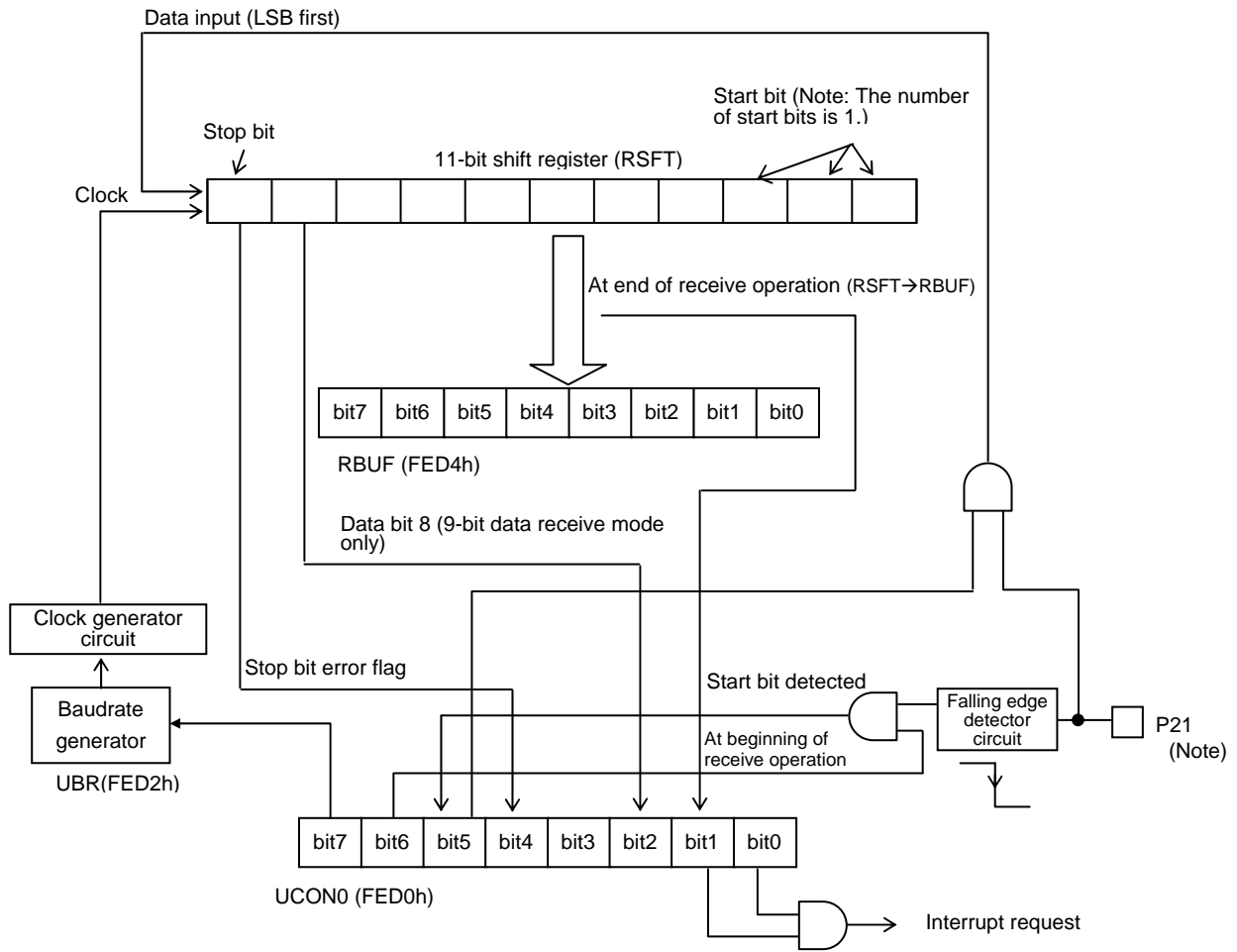
Note: Bit 1 of P2DDR (at FE49H) must be set to 0 when UART1 is to be used in the receive mode.
(UART1 will not operate normally if bit 1 is set to 1)

Figure 3.15.1   UART1 Block Diagram (Receive Mode)
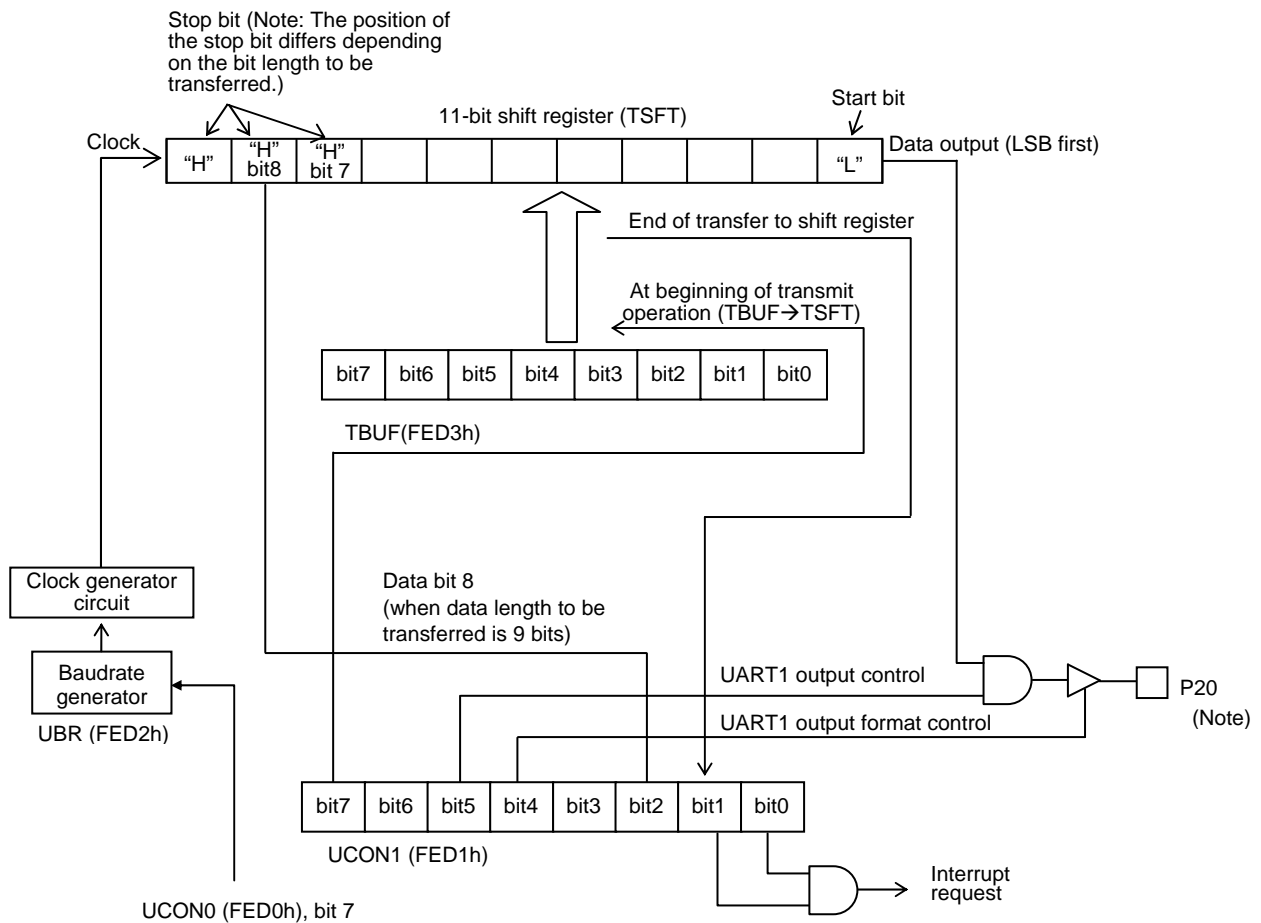
Note: Bit 0 of P2DDR (at FE49H) must be set to 0 when UART1 transmit data is to be output. (Transmit data is not output if bit 0 is set to 1)

Figure 3.15.2   UART1 Block Diagram (Transmit Mode)

3.15.4   Related Registers

3.15.4.1   UART1 control register 0 (UCON0)

1)   This register is an 8-bit register that controls the receive operation and interrupts of UART1.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED0	0000 0000	R/W	UCON0	UBRSEL	STRDET	RECRUN	STPERR	U0B3	RBIT8	RECEND	RECIE

UBRSEL (bit 7): UART1 transfer rate select flag

1)   When this bit is set to 1, the UART1 transfer rate range is set to $\frac{64}{3}$ to $\frac{8192}{3}$ Tcyc.

2)   When this bit is set to 0, the UART1 transfer rate range is set to $\frac{16}{3}$ to $\frac{2048}{3}$ Tcyc.

STRDET (bit 6): UART1 start bit detection control

1)   When this bit is set to 1, the start bit detection (falling edge detection) function is enabled.

2)   When this bit is set to 0, the start bit detection (falling edge detection) function is disabled.

* This bit must be set to 1 to enable the start bit detection function when UART1 is to be used in the continuous receive mode.

* If this bit is set to 1 when the receive port (P21) is held at the low level, RECRUN is automatically set and UARR 1 starts a receive operation.

RECRUN (bit 5): UART1 receive start flag

1) This bit is set and a receive operation starts when a falling edge of the signal at receive port (P21) is detected when the start bit detection function is enabled (STRDET = 1).

2) This bit is automatically cleared at the end of the receive operation (clearing this bit during a receive operation will abort the receive operation).

* When a receive operation is forced to end prematurely, RECEND is set to 1 and the contents of the receive shift register are transferred to RBUF. STPERR is set to 1 if the state of the last data bit that is received on the forced termination is low.

STPERR (bit 4): UART1 stop bit error flag

1) This bit is set at the end of a receive operation if the state of the received stop bit (the last data bit received) is low.

2) This bit must be cleared with an instruction.

* When clearing STPERR during a continuous receive operation, clear it with an instruction before receiving the last bit of the next receive data following the branch to the interrupt routine.

U0B3 (bit 3): General-purpose flag

1) This bit can be used as a general-purpose flag bit. Any attempt to manipulate this bit exerts no influence on the operation of this functional block.

RBIT8 (bit 2): UART1 receive data bit 8 storage bit

1) This bit position is loaded with bit 8 of the received data at the end of the receive operation when the data length is set to 9 bits (UCON1: 8/9BIT=1). (If the receive operation is terminated prematurely, this bit position is loaded with the last received bit but one.)

2) This bit must be cleared with an instruction.

RECEND (bit 1): UART1 end of reception flag

1) This bit is set at the end of a receive operation (When this bit is set, the received data is transferred from the receive shift register (RSFT) to the receive data register (RBUF)).

2) This bit must be cleared with an instruction.

* When clearing RECEND during a continuous receive operation, clear it with an instruction before receiving the last bit of the next receive data following the branch to the interrupt routine.

RECIE (bit 0): UART1 receive interrupt request enable control

1) When this bit and RECEND are set to 1, an interrupt request to vector address 0033H is generated.

Note: Once a receive operation is started with STRDET (bit 6) set to 1, do not execute any bit-manipulation instruction (CLR1, NOT1, or SET1) on this register until the RECEND bit (bit 1) is set to 1 and the receive operation is ended, except when it becomes necessary to force the receive operation to stop. The bit-manipulation instructions, however, can be executed within the interrupt routine that is started by the UART1 receive interrupt source.

3.15.4.2 UART1 control register 1 (UCON1)

1) This register is an 8-bit register that controls the transmit operation, data length, and interrupts of UART1.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED1	0000 0000	R/W	UCON1	TRUN	8/9BIT	TDDR	TCMOS	8/7BIT	TBIT8	TEPTY	TRNSIE

TRUN (bit 7): UART1 transmission control

1) When this bit is set to 1, UART1 starts a transmit operation.

2) This bit is automatically cleared at the end of the transmit operation. (If this bit is cleared during a transmit operation, UART1 is terminated in the middle of operation.)

   * In the continuous transmit mode, this bit is cleared at the end of a transmit operation but is automatically set within the same cycle (Tcyc). Consequently, a transmit operation is performed with intervening 1-Tcyc waits.

   * In the continuous transmit mode, TRUN will not be set automatically if a bit-manipulation-instruction (NOT1, CLR1, or SET1) is executed on the UCON1 register in the same cycle in which TRUN is to be automatically cleared.

8/9BIT (bit 6): UART1 transfer data length control

1) When this bit is set to 1, the data length of UART1 is set to 9 bits.

2) When this bit is set to 0 and 8/7BIT (bit 3) is set to 0, the data length of UART1 is set to 8 bits.

3) When this bit is set to 0 and 8/7BIT (bit 3) is set to 1, the data length of UART1 is set to 7 bits.

   * The UART1 will not run normally if the data length is changed in the middle of a transfer operation. Be sure to manipulate this bit after confirming the completion of a transfer operation.

   * The same data length is used when both transmit and receive operations are to be performed at the same time.

8/9 BIT	8/7 BIT	Data Length (in bits)
0	0	8
0	1	7
1	X	9

TDDR (bit 5): UART1 transmit port output control

1) When this bit is set to 1, the transmit data is placed at the transmit port (P20). (No transmit data is generated if bit 0 of P2DDR (at FE49H) is set to 1.)

2) When this bit is set to 0, no transmit data is placed at the transmit port (P20).

   * The transmit port is placed in the high/open (CMOS/N-channel open drain) mode if this bit is set to 1 when UART1 has stopped a transmit operation (TRUN = 0).

   * This bit must always be set to 0 when the UART1 transmission function is not to be used.

TCMOS (bit 4): UART1 transmit port output type control

    1)   When this bit is set to 1, the output type of the transmit port (P20) is set to CMOS.

    2)   When this bit is set to 0, the output type of the transmit port (P20) is set to N-channel open drain.

8/7BIT (bit 3): UART1 transfer data length control

    1)   See the bit description on 8/9BIT (bit 6).

TBIT8 (bit 2): UART1 transmit data bit 8 storage bit

    1)   This bit carries bit 8 of the transmit data when the data length is set to 9 bits (8/9BIT = 1).

TEPTY (bit 1): UART1 transmit shift register transfer flag

    1)   This bit is set when data transfer from the transmit data register (TBUF) to the transmit shift register (TSFT) ends after the start of transmit operation. (This bit is set in the cycle (Tcyc) following the one in which the transmit control bit (TRUN) is set to 1.)

    2)   This bit must be cleared with an instruction.

    *   When performing continuous mode transmit operation, make sure that this bit is set before each loading of the next transmit data into the transmit data register (TBUF). When this bit is subsequently cleared, the transmit control bit (TRUN) is automatically set at the end of the transmit operation.

TRNSIE (bit 0): UART1 transmit interrupt request enable control

    1)   An interrupt request to vector address 003BH is generated when this bit and TEPTY are set to 1.

Note: Once a transmit operation is started with TRUN (bit 7) set to 1, do not execute any bit-manipulation instruction (CLR1, NOT1, or SET1) on this register until the TRUN bit (bit 7) is set to 0 and the transmit operation is terminated, except when it becomes necessary to force the transmit operation to stop. The bit-manipulation instructions, however, can be executed within the interrupt routine that is started by the UART1 transmit interrupt source.

3.15.4.3 UART1 baudrate generator (UBR)

1) The UART1 baudrate generator is an 8-bit register that defines the transfer rate of UART1.

2) The counter for the baudrate generator is initialized when a UART1 serial transfer operation is stopped or terminated (UCON0: RECRUN=0, UCON1: TRUN=0).

3) The value range of the transfer rate can be selected by the value of UBRSEL.

UBRSEL	TUBR1	Value Range
0	(UBR value + 1) $\times \frac{8}{3}$ Tcyc	$\frac{16}{3}$ to $\frac{2048}{3}$ Tcyc
1	(UBR value + 1) $\times \frac{32}{3}$ Tcyc	$\frac{64}{3}$ to $\frac{8192}{3}$ Tcyc

* Do not change the baudrate in the middle of UART1 serial transfer operation. UART1 will not function normally if the baudrate is changed during the transfer operation. Always make sure that the transfer operation has ended before changing the baudrate.

* The same baudrate is used when both transmit and receive operations are to be performed at the same time (this also applies when continuous data transmission/reception is performed at the same time).

* Setting UBR to 00[H] is inhibited.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED2	0000 0000	R/W	UBR	UBRG7	UBRG6	UBRG5	UBRG4	UBRG3	UBRG2	UBRG1	UBRG0

3.15.4.4 UART1 transmit data register (TBUF)

1) This register is an 8-bit register that stores the data to be transmitted through UART1.

2) Data from the TBUF is transferred to the transmit shift register (TSFT) at the beginning of a transmit operation. (Load the next data after checking the transmit shift register transfer flag (UCON1: TEPTY).)

* Bit 8 of the transmit data must be loaded into the transmit data bit 8 storage bit (UCON1:TBIT8).

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED3	0000 0000	R/W	TBUF	T1BUF7	T1BUF6	T1BUF5	T1BUF4	T1BUF3	T1BUF2	T1BUF1	T1BUF0

3.15.4.5 UART1 receive data register (RBUF)

1) This register is an 8-bit register that stores the data that is received through UART1.

2) The data from the receive shift register (RSFT) is transferred to this RBUF at the end of a receive operation.

* Bit 8 of the received data is transferred in the receive data bit 8 storage bit (UCON0:RBIT8).

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED4	0000 0000	R/W	RBUF	R1BUF7	R1BUF6	R1BUF5	R1BUF4	R1BUF3	R1BUF2	R1BUF1	R1BUF0

3.15.5 UART1 Continuous Communication Processing Examples

3.15.5.1 Continuous 8-bit data receive mode (first received data = 55H)
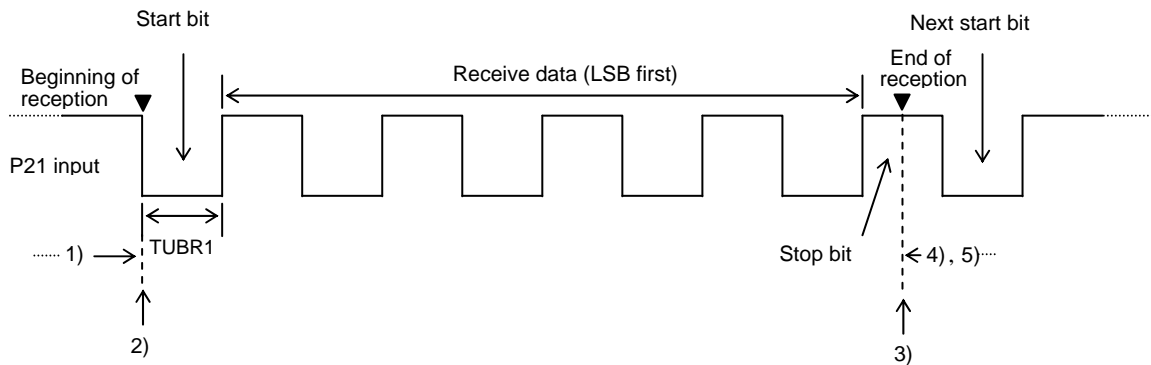
Figure 3.15.3 Example of Continuous 8-bit Data Receive Mode Processing

1) Setting the clock
    - Set the transfer rate (UBR).

    Setting the data length
    - Clear UCON1:8/9BIT.

    Preparing for a receive operation, setting the receive port and interrupts
    - Set up the receive control register (UCON0 = 41H).
    * Set P21DDR (P2DDR:bit 1) to 0 and P21 (P2:bit 1) to 0.

2) Starting a receive operation
    - UCON0:RECRUN is set when a falling edge of the signal at the receive port (P21) is detected.

3) End of a receive operation
    - When the receive operation ends, UCON0:RECRUN is automatically cleared and UCON0:RECEND is set. The UART1 then waits for the start bit of the next received data.

4) Receive interrupt processing
    - Read the received data (RBUF).
    - Clear UCON0:RECEND and STPERR and exit the interrupt processing routine.
    * When changing the data length and transfer rate for the next receive operation, do so before the start bit (falling edge of the signal) is detected at the receive port (P21).

5) Next receive data processing
    - Subsequently, repeat steps 2), 3), and 4) shown above.
    - To end the continuous receive operation, clear UCON0:STRDET during a receive operation, and this receive operation will be the last receive operation that the UART1 executes.

3.15.5.2 Continuous 8-bit data transmit mode (first transmit data = 55H)
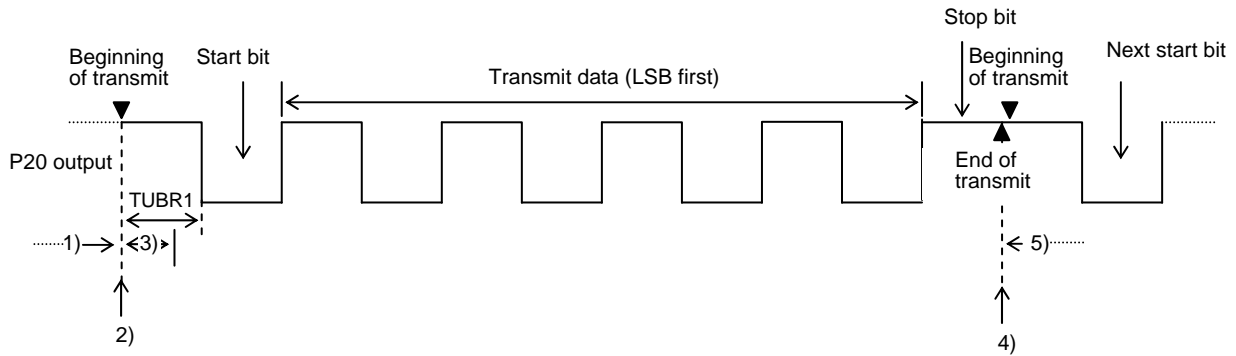
Figure 3.15.4   Example of Continuous 8-bit Data Transmit Mode Processing

1) Setting the clock
   - Set the transfer rate (UBR).

   Setting up transmit data
   - Load the transmit data (TBUF =55H).

   Setting the data length, transmit port, and interrupts
   - Set up the transmit control register (UCON1 = 31H).
   * Set P20DDR (P2DDR:bit 0) to 0 and P20 (P2:bit 0) to 0.

2) Starting a transmit operation
   - Set UCON1:TRUN.

3) Transmit interrupt processing
   - Load the next transmit data (TBUF = xxH).
   - Clear UCON1:TEPTY and exit the interrupt processing routine.

4) End of a transmit operation
   - When the transmit operation ends, UCON1:TRUN is automatically cleared and automatically set in the same cycle (Tcyc) (continuous data transmit mode only; this processing takes 1 Tcyc of time), after which the next transmit operation starts.

5) Next transmit operation
   - Subsequently, repeat steps 3) and 4) shown above.
   - To end the continuous transmit operation, clear UCON1:TRNSIE while not clearing UCON1:TEPTY and exit the interrupt in the step 3) processing, and the transmit operation that is being performed at that time will be the last transmit operation that UART1 executes.

3.15.5.3 Setting up the UART1 communications ports

1) Receive port (P21) settings

Register Data		Receive Port (P21) State	Internal Pull-up Resistor
P21	**P21DDR**		
0	0	Input	Off
1	0	Input	On

* The UART1 cannot receive data normally if P21DDR is set to 1.

2) Transmit port (P20) settings

Register Data				Transmit Port (P20) State	Internal Pull-up Resistor
P20	**P20DDR**	**TDDR**	**TCMOS**		
0	0	1	1	CMOS output	Off
0	0	1	0	N-channel open drain output	Off
1	0	1	0	N-channel open drain output	On

* The UART1 transmits no data if P20DDR is set to 1.

3.15.6 UART1 HALT Mode Operation

3.15.6.1 Receive mode

1) UART1 receive mode processing is enabled in HALT mode. (If UCON0:STRDET is set to 1 when the microcontroller enters HALT mode, receive operation will be restarted if data that sets UCON0:RECRUN is input after the end of a receive operation.)

2) HALT mode can be released using the UART1 receive interrupt.

3.15.6.2 Transmit mode

1) UART1 transmit mode processing is enabled in HALT mode. (If the continuous transmit mode is specified when the microcontroller enters HALT mode, the UART1 will restart transmit processing after terminating a transmit operation. Since UCON1: TEMPTY cannot be cleared in this case, the UART1 stops processing after completing that transmit operation.)

2) HALT mode can be released using the UART1 transmit interrupt.

3.16 Asynchronous Serial Interface 2 (UART2)

3.16.1 Overview

This series of microcontrollers incorporates an asynchronous serial interface 2 (UART2) that has the following characteristics and functions:

1) Data length: 7/ 8/ 9 bits (LSB first)

2) Stop bits: 1 bit (2 bits in continuous transmit mode)

3) Parity bits: None

4) Transfer rate: $\frac{16}{3}$ to $\frac{8192}{3}$ Tcyc

5) Full duplex communication

The independent transmitter and receiver modules allow both transmit and receive operations to be performed at the same time. Both transmitter and receiver modules have a double buffer configuration, so that data can be transmitted and received continuously.

3.16.2 Functions

1) Asynchronous serial (UART2)

- Performs full duplex asynchronous serial communication using a data length of 7, 8, or 9 bits with 1 stop bit.

- The transfer rate of UART2 is programmable within the range of $\frac{16}{3}$ to $\frac{8192}{3}$ Tcyc.

2) Continuous data transmission/reception

- Performs continuous transmission of serial data whose data length and transfer rate are fixed (the data length and transfer rate that are identified at the beginning of transmit operation are used). The number of stop bits used in the continuous transmit mode is 2. (See Figure 3.16.4)

- Performs continuous reception of serial data whose data length and transfer rate vary on each receive operation.

- The transfer rate of UART2 is programmable within the range of $\frac{16}{3}$ to $\frac{8192}{3}$ Tcyc.

- The transmit data is read from the transmit data register (TBUF2) and the received data is stored in the receive data register (RBUF2).

3) Interrupt generation

Interrupt requests are generated at the beginning of transmit operation and at the end of receive operation if the interrupt request enable bit is set.

4) To control the asynchronous serial interface 2 (UART2), it is necessary to manipulate the following special function registers.

- UCON2, UCON3, UBR2, TBUF2, RBUF2

- P2, P2DDR

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEE8	0000 0000	R/W	UCON2	UBRSEL2	STRDET2	RECRUN2	STPERR2	U2B3	RBIT82	RECEND2	RECIE2
FEE9	0000 0000	R/W	UCON3	TRUN2	8/9BIT2	TDDR2	TCMOS2	8/7BIT2	TBIT82	TEPTY2	TRNSIE2
FEEA	0000 0000	R/W	UBR2	U2BRG7	U2BRG6	U2BRG5	U2BRG4	U2BRG3	U2BRG2	U2BRG1	U2BRG0
FEEB	0000 0000	R/W	TBUF2	T2BUF7	T2BUF6	T2BUF5	T2BUF4	T2BUF3	T2BUF2	T2BUF1	T2BUF0
FEEC	0000 0000	R/W	RBUF2	R2BUF7	R2BUF6	R2BUF5	R2BUF4	R2BUF3	R2BUF2	R2BUF1	R2BUF0

3.16.3   Circuit Configuration

3.16.3.1   UART2 control register 2 (UCON2)   (8-bit register)
   1)   This register controls the receive operation and interrupts of UART2.

3.16.3.2   UART2 control register 3 (UCON3)   (8-bit register)
   1)   This register controls the transmit operation, data length, and interrupts of UART2.

3.16.3.3   UART2 baudrate generator (UBR2)   (8-bit reload counter)
   1)   The UART2 baudrate generator is a reload counter for generating internal clocks.

   2)   It can generate clocks at intervals of $(n+1) \times \frac{8}{3}$ Tcyc   or $(n+1) \times \frac{32}{3}$ Tcyc (n = 1 to 255; Note: n = 0 is inhibited).

3.16.3.4   UART2 transmit data register (TBUF2)   (8-bit register)
   1)   This register is an 8-bit register for storing the data to be transmitted.

3.16.3.5   UART2 transmit shift register (TSFT2)   (11-bit shift register)
   1)   This register is used to transmit serial data via UART2.

   2)   This register cannot be accessed directly with an instruction. It must be accessed through the transmit data register (TBUF2).

3.16.3.6   UART2 receive data register (RBUF2)   (8-bit register)
   1)   This register is an 8-bit register for storing received data.

3.16.3.7   UART2 receive shift register (RSFT2)   (11-bit shift register)
   1)   This register is used to receive serial data via UART2.

   2)   This register cannot be accessed directly with an instruction. It must be accessed through the receive data register (RBUF2).

UART2
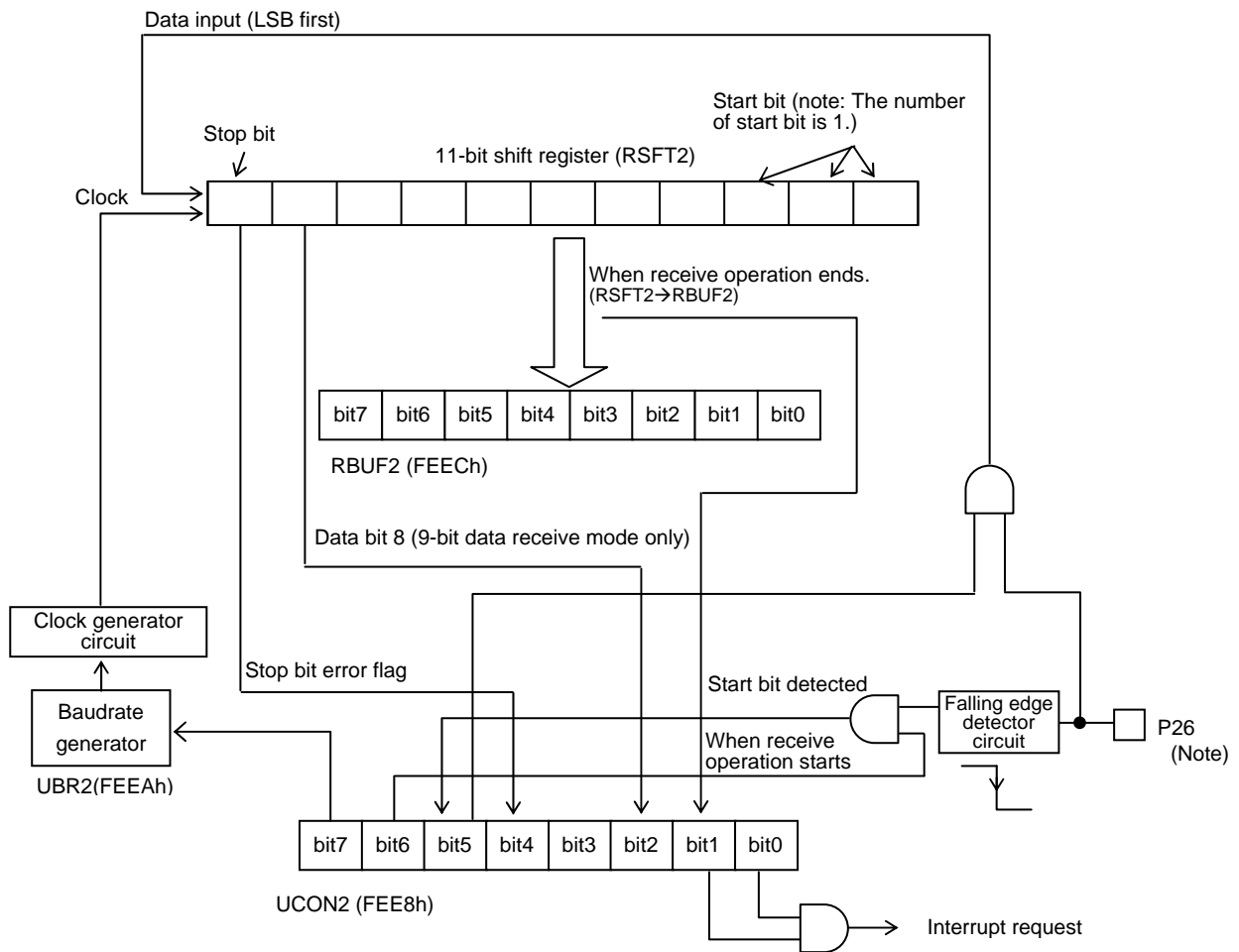
Note: Bit 6 of P2DDR (at FE49H) must be set to 0 when UART2 is to be used in the receive mode.
(UART2 will not operate normally if bit 6 is set to 1)

Figure 3.16.1   UART2 Block Diagram (Receive Mode)
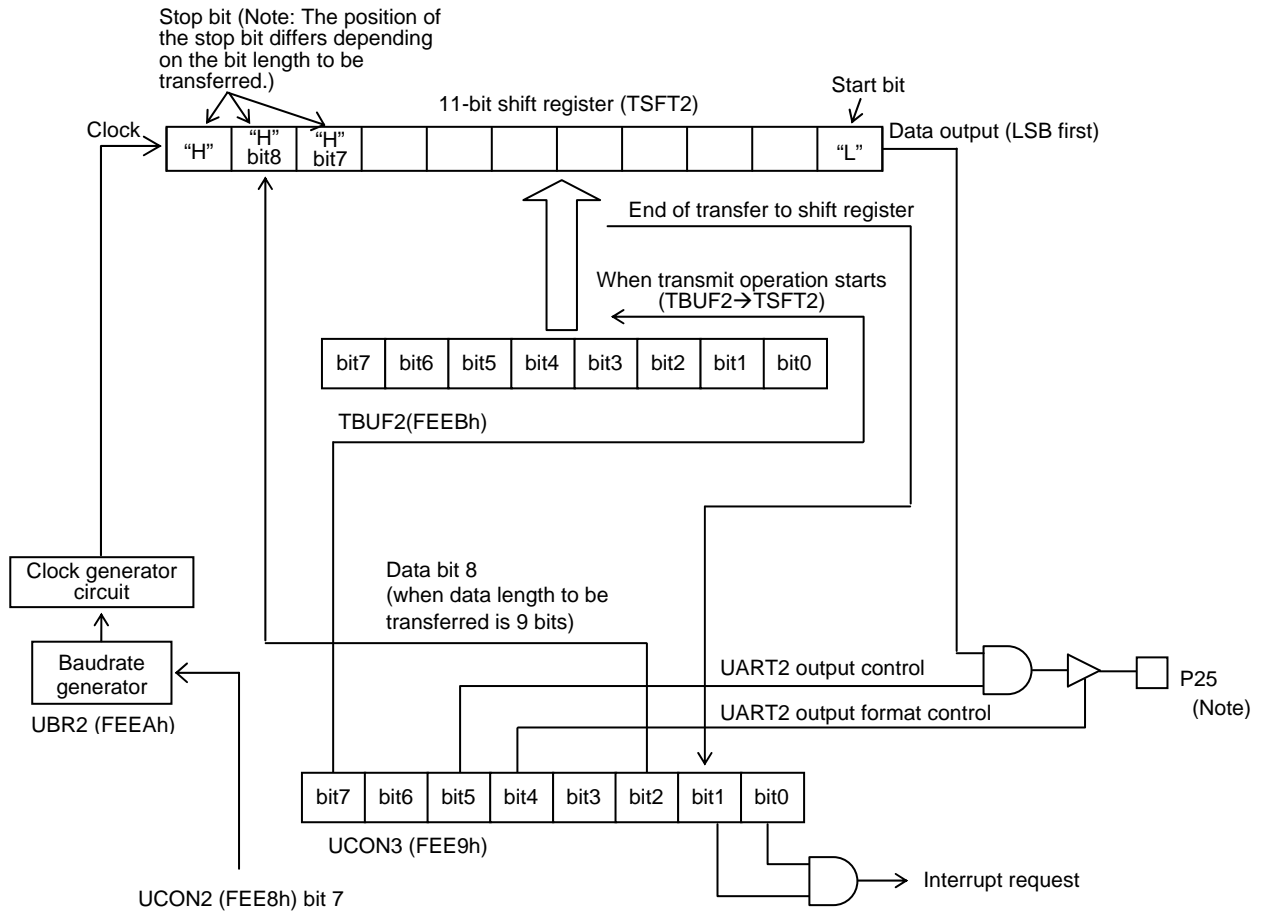
Note: Bit 5 of P2DDR (at FE49H) must be set to 0 when UART2 transmit data is to be output.
(Transmit data is not output if bit 5 is set to 1)

Figure 3.16.2   UART2 Block Diagram (Transmit Mode)

3.16.4   Related Registers

3.16.4.1   UART2 control register 2 (UCON2)

1)   This register is an 8-bit register that controls the receive operation and interrupts of UART2.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEE8	0000 0000	R/W	UCON2	UBRSEL2	STRDET2	RECRUN2	STPERR2	U2B3	RBIT82	RECEND2	RECIE2

UBRSEL2 (bit 7): UART2 transfer rate select flag

1)   When this bit is set to 1, the UART2 transfer rate range is set to $\frac{64}{3}$ to $\frac{8192}{3}$ Tcyc.

2)   When this bit is set to 0, the UART2 transfer rate range is set to $\frac{16}{3}$ to $\frac{2048}{3}$ Tcyc.

STRDET2 (bit 6): UART2 start bit detection control

1)   When this bit is set to 1, the start bit detection (falling edge detection) function is enabled.

2)   When this bit is set to 0, the start bit detection (falling edge detection) function is disabled.

UART2

    * This bit must be set to 1 to enable the start bit detection function when UART2 is to be used in the continuous receive mode.

    * If this bit is set to 1 when the receive port (P26) is held at the low level, RECRUN2 is automatically set and UART2 starts a receive operation.

RECRUN2 (bit 5): UART2 receive start flag

1) This bit is set and the receive operation starts when a falling edge of the signal at receive port (P26) is detected when the start bit detection function is enabled (STRDET2 = 1).

2) This bit is automatically cleared at the end of the receive operation (clearing this bit during a receive operation will abort the receive operation).

    * When a receive operation is forced to end prematurely, RECEND2 is set to 1 and the contents of the receive shift register are transferred to RBUF2. STPERR2 is set to 1 if the state of the last data bit that is received on the forced termination is low.

STPERR2 (bit 4): UART2 stop bit error flag

1) This bit is set at the end of a receive operation if the state of the received stop bit (the last data bit received) is low.

2) This bit must be cleared with an instruction.

    * When clearing STPERR2 during a continuous receive operation, clear it with an instruction before receiving the last bit of the next receive data following the branch to the interrupt routine.

U2B3 (bit 3): General-purpose flag

1) This bit can be used as a general-purpose flag bit. Any attempt to manipulate this bit exerts no influence on the operation of this functional block.

RBIT82 (bit 2): UART2 receive data bit 8 storage bit

1) This bit position is loaded with bit 8 of the received data when the data length is set to 9 bits (UCON3: 8/9BIT2=1). (If the receive operation is terminated prematurely, this bit position is loaded with the last received bit but one.)

2) This bit must be cleared with an instruction.

RECEND2 (bit 1): UART2 receive end flag

1) This bit is set at the end of a receive operation (When this bit is set, the received data is transferred from the receive shift register (RSFT2) to the receive data register (RBUF2)).

2) This bit must be cleared with an instruction.

    * When clearing RECEND2 during a continuous receive operation, clear it with an instruction before receiving the last bit of the next receive data following the branch to the interrupt routine.

RECIE2 (bit 0): UART2 receive interrupt request enable control

1) When this bit and RECEND2 are set to 1, an interrupt request to vector address 0033H is generated.

Note: Once a receive operation is started with STRDET2 (bit 6) set to 1, do not execute any bit-manipulation instruction (CLR1, NOT1, or SET1) on this register until the RECEND2 bit (bit 1) is set to 1 and the receive operation is ended, except when it becomes necessary to force the receive operation to stop. The bit-manipulation instructions, however, can be executed within the interrupt routine that is started by the UART2 receive interrupt source.

3.16.4.2 UART2 control register 3 (UCON3)

1) This register is an 8-bit register that controls the transmit operation, data length, and interrupts of UART2.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEE9	0000 0000	R/W	UCON3	TRUN2	8/9BIT2	TDDR2	TCMOS2	8/7BIT2	TBIT82	TEPTY2	TRNSIE2

TRUN2 (bit 7): UART2 transmit control

1) When this bit is set to 1, UART2 starts a transmit operation.

2) This bit is automatically cleared at the end of the transmit operation. (clearing this bit during a transmit operation will abort the transmit operation.)

   * In the continuous transmit mode, this bit is cleared at the end of a transmit operation but is automatically set within the same cycle (Tcyc). Consequently, transmit operations is performed with intervening 1-Tcyc waits.

   * In the continuous transmit mode, TRUN2 will not be set automatically if a bit-manipulation-instruction (NOT1, CLR1, or SET1) is executed to the UCON3 register in the same cycle in which TRUN2 is to be automatically cleared.

8/9 BIT2 (bit 6): UART2 transfer data length control

1) When this bit is set to 1, the data length of UART2 is set to 9 bits.

2) When this bit is set to 0 and 8/7BIT2 (bit 3) is set to 0, the data length of UART2 is set to 8 bits.

3) When this bit is set to 0 and 8/7BIT2 (bit 3) is set to 1, the data length of UART2 is set to 7 bits.

   * The UART2 will not run normally if the data length is changed in the middle of a transfer operation. Be sure to manipulate this bit after confirming the completion of the transfer operation.

   * The same data length is used when both transmit and receive operations are to be performed at the same time.

8/9 BIT2	8/7 BIT2	Data Length (in bits)
0	0	8
0	1	7
1	X	9

TDDR2 (bit 5): UART2 transmit port output control

1) When this bit is set to 1, the transmit data is placed at the transmit port (P25). (No transmit data is generated if bit 5 of P2DDR (at FE49H) is set to 1.)

2) When this bit is set to 0, no transmit data is placed at the transmit port (P25).

   * The transmit port is placed in the "high/open (CMOS/N-channel open drain)" mode if this bit is set to 1 when UART2 has stopped a transmit operation (TRUN2 = 0).

   * This bit must always be set to 0 when the UART2 transmit function is not to be used.

TCMOS2 (bit 4): UART2 transmit port output type control
1)   When this bit is set to 1, the output type of the transmit port (P25) is set to CMOS.
2)   When this bit is set to 0, the output type of the transmit port (P25) is set to N-channel open drain.

8/7 BIT2 (bit 3): UART2 transfer data length control
1)   Refer to 8/9 BIT2 (bit 6) above.

TBIT82 (bit 2): UART2 transmit data bit 8 storage bit
1)   This bit carries bit 8 of the transmit data when the data length is set to 9 bits (8/9BIT2 = 1).

TEPTY2 (bit 1): UART2 transmit shift register transfer flag
1)   This bit is set when data transfer from the transmit data register (TBUF2) to the transmit shift register (TSFT2) ends after the start of a transmit operation. (This bit is set on the cycle (Tcyc) following the one in which the transmit control bit (TRUN2) is set to 1.)
2)   This bit must be cleared with an instruction.
  *   When performing continuous transmit operation, make sure that this bit is set before each loading of the next transmit data into the transmit data register (TBUF2). When this bit is subsequently cleared, the transmit control bit (TRUN2) is automatically set at the end of the transmit operation.

TRNSIE2 (bit 0): UART2 transmit interrupt request enable control
1)   An interrupt request to vector address 003BH is generated when this bit and TEPTY2 are set to 1.

Note: Once a transmit operation is started with TRUN2 (bit 7) set to 1, do not execute any bit-manipulation instruction (CLR1, NOT1, or SET1) on this register until the TRUN2 bit (bit 7) is set to 0 and the transmit operation is terminated, except when it becomes necessary to force the transmit operation to stop. The bit-manipulation instructions, however, can be executed within the interrupt routine that is started by the UART2 transmit interrupt source.

3.16.4.3 UART2 baudrate generator (UBR2)

1) The UART2 baudrate generator is an 8-bit register that defines the transfer rate of UART2 transfer.

2) The counter for the baudrate generator is initialized when a UART2 serial transfer operation is stopped or terminated (UCON2:RECRUN2 = 0, UCON3:TRUN2=0).

3) The value range of the transfer rate can be changed by the value of UBRSEL2.

UBRSEL2	TUBR2	Value Range
0	(UBR2 value + 1) $\times \frac{8}{3}$ Tcyc	$\frac{16}{3}$ to $\frac{2048}{3}$ Tcyc
1	(UBR2 value + 1) $\times \frac{32}{3}$ Tcyc	$\frac{64}{3}$ to $\frac{8192}{3}$ Tcyc

* Do not change the transfer rate in the middle of UART2 serial transfer operation. UART2 will not function normally if the transfer rate is changed. Always make sure that the transfer operation has ended before changing the baudrate.

* The same transfer rate is used when both transmit and receive operations are to be performed at the same time (this also applies when continuous data transmission/reception is performed at the same time).

* Setting UBR2 to 00[H] is inhibited.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEEA	0000 0000	R/W	UBR2	U2BRG7	U2BRG6	U2BRG5	U2BRG4	2UBRG3	U2BRG2	U2BRG1	U2BRG0

3.16.4.4 UART2 transmit data register (TBUF2)

1) This register is an 8-bit register that stores the data to be transmitted through the UART2.

2) Data from the TBUF2 is transferred to the transmit shift register (TSFT2) at the beginning of a transmit operation. (Load the next data after checking the transmit shift register transfer flag (UCON3:TEPTY2).)

* Bit 8 of the transmit data must be loaded into the transmit data bit 8 storage bit (UCON3:TBIT82).

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEEB	0000 0000	R/W	TBUF2	T2BUF7	T2BUF6	T2BUF5	T2BUF4	T2BUF3	T2BUF2	T2BUF1	T2BUF0

3.16.4.5 UART2 receive data register (RBUF2)

1) This register is an 8-bit register that stores the data that is received through UART2.

2) The data from the receive shift register (RSFT2) is transferred to this RBUF2 at the end of a receive operation.

* Bit 8 of the received data is transferred to the receive data bit 8 storage bit (UCON2:RBIT82).

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEEC	0000 0000	R/W	RBUF2	R2BUF7	R2BUF6	R2BUF5	R2BUF4	R2BUF3	R2BUF2	R2BUF1	R2BUF0

3.16.5   UART2 Continuous Communication Processing Examples

3.16.5.1   Continuous 8-bit data receive mode (first received data = 55H)
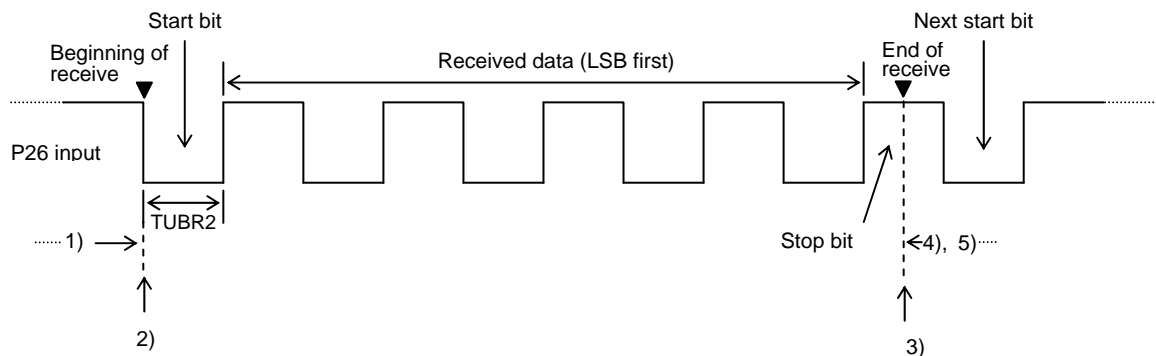
Figure 3.16.3   Example of Continuous 8-bit Data Receive Mode Processing

1)   Setting the clock
   • Set the transfer rate (UBR2).

   Setting the data length
   • Clear UCON3:8/9BIT2.

   Preparing for a receive operation, setting the receive port and interrupts
   • Set up the receive control register (UCON2 = 41H).
   * Set P26DDR (P2DDR:bit 6) to 0 and P26 (P2:bit 6) to 0.

2)   Starting a receive operation
   • UCON2:RECRUN2 is set when a falling edge of the signal at the receive port (P26) is detected.

3)   End of a receive operation
   • When the receive operation ends, UCON2:RECRUN2 is automatically cleared and UCON2:RECEND2 is set. The UART2 then waits for the start bit of the next received data.

4)   Receive interrupt processing
   • Read the received data (RBUF2).
   • Clear UCON2:RECEND2 and STPERR2 and exit the interrupt processing routine.
   * When changing the data length and transfer rate for the next receive operation, do so before the start bit (falling edge of the signal) is detected at the receive port (P26).

5)   Next receive data processing
   • Subsequently, repeat steps 2), 3), and 4) shown above.
   • To end the continuous receive operation, clear UCON2:STRDET2 during a receive operation, and this receive processing will be the last receive operation that UART2 executes.

3.16.5.2 Continuous 8-bit data transmit mode (first transmit data = 55H)
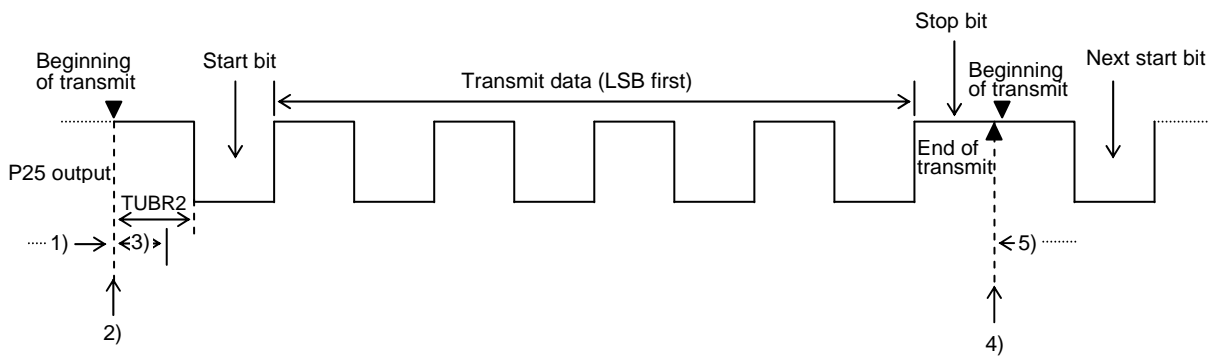
Figure 3.16.4 Example of Continuous 8-bit Data Transmit Mode Processing

1) Setting the clock
   - Set the transfer rate (UBR2).

   Setting up transmit data
   - Load the transmit data (TBUF2 =55H).

   Setting the data length, transmit port, and interrupts
   - Set up the transmit control register (UCON3 = 31H).
   * Set P25DR (P2DDR:bit 5) to 0 and P25 (P2:bit 5) to 0.

2) Starting a transmit operation
   - Set UCON3:TRUN2.

3) Transmit interrupt processing
   - Load the next transmit data (TBUF2 = xxH).
   - Clear UCON3:TEPTY2 and exit the interrupt processing routine.

4) End of a transmit operation
   - When the transmit operation ends, UCON3:TRUN2 is automatically cleared and automatically set in the same cycle (Tcyc) (continuous data transmit mode only; this processing takes 1 Tcyc of time). The UART2 then starts the transmission of the next transmit data.

5) Next transmit data processing
   - Subsequently, repeat steps 3) and 4) shown above.
   - To end the continuous transmit operation, clear UCON3:TRNSIE2 while not clearing UCON3:TEPTY2 and exit the interrupt in the step 3) processing, and the transmit operation that is being performed at that time will be the last transmit operation that UART2 executes.

3.16.5.3 Setting up the UART2 communications ports

1) Receive port (P26) settings

Register Data		Receive Port (P26) State	Internal Pull-up Resistor
P26	**P26DDR**		
0	0	Input	Off
1	0	Input	On

 * The UART2 can receive no data normally if P26DDR is set to 1.

2) Transmit port (P25) settings

Register Data				Transmit Port (P25) State	Internal Pull-up Resistor
P25	**P25DDR**	**TDDR2**	**TCMOS2**		
0	0	1	1	CMOS output	Off
0	0	1	0	N-channel open drain output	Off
1	0	1	0	N-channel open drain output	On

 * The UART2 transmits no data if P25DDR is set to 1.

3.16.6 UART2 HALT Mode Operation

3.16.6.1 Receive mode
1) UART2 receive mode operation is enabled in HALT mode. (If UCON2:STRDET2 is set to 1 when the microcontroller enters HALT mode, receive operation will be restarted if data such that UCON2:RECRUN2 is input at the end of a receive operation.)
2) HALT mode can be released using the UART2 receive interrupt.

3.16.6.2 Transmit mode
1) UART2 transmit mode operation is enabled in HALT mode. (If the continuous transmit mode is specified when the microcontroller enters HALT mode, UART2 will restart transmit processing after terminating a transmit operation. Since UCON3:TEPTY2 cannot be cleared in this case, UART2 stops processing after completing that transmit operation.)
2) HALT mode can be released using the UART2 transmit interrupt.

3.17 PWM0 and PWM1

3.17.1 Overview

This series of microcontrollers incorporates two 12-bit PWMs, named PWM0 and PWM1. Each PWM is made up of a PWM generator circuit that generates multifrequency 8-bit fundamental PWM waves and a 4-bit additional pulse generator.

PWM0 and PWM1 are provided with dedicated input/output pins PWM0 and PWM1, respectively.

3.17.2 Functions

1) PWM0: Fundamental PWM mode (register PWM0L=0)
    - Fundamental wave period = $\frac{(16\text{ to }256)}{3}$ Tcyc (programmable in $\frac{16}{3}$ Tcyc increments, common to PWM1)
    - High-level pulse width = 0 to (Fundamental wave period $-\frac{1}{3}$)Tcyc (programmable in $\frac{1}{3}$ Tcyc increments)

2) PWM0: Fundamental wave + Additional pulse PWM mode
    - Fundamental wave period = $\frac{(16\text{ to }256)}{3}$ Tcyc (programmable in $\frac{16}{3}$ Tcyc increments, common to PWM1)
    - Overall period = Fundamental wave period × 16
    - High-level pulse width = 0 to (Overall period $-\frac{1}{3}$)Tcyc (programmable in $\frac{1}{3}$ Tcyc increments)

3) PWM1: Fundamental wave PWM mode (register PWM1L=0)
    - Fundamental wave period = $\frac{(16\text{ to }256)}{3}$ Tcyc (programmable in $\frac{16}{3}$ Tcyc increments, common to PWM0)
    - High-level pulse width = 0 to (Fundamental wave period $-\frac{1}{3}$)Tcyc (programmable in $\frac{1}{3}$ Tcyc increments)

4) PWM1: Fundamental wave + Additional pulse PWM mode
    - Fundamental wave period = $\frac{(16\text{ to }256)}{3}$ Tcyc (programmable in $\frac{16}{3}$ Tcyc increments, common to PWM0)
    - Overall period = Fundamental wave period × 16
    - High-level pulse width = 0 to (Overall period $-\frac{1}{3}$)Tcyc (programmable in $\frac{1}{3}$ Tcyc increments)

5) Interrupt generation
    - Interrupt requests are generated at the intervals equal to the overall PWM period if the interrupt request enable bit is set.

6) Multiplexed input pin function
    - The PWM0 and PWM1 pins can also serve as input port pins.

7) To control PWM0 and PWM1, it is necessary to manipulate the following special function registers.
   ・ PWM0L, PWM0H, PWM1L, PWM1H, PWM0C, PWM01P

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE20	0000 HHHH	R/W	PWM0L	PWM0L3	PWM0L2	PWM0L1	PWM0L0	-	-	-	-
FE21	0000 0000	R/W	PWM0H	PWM0H7	PWM0H6	PWM0H5	PWM0H4	PWM0H3	PWM0H2	PWM0H1	PWM0H0
FE22	0000 HHHH	R/W	PWM1L	PWM1L3	PWM1L2	PWM1L1	PWM1L0	-	-	-	-
FE23	0000 0000	R/W	PWM1H	PWM1H7	PWM1H6	PWM1H5	PWM1H4	PWM1H3	PWM1H2	PWM1H1	PWM1H0
FE24	0000 0000	R/W	PWM0C	PWM0C7	PWM0C6	PWM0C5	PWM0C4	ENPWM1	ENPWM0	PWM0OV	PWM0IE
FE25	HHHH HHXX	R	PWM01P	-	-	-	-	-	-	PWM1IN	PWM0IN

3.17.3 Circuit Configuration

3.17.3.1 PWM0/PWM1 control register (PWM0C)   (8-bit register)

1) This register controls the operation and interrupts of PWM0 and PWM1.

3.17.3.2 PWM0 compare register L (PWM0L)   (4-bit register)

1) This register controls the additional pulses of PWM0.
2) PWM0L is assigned bits 7 to 4 and all of its lower-order 4 bits are apparently set to 1 when read.
3) When the PWM0 control bit (PWM0C: FE24, bit 2) is set to 0, the output of PWM0 (ternary) can be controlled using bits 7 to 4 of PWM0L.

3.17.3.3 PWM0 compare register H (PWM0H)   (8-bit register)

1) This register controls the fundamental wave pulse width of PWM0.
2) When bits 7 to 4 of PWM0L are all fixed at 0, PWM0 can serve as period-programmable 8-bit PWM that is controlled by PWM0H.

3.17.3.4 PWM1 compare register L (PWM1L)   (4-bit register)

1) This register controls the additional pulses of PWM1.
2) PWM1L is assigned bits 7 to 4 and all of its lower-order 4 bits are apparently set to 1 when read.
3) When the PWM1 control bit (PWM0C: FE24, bit 3) is set to 0, the output of PWM1 (ternary) can be controlled using bits 7 to 4 of PWM1L.

3.17.3.5 PWM1 compare register H (PWM1H)   (8-bit register)

1) This register controls the fundamental wave pulse width of PWM1.
2) When bits 7 to 4 of PWM1L are all fixed at 0, PWM1 can serve as period-programmable 8-bit PWM that is controlled by PWM1H.

3.17.3.6 PWM01 port input register (PWM01P)   (2-bit register)

1) PWM0 data can be read into this register as bit 0.
2) PWM1 data can be read into this register as bit 1.

3.17.4　Related Registers

3.17.4.1　PWM0/PWM1 control register (PWM0C)　(8-bit register)

1)　This register controls the operation and interrupts of PWM0 and PWM1.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE24	0000 0000	R/W	PWM0C	PWM0C7	PWM0C6	PWM0C5	PWM0C4	ENPWM1	ENPWM0	PWM0OV	PWM0IE

PWM0C7 to PWM0C4 (bits 7 to 4): PWM0/PWM1 period control

- Fundamental wave period 　= (Value represented by (PWM0C7 to PWM0C4) + 1) $\times \frac{16}{3}$ Tcyc
- Overall period 　　　　　= Fundamental wave period $\times$ 16

ENPWM1 (bit 3): PWM1 operation control

- When this bit is set to 1, PWM1 is active.
- When this bit is set to 0, the PWM1 output (ternary) can be controlled using bits 7 to 4 of PWM1L.

ENPWM0 (bit 2): PWM0 operation control

- When this bit is set to 1, PWM0 is active.
- When this bit is set to 0, the PWM0 output (ternary) can be controlled using bits 7 to 4 of PWM0L.

PWM0OV (bit 1): PWM0/PWM1 overflow flag

- This bit is set at the interval equal to the overall period of PWM.
- This flag must be cleared with an instruction.

PWM0IE (bit 0): PWM0/PWM1 interrupt request enable control

　　An interrupt to vector address 004BH is generated when this bit and PWM0OV are set to 1.

3.17.4.2　PWM0 compare register L (PWM0L)　(4-bit register)

1)　This register controls the additional pulses of PWM0.

2)　PWM0L is assigned bits 7 to 4 and all of its lower-order 4 bits are apparently set to 1 when read.

3)　When the PWM0 control bit (PWM0C: FE24, bit 2) is set to 0, the output of PWM0 (ternary) can be controlled using bits 7 to 4 of PWM0L.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE20	0000 HHHH	R/W	PWM0L	PWM0L3	PWM0L2	PWM0L1	PWM0L0	-	-	-	-

PWM0 Output	ENPWM0 FE24, bit 2	PWM0L3 FE20, bit 7	PWM0L2 FE20, bit 6	PWM0L1, 0 FE20, bits 5,4
Hi-Z	0	—	0	—
Low	0	0	1	0, 0
High	0	1	1	0, 0

3.17.4.3 PWM0 compare register H (PWM0H)   (8-bit register)

1) This register controls the fundamental wave pulse width of PWM0.

Fundamental wave pulse width = (Value represented by PWM0H7 to PWM0H0) $\times \frac{1}{3}$ Tcyc

2) When bits 7 to 4 of PWM0L are all fixed at 0, PWM0 can serve as period-programmable 8-bit PWM that is controlled by PWM0H.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE21	0000 0000	R/W	PWM0H	PWM0H7	PWM0H6	PWM0H5	PWM0H4	PWM0H3	PWM0H2	PWM0H1	PWM0H0

3.17.4.4 PWM1 compare register L (PWM1L)   (4-bit register)

1) This register controls the additional pulses of PWM1.

2) PWM1L is assigned bits 7 to 4 and all of its lower-order 4 bits are apparently set to 1 when read.

3) When the PWM1 control bit (PWM0C: FE24, bit 3) is set to 0, the output of PWM1 (ternary) can be controlled using bits 7 to 4 of PWM1L.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE22	0000 HHHH	R/W	PWM1L	PWM1L3	PWM1L2	PWM1L1	PWM1L0	-	-	-	-

PWM1 Output	ENPWM1 FE24, bit 3	PWM1L3 FE22, bit 7	PWM1L2 FE22, bit 6	PWM1L1, 0 FE22, bits 5, 4
Hi-Z	0	—	0	—
Low	0	0	1	0, 0
High	0	1	1	0, 0

3.17.4.5 PWM1 compare register H (PWM1H)   (8-bit register)

1) This register controls the fundamental wave pulse width of PWM1.

Fundamental wave pulse width =(Value represented by PWM1H7 to PWM1H0) $\times \frac{1}{3}$ Tcyc

2) When bits 7 to 4 of PWM1L are all fixed at 0, PWM1 can serve as period-programmable 8-bit PWM that is controlled by PWM1H.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE23	0000 0000	R/W	PWM1H	PWM1H7	PWM1H6	PWM1H5	PWM1H4	PWM1H3	PWM1H2	PWM1H1	PWM1H0
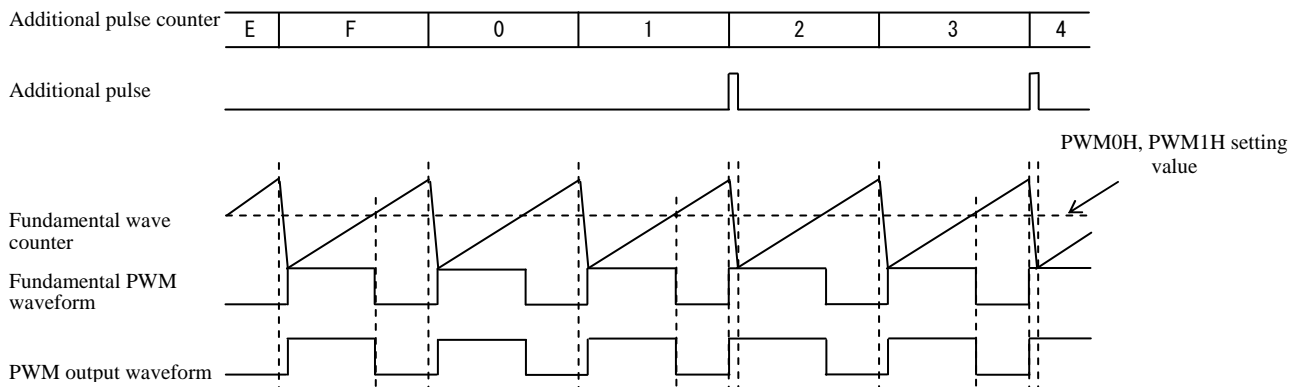
● The 12-bit PWM has the following waveform structure:

  • The overall period consists of 16 fundamental wave periods.
  • A fundamental wave period is represented by an 8-bit PWM. (PWM compare register H) (PWMH)
  • 4 bits are used to designate the fundamental wave period to which additional pulses are to be added. (PWM compare match register L) (PWML)

    12-bit register structure   →   (PWMH), (PWML)=XXXX XXXX, XXXX (12 bits)

● How pulses are added to the fundamental wave periods (Example 1)

  • PWM compare register H (PWMH)   =   00   [H]
  • PWM compare register L (PWML)   =   0 to F   [H]
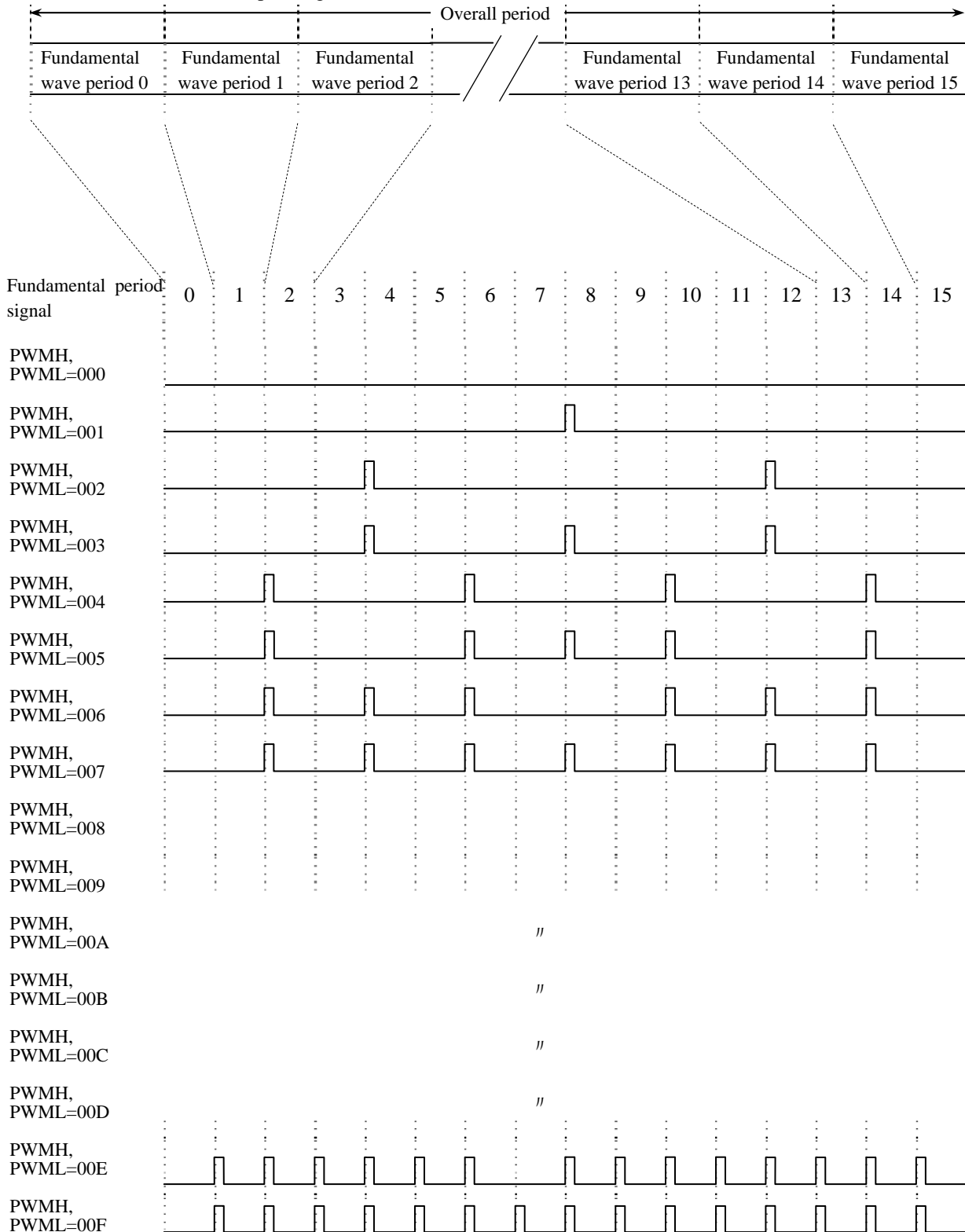
PWM

● How pulses are added to fundamental wave periods (Example 2)

- PWM compare register H (PWMH) = 01 [H]
- PWM compare register L (PWML) = 0 to F [H]

Overall period

| Fundamental wave period 0 | Fundamental wave period 1 | Fundamental wave period 2 | // | Fundamental wave period 13 | Fundamental wave period 14 | Fundamental wave period 15 |

Fundamental period signal   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

PWMH, PWML=010

PWMH, PWML=011

PWMH, PWML=012

PWMH, PWML=013

PWMH, PWML=014

PWMH, PWML=015

PWMH, PWML=016

PWMH, PWML=017

PWMH, PWML=018

PWMH, PWML=019

PWMH, PWML=01A   //

PWMH, PWML=01B   //

PWMH, PWML=01C   //

PWMH, PWML=01D   //

PWMH, PWML=01E

PWMH, PWML=01F

● The fundamental wave period is variable within the range of $\frac{(16\ to\ 256)}{3}$ Tcyc.

Fundamental wave period = (Value represented by PWM0C7 to PWM0C4 + 1) × $\frac{16}{3}$ Tcyc

- The overall period can be changed by changing the fundamental wave period.
- The overall period is made up of 16 fundamental wave periods.

3-120

Examples:

● Wave comparison when the 12-bit PWM contains 237[H].

12-bit register configuration → (PWMH), (PWML) = 237[H]

1. Pulse added system (this series)
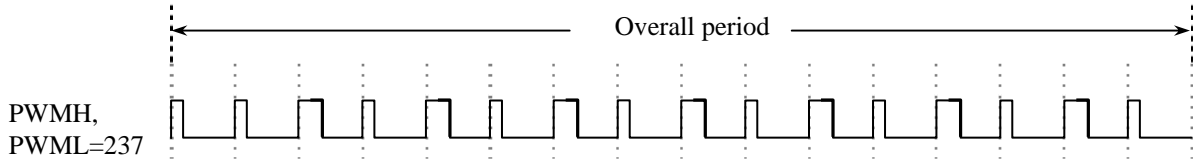
2. Ordinary system

Since the ripple component of the integral output in this system is greater than that of the pulse added system as seen from the figure below, the pulse added system is considered better for motor-controlling uses.

Ripple
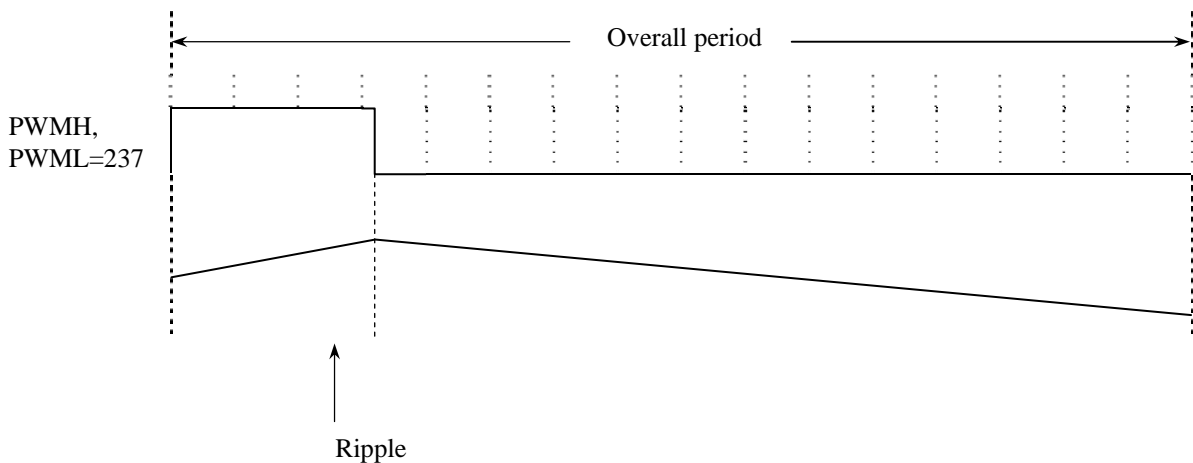
3.17.4.6 PWM01 port input register (PWM01P) (2-bit register)

1) PWM0 data can be read into this register as bit 0.

2) PWM1 data can be read into this register as bit 1.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE25	HHHH HHXX	R	PWM01P	-	-	-	-	-	-	PWM1IN	PWM0IN

(Bits 7 to 2): These bits do not exist. They are always read as 1.

PWM1IN (bit 1): PWM1 pin data (read only)

PWM0IN (bit 0): PWM0 pin data (read only)

3.18 AD Converter (ADC12)

3.18.1 Overview

This series of microcontrollers incorporates a 12-bit resolution AD converter that has the features listed below. It allows the microcontroller to take in analog signals easily.

1) 12-bit resolution
2) Successive approximation
3) AD conversion mode select (resolution switching)
4) 12-channel analog input
5) Conversion time select
6) Automatic reference voltage generation control

3.18.2 Functions

1) Successive approximation

   - The ADC has a resolution of 12 bits.
   - It requires some conversion time after starting conversion processing.
   - The conversion results are placed in the AD conversion result registers (ADRLC, ADRHC).

2) AD conversion select (resolution switching)

   The AD converter supports two AD conversion modes: 12- and 8-bit conversion modes so that the appropriate conversion resolution can be selected according to the operating conditions of the application. Conversion mode switching is accomplished through the AD mode register (ADMRC).

3) 12-channel analog input

   The signal to be converted is selected using the AD control register (ADCRC) out of 12 types of analog signals that are supplied from P0 and P70, P71, XT1, and XT2 pins.

4) Conversion time select

   The AD conversion time can be set to 1/1 to 1/128 (frequency division ratio). The AD mode register (ADMRC) and AD conversion result low byte register (ADRLC) are used to select the conversion time for appropriate AD conversion.

5) Automatic reference voltage generation control

   The ADC incorporates a reference voltage generator that automatically generates the reference voltage when an AD conversion starts and stops the generation when the conversion ends. Accordingly, set/reset control of reference voltage generation is not necessary. Also, there is no need to supply reference voltage externally.

6) It is necessary to manipulate the following special control registers to control the AD converter.

• ADCRC, ADMRC, ADRLC, ADRHC

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE58	0000 0000	R/W	ADCRC	AD CHSEL3	AD CHSEL2	AD CHSEL1	AD CHSEL0	ADCR3	AD START	AD ENDF	ADIE
FE59	0000 0000	R/W	ADMRC	ADMD4	ADMD3	ADMD2	ADMD1	ADMD0	ADMR2	ADTM1	ADTM0
FE5A	0000 0000	R/W	ADRLC	DATAL3	DATAL2	DATAL1	DATAL0	ADRL3	ADRL2	ADRL1	ADTM2
FE5B	0000 0000	R/W	ADRHC	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

3.18.3 Circuit Configuration

3.18.3.1 AD conversion control circuit

1) The AD conversion control circuit runs in two modes: 12- and 8-bit AD conversion modes.

3.18.3.2 Comparator circuit

1) The comparator circuit consists of a comparator that compares the analog input with the reference voltage and a circuit that controls the reference voltage generator circuit and the conversion results. The end of conversion bit (ADENDF) of the AD control register (ADCRC) is set when an analog input channel is selected and the AD conversion terminates in the conversion time designated by the conversion time control register. The conversion results are placed in the AD conversion result registers (ADRHC, ADRLC).

3.18.3.3 Multiplexer 1 (MPX1)

1) Multiplexer 1 is used to select the analog signal to be subject to AD conversion from 12 channels of analog signals.

3.18.3.4 Automatic reference voltage generator circuit

1) The reference voltage generator circuit consists of a network of ladder resistors and a multiplexer (MPX2) and generates the reference voltage that is supplied to the comparator circuit. Generation of the reference voltage is automatically started when an AD conversion starts and stopped when the conversion ends. The reference voltage output ranges from VDD to VSS.

3.18.4 Related Registers

3.18.4.1 AD control register (ADCRC)

1) This register is an 8-bit register that controls the operation of the AD converter.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE58	0000 0000	R/W	ADCRC	AD CHSEL3	AD CHSEL2	AD CHSEL1	AD CHSEL0	ADCR3	AD START	AD ENDF	ADIE

ADCHSEL3 (bit 7):
ADCHSEL2 (bit 6):   **AD conversion input signal select**
ADCHSEL1 (bit 5):
ADCHSEL0 (bit 4):

These 4 bits are used to select the signal to be subject to AD conversion.

ADC12

AD CHSEL3	AD CHSEL2	AD CHSEL1	AD CHSEL0	Signal Input Pin
0	0	0	0	P00/AN0
0	0	0	1	P01/AN1
0	0	1	0	P02/AN2
0	0	1	1	P03/AN3
0	1	0	0	P04/AN4
0	1	0	1	P05/AN5
0	1	1	0	P06/AN6
0	1	1	1	P07/AN7
1	0	0	0	P70/AN8
1	0	0	1	P71/AN 9
1	0	1	0	XT1/AN10
1	0	1	1	XT2/AN11

ADCRC3 (bit 3): Fixed bit

This bit must always be set to 0.

ADSTART (bit 2): AD converter operation control

This bit starts (1) or stops (0) AD conversion processing. Setting this bit to 1 starts AD conversion. The bit is reset automatically when the AD conversion ends. The time specified by the conversion time control register is required to complete the conversion. The conversion time is defined using three bits, i.e., the ADTM2 bit (bit 0) of the AD conversion result register low byte (ADRLC) and the ADTM1 (bit 1) and ADTM0 (bit 0) of the AD mode register (ADMRC).

Setting this bit to 0 stops the AD conversion. No correct conversion results can be obtained if this bit is cleared when AD conversion is in progress.

Never clear this bit or place the microcontroller in HALT or HOLD mode while the AD conversion processing is in progress.

ADENDF (bit 1): End of AD conversion flag

This bit identifies the end of AD conversion. It is set (1) when AD conversion is finished. Then, an interrupt request to vector address 0043H is generated if ADIE is set to 1. If ADENDF is set to 0, it indicates that no AD conversion operation is in progress.

This flag must be cleared with an instruction.

ADIE (bit 0): AD conversion interrupt request enable control

An interrupt request to vector address 0043H is generated when this bit and ADENDF are set to 1.

Notes:

- *It is inhibited to set ADCHSEL3 to ADCHSEL0 to any value between "1100" to "1111."*

- *Do not place the microcontroller in HALT or HOLD mode with ADSTART set to 1. Make sure that ADSTART is set to 0 before putting the microcontroller in HALT or HOLD mode.*

3.18.4.2 AD mode register (ADMRC)

1) This register is an 8-bit register that controls AD converter operation mode.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE59	0000 0000	R/W	ADMRC	ADMD4	ADMD3	ADMD2	ADMD1	ADMD0	ADMR2	ADTM1	ADTM0

ADMD4 (bit 7): Fixed bit

This bit must always be set to 0.

ADMD3 (bit 6): AD conversion mode control (resolution select)

This bit selects the AD converter resolution between 12-bit AD conversion mode (0) and 8-bit AD conversion mode (1).

If this bit is set to 1, the AD converter serves as an 8-bit AD converter. The conversion results are placed only in the AD conversion result register high byte (ADRHC); the contents of the AD conversion result register low byte (ADRLC) remain unchanged.

If this bit is set to 0, the AD converter serves as a 12-bit AD converter. The conversion results are placed in the AD conversion result register high byte (ADRHC) and the higher-order 4 bits of the AD conversion result register low byte (ADRLC).

ADMD2 (bit 5): Fixed bit

This bit must always be set to 0.

ADMD1 (bit 4): Fixed bit

This bit must always be set to 0.

ADMD0 (bit 3): Fixed bit

This bit must always be set to 0.

ADMR2 (bit 2): Fixed bit

This bit must always be set to 0.

ADTM1 (bit 1):
ADTM0 (bit 0): } **AD conversion time control**

These bits and ADTM2 bit (bit 0) of the AD conversion result register low byte (ADRLC) define the conversion time.

ADRLC Register	ADMRC Register		Frequency Division Ratio
ADTM2	ADTM1	ADTM0	
0	0	0	1/1
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	1/32
1	1	0	1/64
1	1	1	1/128

ADC12

Conversion time calculation formulas

- 12-bit AD conversion mode: Conversion time = $((52/(\text{AD division ratio})) + 2) \times 1/3 \times \text{Tcyc}$

- 8-bit AD conversion mode: Conversion time = $((32/(\text{AD division ratio})) + 2) \times 1/3 \times \text{Tcyc}$

Notes:

- *The conversion time is doubled in the following cases:*

    1) *The AD conversion is carried out in the 12-bit AD conversion mode for the first time after a system reset.*

    2) *The AD conversion is carried out for the first time after the AD conversion mode is switched from 8-bit to 12-bit AD conversion mode.*

- *The conversion time determined by the above formula is taken in the second and subsequent conversions or in the AD conversions that are carried out in the 8-bit AD conversion mode.*

3.18.4.3 AD conversion result low byte register (ADRLC)

1) This register is used to hold the lower-order 4 bits of the results of an AD conversion carried out in the 12-bit AD conversion mode and to control the conversion time.

2) Since the data in this register is not established during an AD conversion, the conversion results must be read out only after the AD conversion is completed.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5A	0000 0000	R/W	ADRLC	DATAL3	DATAL2	DATAL1	DATAL0	ADRL3	ADRL2	ADRL1	ADTM2

DATAL3 (bit 7):
DATAL2 (bit 6):
DATAL1 (bit 5):
DATAL0 (bit 4):

AD conversion result lower-order 4 bits

ADRL3 (bit 3): Fixed bit
>   This bit must always be set to 0.

ADRL2 (bit 2): Fixed bit
>   This bit must always be set to 0.

ADRL1 (bit 1): Fixed bit
>   This bit must always be set to 0.

ADTM2 (bit 0): AD conversion time control
>   This bit and AD mode register (ADMRC) bits ADTM1 (bit 1) and ADTM0 (bit 0) are used to control the conversion time. See the subsection on the AD mode register for the procedure to set the conversion time.

Note:

- *The conversion result data contains some errors (quantization error + combination error). Be sure to use only valid conversion results while referring to the latest "SANYO Semiconductors Data Sheet."*

3.18.4.4 AD conversion result high byte register (ADRHC)

1) This register is used to hold the higher-order 8 bits of the results of an AD conversion that is carried out in the 12-bit AD conversion mode. The register stores the whole 8 bits of an AD conversion that is carried out in the 8-bit AD conversion mode.

2) Since the data in this register is not established during an AD conversion, the conversion results must be read out only after the AD conversion is completed.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5B	0000 0000	R/W	ADRHC	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

3.18.5   AD Conversion Example

3.18.5.1   12-bit AD conversion mode

1)   Setting up the 12-bit AD conversion mode

• Set the ADMD3 bit (bit 6) of the AD mode register (ADMRC) to 0.

2)   Setting up the conversion time

• To set the conversion time to 1/32, set ADTM2 (bit 0) of the AD conversion result register low byte (ADRLC) to 1, ADTM1 (bit 1) of the AD mode register (ADMRC) to 0, and ADTM0 (bit 0) of the AD mode register (ADMRC) to 1.

3)   Setting up the input channel

• When using AD channel input AN5, set AD control register ADCRC: ADCHSEL3 (bit 7) to 0, ADCHSEL2 (bit 6) to 1, ADCHSEL1 (bit 5) to 0, and ADCHSEL0 (bit 4) to 1.

4)   Starting AD conversion

• Set ADSTART (bit 2) of the AD control register (ADCRC) to 1.

• The conversion time is doubled after a system reset and when the AD conversion is carried out for the first time after the AD conversion mode is switched from 8-bit to 12-bit AD conversion mode. The conversion time determined by the formula is taken in the second and subsequent conversions.

5)   Testing the end of AD conversion flag

• Monitor ADENDF (bit 1) of the AD control register (ADCRC) until it is set to 1.

• Clear the end of conversion flag (ADENDF) to 0 after confirming that the ADENDF flag (bit 1) is set to 1.

6)   Reading in the AD conversion results

• Read the AD conversion result high byte register (ADRHC) and AD conversion result low byte register (ADRLC). Since the conversion result data contains some errors (quantization error + combination error), use only the valid part of the conversion data selected according to the specifications given in the latest "SANYO Semiconductors Data Sheet."

• Pass the above read data to the application software processing.

• Return to step 4) to repeat the conversion processing.

3.18.6   Hints on the Use of the ADC

1) The conversion time that the user can select varies depending on the frequency of the cycle clock. When preparing a program, refer to the latest "SANYO Semiconductors Data Sheet" to select an appropriate conversion time.

2) Setting ADSTART to 0 while conversion is in progress will stop the conversion function.

3) Do not place the microcontroller in HALT or HOLD mode while AD conversion processing is in progress. Make sure that ADSTART is set to 0 before putting the microcontroller in HALT or HOLD mode.

4) ADSTART is automatically reset and the AD converter stops operation if a reset is triggered while AD conversion processing is in progress.

5) When conversion is finished, the end of AD conversion flag (ADENDF) is set and, at the same time, the AD conversion operation control bit (ADSTART) is reset. The end of conversion condition can be identified by monitoring ADENDF. Setting ADIE causes an interrupt request to vector address 0043H to be generated at the end of conversion.

6) The conversion time is doubled in the following cases:
   • The AD conversion is carried out in the 12-bit AD conversion mode for the first time after a system reset.
   • The AD conversion is carried out for the first time after the AD conversion mode is switched from 8-bit to 12-bit AD conversion mode.
   • The conversion time determined using the "conversion time calculation formula" is adopted in the second and subsequent conversions or in the AD conversions that are carried out in the 8-bit AD conversion mode.

7) The conversion result data contains some errors (quantization error + combination error). Be sure to use only valid conversion results while referring to the latest "SANYO Semiconductors Data Sheet."

8) Make sure that only input voltages that fall within the specified range are supplied to pins P00/AN0 to P07/AN7, P70/AN8, P71/AN9, XT1/AN10, and XT2/AN11. Application of a voltage greater than VDD or lower than VSS to an input pin may exert adverse influences on the converted value of the channel in question or other channels.

9) Take the following preventive actions as countermeasures to keep the reduction in conversion accuracy due to noise interferences as low as possible:
   • Be sure to add external bypass capacitors several μF and thousands pF near the VDD1 and VSS1 pins (as close as possible, desirably 5 mm or less).
   • Add external low-pass filters (RC) or capacitors, most suitable for noise reduction, immediately close to the analog input pins. To avert the adverse coupling influences, use a ground that is free of noise interferences as the ground for the capacitors (rough standard values are: R = less than 5 kΩ, C=1000 pF to 0.1μF).
   • Do not lay analog signal lines close to, in parallel with, or in a crossed arrangement with digital pulse signal lines or signal lines in which large current changes can occur. Shield both ends of analog signal lines with noise-free ground shields.
   • Make sure that no digital pulses are applied to or generated out of pins adjacent to the analog input pin that is being subject to conversion.

• Correct conversion results may not be obtained because of noise interferences if the state of port outputs is changing. To minimize the adverse influences of noise interferences, it is necessary to keep the line resistance across the power supply and the VDD pins of the microcontroller at minimum. This should be kept in mind when designing an application circuit.

• Adjust the amplitudes of the voltage at the oscillator pin and the I/O voltages at the other pins so that they fall within the voltage range between VDD and VSS.

10) To obtain valid conversion data, perform conversion operations on the input several times, discard the maximum and minimum values of the conversion results, and take an average of the remaining data.

3.19 USB Interface

3.19.1 Overview

This series of microcontrollers is provided with a USB (Universal Serial Bus) function control circuit that has the following features:

1) Compatible with the USB Specification Version 2.0.
2) Compatible with Full Speed (12 Mbps) specifications.
3) Supports the control transfer, bulk transfer, interrupt transfer, and isochronous transfer modes.

3.19.2 Functions

1) USB function control
   - Can define a maximum of 5 endpoints (including the control endpoint EP0).
   - The endpoint buffer for data transmission and reception (64 bytes maximum) is mapped into RAM.
   - The USB data sampling clock (48 MHz) can be derived from a clock that is generated by the internal PLL circuit.

2) Interrupt generation

   An interrupt request is generated at the end of a USB transaction or on detection of a bus reset signal, a suspend condition, an SOF packet, or resume signal if the corresponding interrupt request enable bit is set.

3) To control the USB interface, it is necessary to manipulate the following special function registers.
   - USBDIV, PLLCNT
   - USCTRL, USPORT, USBINT, EP0INT, EP1INT, EP2INT,
     EP3INT, EP4INT, FRAMEL, FRAMEH, USBADR, EPINFO,
     EP0STA, EP0MP, EP0RX, EP0TX, EP1STA, EP2STA, EP3STA,
     EP4STA, EP1CNT,EP1RX, EP2CNT, EP2RX, EP3CNT, EP3RX,
     EP4CNT, EP4RX, TESTR0, TESTR1, EPADOF
   - P3, P3DDR

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE04	0000 0000	R/W	USBDIV	CF48ON	DVCKON	DVCKDR	P73NDL	CF12OFF	UDVSEL2	UDVSEL1	UDVSEL0
FE0D	0000 0000	R/W	PLLCNT	SELREF2	SELREF1	SELREF0	PLLTEST	VCOSTP	CMPSTP	PLLCNT1	PLLCNT0
FE80	0000 0000	R/W	USCTRL	USBON	USBRUN	VD3OEN	VD3KIL	IDLFG	IDLEN	DPIEZ	DMIEZ
FE81	0000 0000	R/W	USPORT	DDRSOF	VD3OE2	USBSIO	SUSPND	DDRDP	DDRDM	PORTDP	PORTDM
FE82	0000 0000	R/W	USBINT	BRSFG	BRSEN	BACFG	BACEN	SOFFG	SOFEN	USBINT1	ENPEN
FE83	0000 0000	R/W	EP0INT	AK0FG	AK0EN	NK0FG	NK0EN	ER0FG	ER0EN	ST0FG	ST0EN
FE84	0000 0000	R/W	EP1INT	AK1FG	AK1EN	NK1FG	NK1EN	ER1FG	ER1EN	ST1FG	ST1EN
FE85	0000 0000	R/W	EP2INT	AK2FG	AK2EN	NK2FG	NK2EN	ER2FG	ER2EN	ST2FG	ST2EN
FE86	0000 0000	R/W	EP3INT	AK3FG	AK3EN	NK3FG	NK3EN	ER3FG	ER3EN	ST3FG	ST3EN
FE87	0000 0000	R/W	EP4INT	AK4FG	AK4EN	NK4FG	NK4EN	ER4FG	ER4EN	ST4FG	ST4EN
FE8A	0000 0000	R	FRAMEL	FRM07	FRM06	FRM05	FRM04	FRM03	FRM02	FRM01	FRM00

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE8B	HHHH H000	R	FRAMEH	-	-	-	-	-	FRM10	FRM09	FRM08
FE8C	0000 0000	R/W	USBADR	ADREN	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
FE8D	0000 0000	R/W	EPINFO	EPNO3	EPNO2	EPNO1	EPNO0	TKN1	TKN0	CTKN1	CTKN0
FE8E	0000 0000	R/W	EP0STA	E0EN	E0TGL	E0OVR	E0STL	E0ACK	E0CSU	E0CST	E0CRW
FE8F	H000 0000	R/W	EP0MP	-	E0MP6	E0MP5	E0MP4	E0MP3	E0MP2	E0MP1	E0MP0
FE90	H000 0000	R/W	EP0RX	-	E0RX6	E0RX5	E0RX4	E0RX3	E0RX2	E0RX1	E0RX0
FE91	H000 0000	R/W	EP0TX	-	E0TX6	E0TX5	E0TX4	E0TX3	E0TX2	E0TX1	E0TX0
FE92	0000 0000	R/W	EP1STA	E1EN	E1TGL	E1OVR	E1STL	E1ACK	E1DIR	E1ISO	E1BNK
FE93	0000 0000	R/W	EP2STA	E2EN	E2TGL	E2OVR	E2STL	E2ACK	E2DIR	E2ISO	E2BNK
FE94	0000 0000	R/W	EP3STA	E3EN	E3TGL	E3OVR	E3STL	E3ACK	E3DIR	E3ISO	E3BNK
FE95	0000 0000	R/W	EP4STA	E4EN	E4TGL	E4OVR	E4STL	E4ACK	E4DIR	E4ISO	E4BNK
FE98	H000 0000	R/W	EP1CNT	-	E1CN6	E1CN5	E1CN4	E1CN3	E1CN2	E1CN1	E1CN0
FE99	H000 0000	R/W	EP1RX	-	E1RX6	E1RX5	E1RX4	E1RX3	E1RX2	E1RX1	E1RX0
FE9A	H000 0000	R/W	EP2CNT	-	E2CN6	E2CN5	E2CN4	E2CN3	E2CN2	E2CN1	E2CN0
FE9B	H000 0000	R/W	EP2RX	-	E2RX6	E2RX5	E2RX4	E2RX3	E2RX2	E2RX1	E2RX0
FE9C	H000 0000	R/W	EP3CNT	-	E3CN6	E3CN5	E3CN4	E3CN3	E3CN2	E3CN1	E3CN0
FE9D	H000 0000	R/W	EP3RX	-	E3RX6	E3RX5	E3RX4	E3RX3	E3RX2	E3RX1	E3RX0
FE9E	H000 0000	R/W	EP4CNT	-	E4CN6	E4CN5	E4CN4	E4CN3	E4CN2	E4CN1	E4CN0
FE9F	H000 0000	R/W	EP4RX	-	E4RX6	E4RX5	E4RX4	E4RX3	E4RX2	E4RX1	E4RX0
FEAC	0000 0000	R/W	TESTR0	DPLTST	CMPTST	CMPKIL	TTXCLK	TTXREQ	TADR2	TADR1	P71NDL
FEAD	0000 0000	R/W	TESTR1	TDAT7	TDAT6	TDAT5	TDAT4	TDAT3	TDAT2	TDAT1	TDAT0
FEAE	0000 0000	R/W	EPADOF	E4OF1	E4OF0	E3OF1	E3OF0	E2OF1	E2OF0	E1OF1	E1OF0

3.19.3   Circuit Configuration

The USB interface control circuit is made up of the functional blocks shown below.
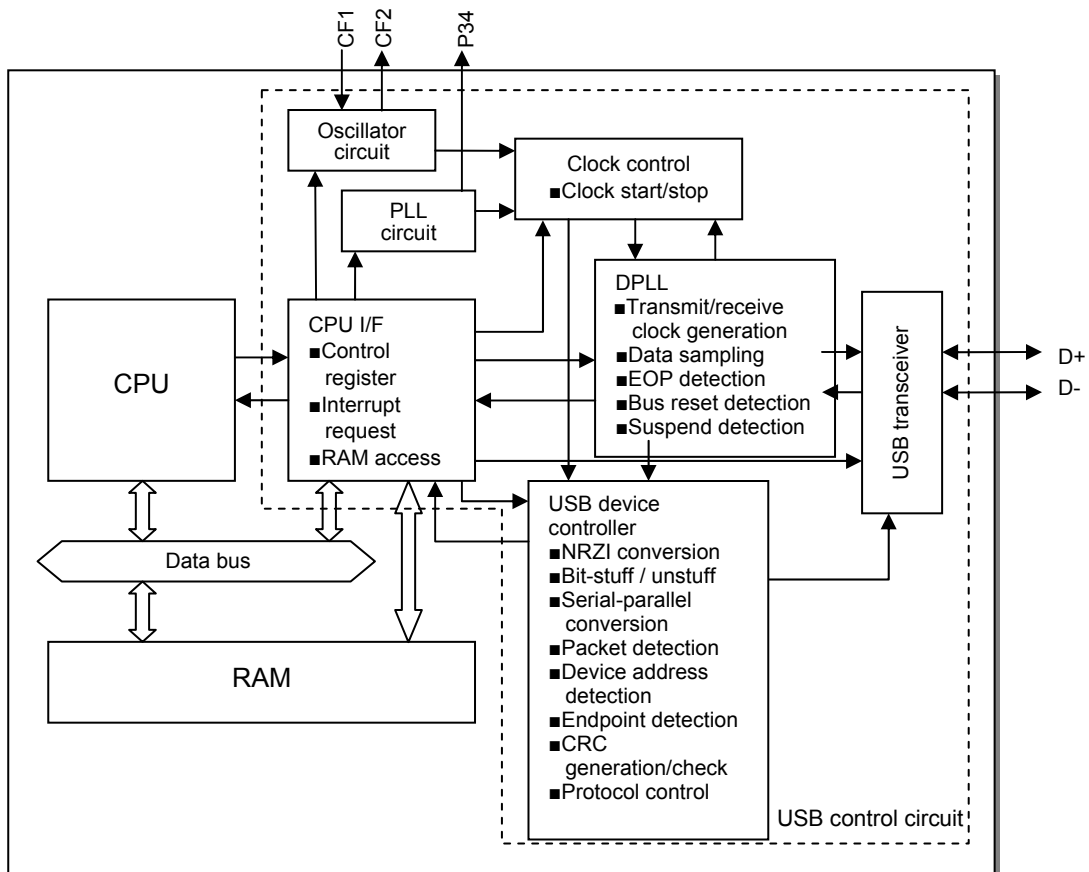
Figure 3.19.1   USB Interface Control Circuit Block Diagram

3-131

3.19.3.1   USB interface PLL circuit

The internal USB interface PLL circuit generates the USB data sampling clock (48 MHz).

3.19.3.2   Clock control circuit

The clock control circuit turns on and off the clock to the DPLL and USB device controller.

3.19.3.3   DPLL circuit

1) Generates the transmit/receive clock (12 MHz).
2) Samples the received data.
3) Detects EOP.
4) Detects the bus reset signal.
5) Detects the suspend condition

3.19.3.4   USB device controller circuit

1) Performs NRZI encoding/decoding.
2) Performs bit stuffing/unstuffing.
3) Performs serial-to-parallel conversion.
4) Detects packets.
5) Detects the device address and endpoints.
6) Performs CRC generation and check.
7) Exercises protocol control.

3.19.3.5   CPU interface circuit

1) Contains registers for controlling the operation of the USB interface circuit.
2) Generates interrupt requests to the CPU.
3) Has a data transmit/receive buffer.
4) Transfers transmit data from RAM (endpoint buffer) to the transmit/receive buffer.
5) Transfers receive data from the transmit/receive buffer to RAM (end point buffer).

3.19.3.6   Endpoint configuration

1) The USB interface circuit allows a maximum of 5 endpoints to be configured.
2) The table below lists the transfer types that each endpoint supports, buffer sizes, and the RAM addresses to which the data buffers are mapped.

Table 3.19.1   Endpoint Configuration

	Transfer Type	Settings		Maximum Size(bytes)	RAM Address
EP0	Control	Receive		64	0200H to 023FH
		Transmit			0240H to 027FH
EP1	Interrupt, bulk, isochronous	Bank 0	Offset 0	64	0280H to 02BFH
			Offset 1		0290H to 02CFH
			Offset 2		02A0H to 02DFH
			Offset 3		02B0H to 02EFH
		Bank 1	Offset 0		02C0H to 02FFH
			Offset 1		02D0H to 030FH
			Offset 2		02E0H to 031FH
			Offset 3		02F0H to 032FH
EP2	Interrupt, bulk, isochronous	Bank 0	Offset 0	64	0300H to 033FH
			Offset 1		0310H to 034FH
			Offset 2		0320H to 035FH
			Offset 3		0330H to 036FH
		Bank 1	Offset 0		0340H to 037FH
			Offset 1		0350H to 038FH
			Offset 2		0360H to 039FH
			Offset 3		0370H to 03AFH
EP3	Interrupt, bulk, isochronous	Bank 0	Offset 0	64	0380H to 03BFH
			Offset 1		0390H to 03CFH
			Offset 2		03A0H to 03DFH
			Offset 3		03B0H to 03EFH
		Bank 1	Offset 0		03C0H to 03FFH
			Offset 1		03D0H to 040FH
			Offset 2		03E0H to 041FH
			Offset 3		03F0H to 042FH
EP4	Interrupt, bulk, isochronous	Bank 0	Offset 0	64	0380H to 03BFH
			Offset 1		0390H to 03CFH
			Offset 2		03A0H to 03DFH
			Offset 3		03B0H to 03EFH
		Bank 1	Offset 0		03C0H to 03FFH
			Offset 1		03D0H to 040FH
			Offset 2		03E0H to 041FH
			Offset 3		03F0H to 042FH

3.19.3.7   Related I/O pins
1)   The table below lists the I/O pins that are associated with the USB interface control circuit.

Table 3.19.2   USB Related I/O Pins

Pin Name	I/O	Description
D＋	I/O	USB serial data I/O pin
D－	I/O	USB serial data I/O pin
P34	I/O	Pin connected to the internal PLL filter circuit
P70	O	D＋ pull-up control pin
P27	O	D＋ pull-up control pin

2)   The USB port peripheral circuit is shown in the figure below.

3)   The pull-up resistor (1.5 kΩ) at the D+ pin must be connected to the P70 or P27 pin so that it can be turned on and off according to the presence or absence of Vbus. The on/off control of this register must be accomplished through bit 5 (VD3OEN) of the USCTRL register (FE80H) or bit 6 (VD3OE2) of the USPORT resister (FE81H).
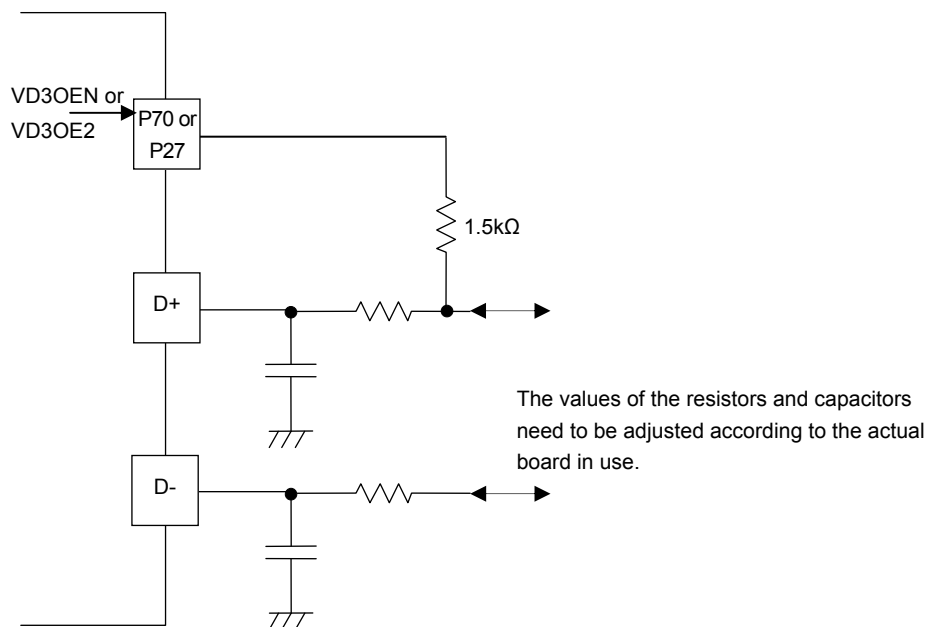
Figure 3.19.2   USB Port Peripheral Circuit

4)   To generate the USB 48 MHz clock using the internal PLL circuit, it is necessary to connect an external filter circuit to the P34/UFILT pin as shown below.
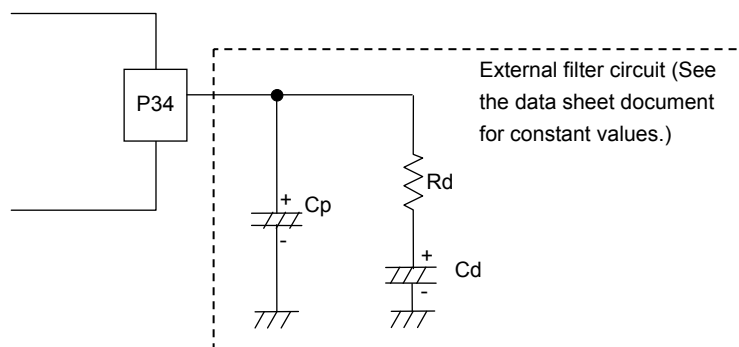
Figure 3.19.3   External Filter Circuit for the Internal PLL Circuit

3-134

3.19.4 Related Registers

3.19.4.1 USB frequency divided clock control register (USBDIV)

1) This register is used to select the frequency of the frequency divided clock that is derived from the USB 48 MHz clock.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE04	0000 0000	R/W	USBDIV	CF48ON	DVCKON	DVCKDR	P73NDL	CF12OFF	UDVSEL2	UDVSEL1	UDVSEL0

CF48ON (bit 7): Reserved bit. This bit must always be set to 0.

DVCKON (bit 6): Frequency divided clock select flag

1) This bit must be set to 1 to select the frequency divided clock derived from the USB 48 MHz clock as the main clock. When the OCR register is configured to select the main clock (CLKCB4 set to 1) in this case, the frequency divided clock is supplied as the system clock.

2) This bit must be set to 0 to select the CF clock as the main clock.

3) When changing the state of this bit, make sure that the system clock selection setting of the OCR register is set to a value other than main clock (CLKCB4 = 0).

DVCKDR (bit 5): Frequency divided clock external output control flag

1) This bit must be set to 1 to generate the frequency divided clock derived from the USB 48 MHz clock output from pin P73. The frequency divided clock is transmitted from pin P73 when the P73 output enable bit (P7 register, bit 7) is set to 1 and the P73 data latch (P7 register, bit 3) is set to 0 in this case.

2) This bit must be set to 0 to suppress the generation of the frequency divided clock to the external circuitry.

P73NDL (bit 4): Reserved bit. This bit must always be set to 0.

CF12BOFF (bit 3): Reserved bit. This bit must always be set to 0.

UDVSEL2 (bit 2): Frequency divided clock frequency select

UDVSEL1 (bit 1): Frequency divided clock frequency select

UDVSEL0 (bit 0): Frequency divided clock frequency select

1) These bits are used to select the frequency of the frequency divided clock derived from the USB 48 MHz clock.

2) The bits must be set to select a frequency divided clock frequency of 8 to 12 MHz when the frequency divided clock is to be supplied as the system clock to drive the USB interface control circuit.

3) When making an attempt to change the frequency divided clock frequency setting to any value other than the initial value (UDVSEL=000), set it to the "clock stopped state" value (UDVSEL=000) before setting up a new value

Example: Changing the frequency divided clock frequency from 12 MHz to 8 MHz

UDVSEL = 100 → 000 → 011

Table 3.19.3   Frequency Divided Clock Frequencies

UDVSEL[2:0]	Frequency (MHz)
000	Frequency divided clock stopped
001	4.8
010	6
011	8
100	12
101	24 (inhibited)
110	4
111	16

3.19.4.2 USB PLL control register (PLLCNT)
1) This register is an 8-bit register that controls the operation of the PLL oscillator circuit for the USB.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE0D	0000 0000	R/W	PLLCNT	SELREF2	SELREF1	SELREF0	PLLTEST	VCOSTP	CMPSTP	PLLCNT1	PLLCNT0

SELREF2 (bit 7): PLL reference clock frequency select

SELREF1 (bit 6): PLL reference clock frequency select

SELREF0 (bit 5): PLL reference clock frequency select
1) These bits are used to select the frequency of the ceramic resonator connected to the CF pins (CF1 and CF2).

Table 3.19.4   Ceramic Resonator Frequencies

SELREF[2:0]	Frequency (MHz)
000	8
001	9
010	10
011	11
100	12
101	2
110	3
111	16

PLLTEST (bit 4): Test bit for the PLL circuit. This bit must always be set to 0.

VCOSTP (bit 3): PLLVCO operation control flag

CMPSTP (bit 2): PLL phase comparator operation control flag
1) This bit must be set to 0 together with VCOSTP and CMPSTP to generate the USB 48 MHz clock from the internal PLL circuit. In this case, P3DDR (FE4DH), bit 4 (P34DDR) and P3 (FE4CH), bit 4 (P34) must be set to 0. In addition, it is necessary to connect an external filter to the P34/UFILT pin as shown in Figure 3.19.3.

2) Must be set to 1 if the PLL circuit is not to be used.

PLLCNT1 (bit 1): Reserved bit.

PLLCNT0 (bit 0): Reserved bit.

3.19.4.3 USB operation control register (USCTRL)
1) This register is used to turn on and off the USB clock.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE80	0000 0000	R/W	USCTRL	USBON	USBRUN	VD3OEN	VD3KIL	IDLFG	IDLEN	DPIEZ	DMIEZ

USBON (bit 7): USB clock control flag

USBRUN (bit 6): USB clock control flag
1) These bits turn on and off the USB clock.

Table 3.19.5   USB Clock Control Settings

USBON	USBRUN	USB Operation
0	—	Enables to detect only USB bus active conditions (*1). The other USB functions are disabled.
1	0	Enables to detect only USB bus reset and USB bus active conditions. The other USB functions are disabled.
1	1	Enables the entire USB block.

*1: Bus state other than bus idle (J state), i.e., K state or SE0

VD3OEN (bit 5): D+ pull-up control flag
1) Setting this bit to 1 causes a high level to be generated from the P70 pin for D+ pull-up. In this case, bit 4 (P70DDR) and bit 0 (P70) of the P7 register (FE5CH) must be set to 0.
2) Setting this bit to 0 disables a high level to be generated from the P70 pin for D+ pull-up. In this case, the P70 pin is held in the "Hi-Z" state if bit 4 (P70DDR) and bit 0 (P70) of the P7 register (FE5CH) are set to 0.

VD3KIL (bit 4): Reserved. Must always be set to 0.

IDLFG (bit 3): Suspend detection flag
1) This flag is set when a suspend condition (staying in bus idle state for 3 ms or longer) is detected.
2) This bit remains set until the suspend condition is released.
3) This flag must be cleared with an instruction.

IDLEN (bit 2): Suspend interrupt request enable flag
1) An interrupt request to vector address 0033H is generated when this bit and IDLFG are set to 1.

DPIEZ (bit 1): D+ pin input enable flag
1) Setting this bit to 1 disables the D+ pin for data read.
2) Setting this bit to 0 enables the D+ pin for data read.
3) This bit must be set to 0 when performing USB communication.

DMIEZ (bit 0):D− pin input enable flag
1) Setting this bit to 1 disables the D- pin for data read.
2) Setting this bit to 0 enables the D- pin for data read.
3) This bit must be set to 0 when performing USB communication.

3.19.4.4   USB port control register (USPORT)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE81	0000 0000	R/W	USPORT	DDRSOF	VD3OE2	USBSIO	SUSPND	DDRDP	DDRDM	PORTDP	PORTDM

DDRSOF (bit 7): SOF detect pulse output control
1) Setting DDRSOF to 1, P7 (FE5CH), bit 6 (P72DDR) to 1, and P7 (FE5CH), bit 2 (P72) to 0 causes an SOF detect pulse (approx. 80 ns) to be generated from P72 on reception of an SOF.

VD3OE2 (bit 6): D+ pull up control flag.

1) Setting this bit to 1 causes a high level to be generated from the P27 pin for D+ pull-up. In this case, bit 7 (P27) of P2 (FE48H) and bit 7 (P27DDR) of P2DDR (FE49H) must be set to 0.

2) Setting this bit to 0 disables the generation of a high level from the P27 pin for D+ pull-up. If bit 7 (P27) of P2 (FE48H) and bit 7 (P27DDR) of P2DDR (FE49H) are set to 0 in this case, the P27 pin is placed in the "Hi-Z" state.

USBSIO (bit 5): Reserved bit. This bit must always be set to 0.

SUSPND (bit 4): USB transceiver operation control flag

1) Setting this bit to 1 places the USB transceiver into the suspended state. The USB transceiver can detect bus active conditions even in this state.

2) When this bit is set to 0, the USB transceiver continues its normal operation.

DDRDP (bit 3): D+ pin general-purpose output control

1) When this bit is set to 1, PORTDP data is generated from the D+ pin.

2) This bit must be set to 0 to perform normal USB communication.

DDRDM (bit 2): D− pin general-purpose output control

1) When this bit is set to 1, PORTDM data is generated from the D- pin.

2) This bit must be set to 0 to perform normal USB communication.

PORTDP (bit 1): D+ port data latch

1) PORTDP data is generated from the D+ pin when DDRDP is set to 1.

2) Reading this bit with an instruction causes the data at the D+ pin to be read in. If USPORT is manipulated using the NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, however, not the data at the D+ pin but the register contents are referenced.

PORTDM (bit 0): D− port data latch

1) PORTDM data is generated from the D- pin when DDRDM is set to 1.

2) Reading this bit with an instruction causes the data at the D- pin to be read in. If USPORT is manipulated using the NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, however, not the data at the D- pin but the register contents are referenced.

3.19.4.5   USB interrupt control register (USBINT)

1) This register controls USB interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE82	0000 0000	R/W	USBINT	BRSFG	BRSEN	BACFG	BACEN	SOFFG	SOFEN	USBINT1	ENPEN

BRSFG (bit 7): Bus reset detect flag

1) This flag is set when a USB bus reset state (SE0 state continuing for 2.5 μs or longer) is detected.

2) The bit remains set until the bus reset state is released.

3) This flag must be cleared with an instruction.

BRSEN (bit 6): Bus reset interrupt request enable flag
1)   An interrupt request to vector address 0033H is generated when this bit and BRSFG are set to 1.

BACFG (bit 5): Bus active detect flag
1)   This flag is set when a USB bus active state (bus state other than bus idle) is detected.
2)   This flag must be cleared with an instruction.

BACEN (bit 4): Bus active interrupt request enable flag
1)   Hold mode release signal and an interrupt request to vector address 0013H are generated when this bit and BACFG are set to 1.

SOFFG (bit 3): SOF detect flag
1)   This flag is set when an SOF is detected.
2)   This flag must be cleared with an instruction.

SOFEN (bit 2): SOF interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit and SOFFG are set to 1.

USBINT1 (bit 1): Reserved bit. This bit must always be set to 0.

ENPEN (bit 0): Endpoint interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit is set to 1 and either one of interrupt sources EP0INT, EP1INT, EP2INT, EP3INT, and EP4INT is established.

3.19.4.6   EP0 interrupt control register (EP0INT)
1)   This register controls endpoint 0 interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE83	0000 0000	R/W	EP0INT	AK0FG	AK0EN	NK0FG	NK0EN	ER0FG	ER0EN	ST0FG	ST0EN

AK0FG (bit 7): EP0 ACK end flag
1)   This flag is set to 1 when the endpoint 0 transaction terminates normally with an ACK. See the column entitled "End Flags" in Table 3.19.11.
2)   This flag must be cleared with an instruction.

AK0EN (bit 6): EP0 ACK interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, AK0FG, and ENPEN are all set to 1.

NK0FG (bit 5): EP0 NAK end flag
1)   This flag is set to 1 when the endpoint 0 transaction terminates with an NAK. See the column entitled "End Flags" in Table 3.19.11.
2)   This flag must be cleared with an instruction.

NK0EN (bit 4): EP0 NAK interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, NK0FG, and ENPEN are all set to 1.

ER0FG (bit 3): EP0 error end flag
1)   This flag is set to 1 when the endpoint 0 transaction terminates with an error. See the column entitled "End Flags" in Table 3.19.11.
2)   This flag must be cleared with an instruction.

ER0EN (bit 2): EP0 error interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, ER0FG, and ENPEN are all set to 1.

ST0FG (bit 1): EP0 stall end flag
1)   This flag is set to 1 when the endpoint 0 transaction terminates with a stall. See the column entitled "End Flags" in Table 3.19.11.
2)   This flag must be cleared with an instruction.

ST0EN (bit 0): EP0 stall interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, ST0FG, and ENPEN are all set to 1.

3.19.4.7   EP1 interrupt control register (EP1INT)
1)   This register controls endpoint 1 interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE84	0000 0000	R/W	EP1INT	AK1FG	AK1EN	NK1FG	NK1EN	ER1FG	ER1EN	ST1FG	ST1EN

AK1FG (bit 7): EP1 ACK end flag
1)   This flag is set to 1 when the endpoint 1 transaction terminates normally with an ACK. See the column entitled "End Flags" in Table 3.19.12.
2)   This flag must be cleared with an instruction.

AK1EN (bit 6): EP1 ACK interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, AK1FG, and ENPEN are all set to 1.

NK1FG (bit 5): EP1 NAK end flag
1)   This flag is set to 1 when the endpoint 1 transaction terminates with an NAK. See the column entitled "End Flags" in Table 3.19.12.
2)   This flag must be cleared with an instruction.

NK1EN (bit 4): EP1 NAK interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, NK1FG, and ENPEN are all set to 1.

ER1FG (bit 3): EP1 error end flag
1)   This flag is set to 1 when the endpoint 1 transaction terminates with an error. See the column entitled "End Flags" in Table 3.19.12.
2)   This flag must be cleared with an instruction.

ER1EN (bit 2): EP1 error interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, ER1FG, and ENPEN are all set to 1

ST1FG (bit 1): EP1 stall end flag
1) This flag is set to 1 when the endpoint 1 transaction terminates with a stall. See the column entitled "End Flags" in Table 3.19.12.
2) This flag must be cleared with an instruction.

ST1EN (bit 0): EP1 stall interrupt request enable flag
1) An interrupt request to vector address 003BH is generated when this bit, ST1FG, and ENPEN are all set to 1.

3.19.4.8 EP2 interrupt control register (EP2INT)
1) This register controls endpoint 2 interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE85	0000 0000	R/W	EP2INT	AK2FG	AK2EN	NK2FG	NK2EN	ER2FG	ER2EN	ST2FG	ST2EN

AK2FG (bit 7): EP2 ACK end flag
1) This flag is set to 1 when the endpoint 2 transaction terminates normally with an ACK. See the column entitled "End Flags" in Table 3.19.12.
2) This flag must be cleared with an instruction.

AK2EN (bit 6): EP2 ACK interrupt request enable flag
1) An interrupt request to vector address 003BH is generated when this bit, AK2FG, and ENPEN are all set to 1.

NK2FG (bit 5): EP2 NAK end flag
1) This flag is set 1 when the endpoint 2 transaction terminates with an NAK. See the column entitled "End Flags" in Table 3.19.12.
2) This flag must be cleared with an instruction.

NK2EN (bit 4): EP2 NAK interrupt request enable flag
1) An interrupt request to vector address 003BH is generated when this bit, NK2FG, and ENPEN are all set to 1.

ER2FG (bit 3): EP2 error end flag
1) This flag is set to 1 when the endpoint 2 transaction terminates with an error. See the column entitled "End Flags" in Table 3.19.12.
2) This flag must be cleared with an instruction.

ER2EN (bit 2): EP2 error interrupt request enable flag
1) An interrupt request to vector address 003BH is generated when this bit, ER2FG, and ENPEN are all set to 1.

ST2FG (bit 1): EP2 stall end flag
1) This flag is set to 1 when the endpoint 2 transaction terminates with a stall. See the column entitled "End Flags" in Table 3.19.12.
2) This flag must be cleared with an instruction.

ST2EN (bit 0): EP2 stall interrupt request enable flag
1) An interrupt request to vector address 003BH is generated when this bit, ST2FG, and ENPEN are all set to 1.

3.19.4.9 EP3 interrupt control register (EP3INT)

1) This register controls endpoint 3 interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE86	0000 0000	R/W	EP3INT	AK3FG	AK3EN	NK3FG	NK3EN	ER3FG	ER3EN	ST3FG	ST3EN

AK3FG (bit 7): EP3 ACK end flag

1) This flag is set to 1 when the endpoint 3 transaction terminates normally with an ACK. See the column entitled "End Flags" in Table 3.19.12.

2) This flag must be cleared with an instruction.

AK3EN (bit 6): EP3 ACK interrupt request enable flag

1) An interrupt request to vector address 003BH is generated when this bit, AK3FG, and ENPEN are all set to 1.

NK3FG (bit 5): EP3 NAK end flag

1) This flag is set to 1 when the endpoint 3 transaction terminates with an NAK. See the column entitled "End Flags" in Table 3.19.12.

2) This flag must be cleared with an instruction.

NK3EN (bit 4): EP3 NAK interrupt request enable flag

1) An interrupt request to vector address 003BH is generated when this bit, NK3FG, and ENPEN are all set to 1.

ER3FG (bit 3): EP3 error end flag

1) This flag is set to 1 when the endpoint 3 transaction terminates with an error. See the column entitled "End Flags" in Table 3.19.12.

2) This flag must be cleared with an instruction.

ER3EN (bit 2): EP3 error interrupt request enable flag

1) An interrupt request to vector address 003BH is generated when this bit, ER3FG, and ENPEN are all set to 1.

ST3FG (bit 1): EP3 stall end flag

1) This flag is set to 1 when the endpoint 3 transaction terminates with a stall. See the column entitled "End Flags" in Table 3.19.12.

2) This flag must be cleared with an instruction.

ST3EN (bit 0): EP3 stall interrupt request enable flag

1) An interrupt request to vector address 003BH is generated when this bit, ST3FG, and ENPEN are all set to 1.

3.19.4.10 EP4 interrupt control register (EP4INT)

1) This register controls endpoint 4 interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE87	0000 0000	R/W	EP4INT	AK4FG	AK4EN	NK4FG	NK4EN	ER4FG	ER4EN	ST4FG	ST4EN

AK4FG (bit 7): EP4 ACK end flag

1) This flag is set to 1 when the endpoint 4 transaction terminates normally with an ACK. See the column entitled "End Flags" in Table 3.19.12.

2) This flag must be cleared with an instruction.

AK4EN (bit 6): EP4 ACK interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, AK4FG, and ENPEN are all set to 1.

NK4FG (bit 5): EP4 NAK end flag
1)   This flag is set to 1 when the endpoint 4 transaction terminates with an NAK. See the column entitled "End Flags" in Table 3.19.12.
2)   This flag must be cleared with an instruction.

NK4EN (bit 4): EP4 NAK interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, NK4FG, and ENPEN are all set to 1.

ER4FG (bit 3): EP4 error end flag
1)   This flag is set to 1 when the endpoint 4 transaction terminates with an error. See the column entitled "End Flags" in Table 3.19.12.
2)   This flag must be cleared with an instruction.

ER4EN (bit 2): EP4 error interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, ER4FG, and ENPEN are all set to 1.

ST4FG (bit 1): EP4 stall end flag
1)   This flag is set to 1 when the endpoint 4 transaction terminates with a stall. See the column entitled "End Flags" in Table 3.19.12.
2)   This flag must be cleared with an instruction.

ST4EN (bit 0): EP4 stall interrupt request enable flag
1)   An interrupt request to vector address 003BH is generated when this bit, ST4FG, and ENPEN are all set to 1.

3.19.4.11 Frame number register (FRAMEL, FRAMEH)
1)   This register is loaded with the frame number when SOF data is received.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE8A	0000 0000	R	FRAMEL	FRM07	FRM06	FRM05	FRM04	FRM03	FRM02	FRM01	FRM00
FE8B	HHHH H000	R	FRAMEH	-	-	-	-	-	FRM10	FRM09	FRM08

3.19.4.12 USB address register (USBADR)
1)   This register is used to set the address of the USB device.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE8C	0000 0000	R/W	USBADR	ADREN	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

ADREN (bit 7): Device address enable flag
1)   A 1 in this bit enables ADDR[6:0]. Of the tokens sent from the host, only the tokens whose device address matches the contents of ADDR[6: 0] are processed.
2)   A 0 in this bit disables ADDR[6:0]. Of the tokens sent from the host, only the tokens whose device address is 0 are processed.

ADDR6 (bit 6): Device address

ADDR5 (bit 5): Device address

ADDR4 (bit 4): Device address

ADDR3 (bit 3): Device address

ADDR2 (bit 2): Device address

ADDR1 (bit 1): Device address

ADDR0 (bit 0): Device address

    1)   These bits are used to set the device address assigned by the host.

3.19.4.13 Endpoint information register (EPINFO)

    1)   This register contains information about the endpoint number and token type.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE8D	0000 0000	R	EPINFO	EPNO3	EPNO2	EPNO1	EPNO0	TKN1	TKN0	CTKN1	CTKN0

EPNO3 (bit 7): Endpoint number

EPNO2 (bit 6): Endpoint number

EPNO1 (bit 5): Endpoint number

EPNO0 (bit 4): Endpoint number

    1)   These register bit positions are loaded with the endpoint number from the token packet when one of the following conditions is established:

        (1)   The transaction terminates normally with an ACK.

        (2)   NKnEN (n = 0 to 4) is set to 1 and the transaction for endpoint n terminates with an NAK.

        (3)   ERnEN (n = 0 to 4) is set to 1 and the transaction for endpoint n terminates with an error.

        (4)   STnEN (n = 0 to 4) is set to 1 and the transaction for endpoint n terminates with a stall.

TKN1 (bit 3): Token ID

TKN0 (bit 2): Token ID

    1)   These register bit positions are loaded with the token ID when either of the above conditions (1) to (4) is established.

CTKN1 (bit 1): EP0 token ID

CTKN0 (bit 0): EP0 token ID

    1)   These register bit positions are loaded with the token ID when either of the above conditions (1) to (4) is established for an endpoint 0 transaction.

Table 3.19.6   Token IDs

Token	Token ID
OUT	00
IN	10
SETUP	11

3.19.4.14 EP0 status register (EP0STA)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE8E	0000 0000	R/W	EP0STA	E0EN	E0TGL	E0OVR	E0STL	E0ACK	E0CSU	E0CST	E0CRW

E0EN (bit 7): EP0 operation enable flag
1) When this bit is set to 1, the USB interface processes the token at endpoint 0.
2) When this bit is set to 0, the USB interface does not process the token at endpoint 0.

E0TGL (bit 6): EP0 data toggle
1) This bit is automatically set to 1 when a SETUP transaction terminates normally with an ACK.
2) On an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E0TGL. The state of E0TGL is automatically inverted when the transaction terminates normally with an ACK.
3) On an IN transaction, the USB interface transmits the data packet with a packet ID matching E0TGL. The state of E0TGL is automatically inverted when the USB interface receives an ACK from the host.
4) This bit is automatically cleared to 0 when an OUT transaction or IN transaction in the status stage terminates normally with an ACK.

E0OVR (bit 5): EP0 payload over flag
1) This bit is automatically set to 1 when the USB interface receives a volume of data exceeding the maximum payload size defined in EP0MP.
2) This flag must be cleared with an instruction.

E0STL (bit 4): EP0 stall flag
1) When this bit is set to 1, the USB interface returns a STALL handshake for IN and OUT transactions.
2) This bit is automatically set to 1 when a STALL handshake is returned due to a protocol violation.
3) This bit is automatically cleared to 0 when a SETUP transaction terminates normally with an ACK.

E0ACK (bit 3): EP0 ACK flag
1) When this bit is set to 1, the USB interface transmits a data packet for an IN transaction and an ACK handshake for an OUT transaction.
2) When this bit is set to 0, the USB interface returns an NAK handshake for IN and OUT transactions.
3) See Table 3.19.11.

E0CSU (bit 2): EP0 complete setup stage flag
1) This bit is automatically set to 1 when a SETUP transaction terminates normally with an ACK.
2) This bit is automatically cleared to 0 when an IN transaction in the status stage terminates normally with an ACK.
3) See Table 3.19.11.

E0CST (bit 1): EP0 status stage flag
1) A 1 in this bit indicates that the USB interface is in the control transfer status stage.
2) This bit is automatically cleared to 0 when a SETUP transaction terminates normally with an ACK.
3) This bit is automatically set to 1 when an OUT transaction in the status stage terminates normally with an ACK.
4) See Table 3.19.11.

E0CRW (bit 0): EP0 transfer direction flag
1) A 1 in this bit identifies a control read transfer.
2) A 0 in this bit identifies a control write transfer.
3) This bit is automatically cleared to 0 when a SETUP transaction terminates normally with an ACK.
4) See Table 3.19.11.

3.19.4.15 EP0 maximum payload register (EP0MP)
1) This register is used to define the maximum payload size of endpoint 0.
2) The legitimate values are 08[H], 10[H], 20[H], and 40[H].

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE8F	H000 0000	R/W	EP0MP	-	E0MP6	E0MP5	E0MP4	E0MP3	E0MP2	E0MP1	E0MP0

3.19.4.16 EP0 receive data count register (EP0RX)
1) This register indicates the number of data bytes received at endpoint 0.
2) The contents of this register are updated when a SETUP or OUT transaction for endpoint 0 terminates normally with an ACK.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE90	H000 0000	R/W	EP0RX	-	E0RX6	E0RX5	E0RX4	E0RX3	E0RX2	E0RX1	E0RX0

3.19.4.17 EP0 transmit data count register (EP0TX)
1) This register is used to set the number of data bytes to be transmitted through endpoint 0.
2) The legitimate value range is from 00[H] to 40[H].

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE91	H000 0000	R/W	EP0TX	-	E0TX6	E0TX5	E0TX4	E0TX3	E0TX2	E0TX1	E0TX0

3.19.4.18 EP1 status register (EP1STA)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE92	0000 0000	R/W	EP1STA	E1EN	E1TGL	E1OVR	E1STL	E1ACK	E1DIR	E1ISO	E1BNK

E1EN (bit 7): EP1 operation enable flag
1) When this bit is set to 1, the USB interface processes the token at endpoint 1.
2) When this bit is set to 0, the USB interface does not process the token at endpoint 1.

E1TGL (bit 6): EP1 data toggle
1) On an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E1TGL. The state of E1TGL is automatically inverted when the transaction terminates normally with an ACK. The E1TGL state, however, does not invert for isochronous transfers.
2) On an IN transaction, the USB interface transmits the data packet with a packet ID matching E1TGL. The state of E1TGL is automatically inverted when the USB interface receives an ACK from the host.

E1OVR (bit 5): EP1 payload over flag

    1)   This bit is automatically set to 1 when the USB interface receives a volume of data exceeding the maximum payload size defined in EP1CNT.

    2)   This flag must be cleared with an instruction.

E1STL (bit 4): EP1 stall flag

    1)   The USB interface returns a STALL handshake for IN and OUT transactions.

E1ACK (bit 3): EP1 ACK flag

    1)   When this bit is set to 1, the USB interface transmits a data packet for an IN transaction and an ACK handshake for an OUT transaction.

    2)   When this bit is set to 0, the USB interface returns an NAK handshake for IN and OUT transactions.

    3)   See Table 3.19.12.

E1DIR (bit 2): EP1 transfer direction flag

    1)   When this bit is set to 1, the USB interface processes only IN tokens at endpoint 1.

    2)   When this bit is set to 0, the USB interface processes only OUT tokens at endpoint 1.

    3)   See Table 3.19.12.

E1ISO (bit 1): EP1 isochronous transfer flag

    1)   When this bit is set to 1, the transfer type of endpoint 1is isochronous transfer.

    2)   When this bit is set to 0, the transfer type of endpoint 1 is either bulk transfer or interrupt transfer.

    3)   See Table 3.19.12.

E1BNK (bit 0): EP1 transfer RAM address control flag

    1)   Setting this bit and E1OF[1:0] (EPADOF, bits 1 and 0) allocates the data transmit/receive buffer areas for endpoint 1.

Table 3.19.7   EP1 Buffer Areas

E1BNK	E1OF1	E1OF0	RAM address
0	0	0	0280H to 02BFH
0	0	1	0290H to 02CFH
0	1	0	02A0H to 02DFH
0	1	1	02B0H to 02EFH
1	0	0	02C0H to 02FFH
1	0	1	02D0H to 030FH
1	1	0	02E0H to 031FH
1	1	1	02F0H to 032FH

3.19.4.19 EP2 status register (EP2STA)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE93	0000 0000	R/W	EP2STA	E2EN	E2TGL	E2OVR	E2STL	E2ACK	E2DIR	E2ISO	E2BNK

E2EN (bit 7): EP2 operation enable flag
1) When this bit is set to 1, the USB interface processes the token at endpoint 2.
2) When this bit is set to 0, the USB interface does not process the token at endpoint 2.

E2TGL (bit 6): EP2 data toggle
1) On an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E2TGL. The state of E2TGL is automatically inverted when the transaction terminates normally with an ACK. The E2TGL state does not invert, however, for isochronous transfers.
2) On an IN transaction, the USB interface transmits the data packet with a packet ID matching E2TGL. The state of E2TGL is automatically inverted when the USB interface receives an ACK from the host.

E2OVR (bit 5): EP2 payload over flag
1) This bit is automatically set to 1 when the USB interface receives a volume of data exceeding the maximum payload size defined in EP2CNT.
2) This flag must be cleared with an instruction.

E2STL (bit 4): EP2 stall flag
1) The USB interface returns a STALL handshake for IN and OUT transactions.

E2ACK (bit 3): EP2 ACK flag
1) When this bit is set to 1, the USB interface transmits a data packet for an IN transaction and an ACK handshake for an OUT transaction.
2) When this bit is set to 0, the USB interface returns an NAK handshake for IN and OUT transactions.
3) See Table 3.19.12.

E2DIR (bi 2): EP2 transfer direction flag
1) When this bit is set to 1, the USB interface processes only IN tokens at endpoint 2.
2) When this bit is set to 0, the USB interface processes only OUT tokens at endpoint 2.
3) See Table 3.19.12.

E2ISO (bit 1): EP2 isochronous transfer flag
1) When this bit is set to 1, the transfer type of endpoint 2 is isochronous transfer.
2) When this bit is set to 0, the transfer type of endpoint 2 is either bulk transfer or interrupt transfer.
3) See Table 3.19.12.

E2BNK (bit 0): EP2 transfer RAM address control flag

1) Setting this bit and E2OF[1:0] (EPADOF, bits 3 and 2) allocates the data transmit/receive buffer areas for endpoint 2.

Table 3.19.8   EP2 Buffer Areas

E2BNK	E2OF1	E2OF0	RAM address
0	0	0	0300H to 033FH
0	0	1	0310H to 034FH
0	1	0	0320H to 035FH
0	1	1	0330H to 036FH
1	0	0	0340H to 037FH
1	0	1	0350H to 038FH
1	1	0	0360H to 039FH
1	1	1	0370H to 03AFH

3.19.4.20 EP3 status register (EP3STA)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE94	0000 0000	R/W	EP3STA	E3EN	E3TGL	E3OVR	E3STL	E3ACK	E3DIR	E3ISO	E3BNK

E3EN (bit 7): EP3 operation enable flag

1) When this bit is set to 1, the USB interface processes the token at endpoint 3.

2) When this bit is set to 0, the USB interface does not process the token at endpoint 3.

E3TGL (bit 6): EP3 data toggle

1) On an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E3TGL. The state of E3TGL is automatically inverted when the transaction terminates normally with an ACK. The E3TGL state does not invert, however, for isochronous transfers.

2) On an IN transaction, the USB interface transmits the data packet with a packet ID matching E3TGL. The state of E3TGL is automatically inverted when the USB interface receives an ACK from the host.

E3OVR (bit 5): EP3 payload over flag

1) This bit is automatically set to 1 when the USB interface receives a volume of data exceeding the maximum payload size defined in EP3CNT.

2) This flag must be cleared with an instruction.

E3STL (bit 4): EP3 stall flag

1) The USB interface returns a STALL handshake for IN and OUT transactions.

E3ACK (bit 3): EP3 ACK flag

1) When this bit is set to 1, the USB interface transmits a data packet for an IN transaction and an ACK handshake for an OUT transaction.

2) When this bit is set to 0, the USB interface returns an NAK handshake for IN and OUT transactions.

3) See Table 3.19.12.

E3DIR (bit 2): EP3 transfer direction flag
1) When this bit is set to 1, the USB interface processes only the IN token at endpoint 3.
2) When this bit is set to 0, the USB interface processes only the OUT token at endpoint 3.
3) See Table 3.19.12.

E3ISO (bit 1): EP3 isochronous transfer flag
1) When this bit is set to 1, the transfer type of endpoint 3 is isochronous transfer.
2) When this bit is set to 0, the transfer type of endpoint 3 is either bulk transfer or interrupt transfer.
3) See Table 3.19.12

E3BNK (bit 0): EP3 transfer RAM address control flag
1) Setting this bit and E3OF[1:0] (EPADOF, bits 5 and 4) allocates the data transmit/receive buffer areas for endpoint 3.

Table 3.19.9   EP3 Buffer Areas

E3BNK	E3OF1	E3OF0	RAM address
0	0	0	0380H to 03BFH
0	0	1	0390H to 03CFH
0	1	0	03A0H to 03DFH
0	1	1	03B0H to 03EFH
1	0	0	03C0H to 03FFH
1	0	1	03D0H to 040FH
1	1	0	03E0H to 041FH
1	1	1	03F0H to 042FH

3.19.4.21   EP4 status register (EP4STA)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE95	0000 0000	R/W	EP4STA	E4EN	E4TGL	E4OVR	E4STL	E4ACK	E4DIR	E4ISO	E4BNK

E4EN (bit 7): EP4 operation enable flag
1) When this bit is set to 1, the USB interface processes the token at endpoint 4.
2) When this bit is set to 0, the USB interface does not process the token at endpoint 4.

E4TGL (bit 6): EP4 data toggle
1) On an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E4TGL. The state of E4TGL is automatically inverted when the transaction terminates normally with an ACK. The E4TGL state does not invert, however, for isochronous transfers.
2) On an IN transaction, the USB interface transmits the data packet with a packet ID matching E4TGL. The state of E4TGL is automatically inverted when the USB interface receives an ACK from the host.

E4OVR (bit 5): EP4 payload over flag
1) This bit is automatically set to 1 when the USB interface receives a volume of data exceeding the maximum payload size defined in EP4CNT.
2) This flag must be cleared with an instruction.

E4STL (bit 4): EP4 stall flag
    1)   The USB interface returns a STALL handshake for IN and OUT transactions.

E4ACK (bit 3): EP4 ACK flag
    1)   When this bit is set to 1, the USB interface transmits a data packet for an IN transaction and an ACK handshake for an OUT transaction.
    2)   When this bit is set to 0, the USB interface returns an NAK handshake for IN and OUT transactions.
    3)   See Table 3.19.12.

E4DIR (bit 2): EP4 transfer direction flag
    1)   When this bit is set to 1, the USB interface processes only the IN token at endpoint 4.
    2)   When this bit is set to 0, the USB interface processes only the OUT token at endpoint 4.
    3)   See Table 3.19.12.

E4ISO (bit 1): EP4 isochronous transfer flag
    1)   When this bit is set to 1, the transfer type of endpoint 4 is isochronous transfer.
    2)   When this bit is set to 0, the transfer type of endpoint 4 is either bulk transfer or interrupt transfer.
    3)   See Table 3.19.12.

E4BNK (bit 0): EP4 transfer RAM address control flag
    1)   Setting this bit and E4OF[1:0] (EPADOF, bits 7 and 6) allocates the data transmit/receive buffer areas for endpoint 4.

Table 3.19.10   EP4 Buffer Areas

E4BNK	E4OF1	E4OF0	RAM address
0	0	0	0380H to 03BFH
0	0	1	0390H to 03CFH
0	1	0	03A0H to 03DFH
0	1	1	03B0H to 03EFH
1	0	0	03C0H to 03FFH
1	0	1	03D0H to 040FH
1	1	0	03E0H to 041FH
1	1	1	03F0H to 042FH

3.19.4.22   EP1 count register (EP1CNT)
    1)   This register must be loaded with the number of transmit data bytes for endpoint 1 when the transfer direction of endpoint 1 is IN (E1DIR=1).
    2)   When the transfer direction of endpoint 1 is OUT (E1DIR=0), this register must be loaded with the maximum payload size of endpoint 1.
    3)   The legitimate value range is from 00[H] to 40[H].

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE98	H000 0000	R/W	EP1CNT	-	E1CN6	E1CN5	E1CN4	E1CN3	E1CN2	E1CN1	E1CN0

3.19.4.23 EP1 receive data count register (EP1RX)

1) This register indicates the number of data bytes received at endpoint 1.

2) The contents of this register are updated when an OUT transaction for endpoint 1 terminates normally with an ACK.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE99	H000 0000	R	EP1RX	-	E1RX6	E1RX5	E1RX4	E1RX3	E1RX2	E1RX1	E1RX0

3.19.4.24 EP2 count register (EP2CNT)

1) This register must be loaded with the number of transmit data bytes for endpoint 2 when the transfer direction of endpoint 2 is IN (E2DIR=1).

2) When the transfer direction of endpoint 2 is OUT (E2DIR=0), this register must be loaded with the maximum payload size of endpoint 2.

3) The legitimate value range is from 00[H] to 40[H].

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE9A	H000 0000	R/W	EP2CNT	-	E2CN6	E2CN5	E2CN4	E2CN3	E2CN2	E2CN1	E2CN0

3.19.4.25 EP2 receive data count register (EP2RX)

1) This register indicates the number of data bytes received at endpoint 2.

2) The contents of this register are updated when an OUT transaction for endpoint 2 terminates normally with an ACK.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE9B	H000 0000	R/W	EP2RX	-	E2RX6	E2RX5	E2RX4	E2RX3	E2RX2	E2RX1	E2RX0

3.19.4.26 EP3 count register (EP3CNT)

1) This register must be loaded with the number of transmit data bytes for endpoint 3 when the transfer direction of endpoint 3 is IN (E3DIR=1).

2) When the transfer direction of endpoint 3 is OUT (E3DIR=0), this register must be loaded with the maximum payload size of endpoint 3.

3) The legitimate value range is from 00[H] to 40[H].

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE9C	H000 0000	R/W	EP3CNT	-	E3CN6	E3CN5	E3CN4	E3CN3	E3CN2	E3CN1	E3CN0

3.19.4.27 EP3 receive data count register (EP3RX)

1) This register indicates the number of data bytes received at endpoint 3.

2) The contents of this register are updated when an OUT transaction for endpoint 3 terminates normally with an ACK.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE9D	H000 0000	R/W	EP3RX	-	E3RX6	E3RX5	E3RX4	E3RX3	E3RX2	E3RX1	E3RX0

3.19.4.28 EP4 count register (EP4CNT)

1) This register must be loaded with the number of transmit data bytes for endpoint 4 when the transfer direction of endpoint 4 is IN (E4DIR=1).

2) When the transfer direction of endpoint 4 is OUT (E4DIR=0), this register must be loaded with the maximum payload size of endpoint 4.

3) The legitimate value range is from 00[H] to 40[H].

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE9E	H000 0000	R/W	EP4CNT	-	E4CN6	E4CN5	E4CN4	E4CN3	E4CN2	E4CN1	E4CN0

3.19.4.29 EP4 receive data count register (EP4RX)

1) This register indicates the number of data bytes received at endpoint 4.
2) The contents of this register are updated when an OUT transaction for endpoint 4 terminates normally with an ACK.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE9F	H000 0000	R/W	EP4RX	-	E4RX6	E4RX5	E4RX4	E4RX3	E4RX2	E4RX1	E4RX0

3.19.4.30 Test register 0 (TESTR0)

1) TESTR0 is a test register.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEAC	0000 0000	R/W	TESTR0	DPLTST	CMPTST	CMPKIL	TTXCLK	TTXREQ	TADR2	TADR1	P71NDL

DPLTST (bit 7): Test bit. This bit must always be set to 0.

CMPTST (bit 6): Test bit. This bit must always be set to 0.

CMPKIL (bit 5): Test bit. This bit must always be set to 0.

TTXCLK (bit 4): Test bit. This bit must always be set to 0.

TTXREQ (bit 3): Test bit. This bit must always be set to 0.

TADR2 (bit 2): Test bit. This bit must always be set to 0.

TADR1 (bit 1): Test bit. This bit must always be set to 0.

P71NDL (bit 0): Test bit. This bit must always be set to 0.

3.19.4.31 Test register 1 (TESTR1)

1) TESTR1 is a test register.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEAD	0000 0000	R	TESTR1	TDAT7	TDAT6	TDAT5	TDAT4	TDAT3	TDAT2	TDAT1	TDAT0

3.19.4.32 Endpoint buffer offset register (EPADOF)

1) This register designates the buffer offsets for endpoints 1 to 4.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEAE	0000 0000	R/W	EPADOF	E4OF1	E4OF0	E3OF1	E3OF0	E2OF1	E2OF0	E1OF1	E1OF0

E4OF1 (bit 7):   EP4 buffer offset
E4OF0 (bit 6):   EP4 buffer offset
   1)   See Table 3.19.10.

E3OF1 (bit 5):   EP3 buffer offset
E3OF0 (bit 4):   EP3 buffer offset
   1)   See Table 3.19.9.

E2OF1 (bit 3):   EP2 buffer offset
E2OF0 (bit 2):   EP2 buffer offset
   1)   See Table 3.19.8.

E1OF1 (bit 1):   EP1 buffer offset
E1OF0 (bit 0):   EP1 buffer offset
   1)   See Table 3.19.7.

Table 3.19.11   Endpoint 0 Status Register Transition Chart

EP0STA[3:0] ACK	CSU	CST	CRW	Token	Receive toggle	Receive data	Receive buffer	Response	EP0STA[3:0] ACK	CSU	CST	CRW	End flag
0	0	0	0	SETUP	-	Invalid	Update	---	0	0	0	0	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	-	-	---	0	0	0	0	-
				IN	-	-	-	---	0	0	0	0	-
0	0	0	1	SETUP	-	Invalid	Update	---	0	0	0	1	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	-	-	---	0	0	0	1	-
				IN	-	-	-	---	0	0	0	1	-
0	0	1	0	SETUP	-	Invalid	Update	---	0	0	1	0	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	-	-	---	0	0	1	0	-
				IN	-	-	-	---	0	0	1	0	-
0	0	1	1	SETUP	-	Invalid	Update	---	0	0	1	1	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	-	-	---	0	0	1	1	-
				IN	-	-	-	---	0	0	1	1	-
0	1	0	0	SETUP	-	Invalid	Update	---	0	1	0	0	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	Invalid	-	---	0	1	0	0	-
				OUT	-	Valid	-	NAK	0	1	0	0	NAK
				IN	-	-	-	NAK	0	1	0	0	NAK
0	1	0	1	SETUP	-	Invalid	Update	---	0	1	0	1	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	Invalid	-	---	0	1	0	1	-
				OUT	-	Valid	-	NAK	0	1	0	1	NAK
				IN	-	-	-	NAK	0	1	0	1	NAK
0	1	1	0	SETUP	-	Invalid	Update	---	0	1	1	0	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	Invalid	-	---	0	1	1	0	-
				OUT	-	Valid	-	NAK	0	1	1	0	NAK
				IN	-	-	-	NAK	0	1	1	0	NAK
0	1	1	1	SETUP	-	Invalid	Update	---	0	1	1	1	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	Invalid	-	---	0	1	1	1	-
				OUT	-	Valid	-	NAK	0	1	1	1	NAK
				IN	-	-	-	NAK	0	1	1	1	NAK
1	0	0	0	SETUP	-	Invalid	Update	---	1	0	0	0	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	-	-	---	1	0	0	0	-
				IN	-	-	-	---	1	0	0	0	-
1	0	0	1	SETUP	-	Invalid	Update	---	1	0	0	1	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	-	-	---	1	0	0	1	-
				IN	-	-	-	---	1	0	0	1	-
1	0	1	0	SETUP	-	Invalid	Update	---	1	0	1	0	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	-	-	---	1	0	1	0	-
				IN	-	-	-	---	1	0	1	0	-
1	0	1	1	SETUP	-	Invalid	Update	---	1	0	1	1	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	-	-	---	1	0	1	1	-
				IN	-	-	-	---	1	0	1	1	-

(Continued on next page.)

(Continued from preceding page.)

EP0STA[3:0] ACK	CSU	CST	CRW	Token	Receive toggle	Receive data	Receive buffer	Response	EP0STA[3:0] ACK	CSU	CST	CRW	End flag
1	1	0	0	SETUP	-	Invalid	Update	---	1	1	0	0	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	Mis-match	Invalid	-	---	1	1	0	0	Error
					Mis-match	Valid	-	ACK	1	1	0	0	-
					Match	Invalid	Update	---	1	1	0	0	Error
					Match	Valid	Update	ACK	0	1	0	0	ACK
				IN	-	-	-	Status-IN(*1)	0	0	0	0	ACK (*3)
1	1	0	1	SETUP	-	Invalid	Update	---	1	1	0	1	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	Invalid	-	---	1	1	0	1	Error
				OUT	-	Valid	-	ACK	0	1	1	1	ACK
				IN	-	-	-	Tx Data (*2)	0	1	0	1	ACK (*3)
1	1	1	0	SETUP	-	Invalid	Update	---	1	1	1	0	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	Invalid	-	---	1	1	1	0	Error
				OUT	-	Valid	-	STALL	1	1	1	0	STALL
				IN	-	-	-	Status-IN(*1)	0	0	0	0	ACK (*3)
1	1	1	1	SETUP	-	Invalid	Update	---	1	1	1	1	Error
				SETUP	-	Valid	Update	ACK	0	1	0	0	ACK
				OUT	-	Invalid	-	---	1	1	1	1	Error
				OUT	-	Valid	-	ACK	0	1	1	1	ACK
				IN	-	-	-	STALL	1	1	1	1	STALL

Shaded columns contain register contents, a receive buffer update, or a response to the token from the host.

*1: A data packet containing no data is transmitted.

*2: The number of bytes specified in EP0TX are transmitted.

*3: If the reception of an ACK packet from host fails, the error end flag (bit 3) in EP0INT is automatically set to 1 but the ACK flag (bit 3) in EP0STA is not cleared automatically.

Table 3.19.12   Endpoint n (n=1 to 4) Status Register Transition Chart

STL	ACK	DIR	ISO	Token	Receive toggle	Receive data	Receive buffer	Response	STL	ACK	DIR	ISO	End flag
0	0	0	0	OUT	-	Invalid	-	---	0	0	0	0	Error
				OUT		Valid	-	NAK	0	0	0	0	NAK
				IN	-	-	-	---	0	0	0	0	-
0	0	0	1	OUT	-	Invalid	-	---	0	0	0	1	Error
				OUT		Valid		---	0	0	0	1	-
				IN	-	-	-	---	0	0	0	1	-
0	0	1	0	OUT	-	-	-	---	0	0	1	0	-
				IN	-	-	-	NAK	0	0	1	0	NAK
0	0	1	1	OUT	-	-	-	---	0	0	1	1	-
				IN	-	-	-	Tx 0(*1)	0	0	1	1	-
0	1	0	0	OUT	Mis-match	Invalid	-	---	0	1	0	0	Error
						Valid	-	ACK	0	1	0	0	-
					Match	Invalid	Update	---	0	1	0	0	Error
						Valid	Update	ACK	0	0	0	0	ACK
				IN	-	-	-	---	0	1	0	0	-
0	1	0	1	OUT	-	Invalid	Update	---	0	1	0	1	Error
						Valid	Update	---	0	0	0	1	ACK
				IN	-	-	-	---	0	1	0	1	-
0	1	1	0	OUT	-	-	-	---	0	1	1	0	-
				IN	-	-	-	Tx Data (*2)	0	0	1	0	ACK
0	1	1	1	OUT	-	-	-	---	0	1	1	1	-
				IN	-	-	-	Tx Data (*2)	0	0	1	1	ACK
1	0	0	0	OUT	-	Invalid		---	1	0	0	0	Error
						Valid		STALL	1	0	0	0	STALL
				IN	-	-	-	---	1	0	0	0	-
1	0	0	1	OUT	-	Invalid	-	---	1	0	0	1	Error
						Valid	-	---	1	0	0	1	-
				IN	-	-	-	---	1	0	0	1	-
1	0	1	0	OUT	-	-	-	---	1	0	1	0	-
				IN			-	-	STALL	1	0	1	0	STALL
1	0	1	1	OUT	-	-	-	---	1	0	1	1	-
				IN	-	-	-	Tx 0(*1)	1	0	1	1	-
1	1	0	0	OUT	-	Invalid	-	---	1	1	0	0	Error
						Valid	-	STALL	1	1	0	0	STALL
				IN	-	-	-		1	1	0	0	-
1	1	0	1	OUT	-	Invalid	Update	---	1	1	0	1	Error
						Valid	Update	---	1	0	0	1	ACK
				IN	-	-	-	---	1	1	0	1	-
1	1	1	0	OUT	-	-	-	---	1	1	1	0	-
				IN	-	-	-	STALL	1	1	1	0	STALL
1	1	1	1	OUT	-	-	-	---	1	1	1	1	-
				IN	-	-	-	Tx Data (*2)	1	0	1	1	ACK

Shaded columns contain register contents, a receive buffer update, or a response to the token from the host.

*1:   A data packet containing no data is transmitted.

*2:   The number of bytes specified in EPnTX are transmitted.

3.19.5   USB Communication Examples

3.19.5.1   Setting up clocks

1)   Set up PLLCNT (FE0DH) according to the frequency of the resonator connected across the CF1 and CF2 pins. See 3.19.4.2, "USB PLL control register."

2)   Set up USBDIV (FE04H) when using, as the system clock, the clock derived by dividing the 48 MHz clock which is generated by the internal PLL for the USB. See 3.19.4.1, "USB frequency divided clock control register."

3)   Set both bit 7 (CLKSGL) and bit 4 (CLKCB4) of the OCR register to 1 to select the main clock as the system clock.

* When performing USB communication, it is necessary to set the system clock frequency to 8 to 16 MHz.

Table 3.19.13   Sample Clock Settings

Resonator Frequency	PLLCNT	USBDIV	OCR	System Clock Frequency
8MHz	00[H]	00[H]	92[H]	8 MHz
		44[H]		12 MHz
		47[H]		16 MHz
12 MHz	80[H]	00[H]		12 MHz
		47[H]		16 MHz

3.19.5.2   Configuration for USB communication

Refer to the separate document entitled "USB Application Note."

3.19.6   USB Interface HALT Mode Operation

1)   When HALT mode is entered, USB communication involving data transmission and reception cannot proceed normally since the automatic data transfer between RAM (endpoint buffer) and the transmit/receive buffer is interrupted in that case.

2)   HALT mode can be released by generating a USB interface interrupt that involves neither data transmission nor reception.

3.20 Infrared Remote Control Receiver Circuit 2 (REMOREC2)

3.20.1 Overview

This series of microcontrollers is equipped with an infrared remote control receiver circuit 2 (REMOREC2) that has the following features and functions:

1) Noise filtering
2) Supports 5 receive formats.
   • Receive format A

Guide pulse	: Half clock
Data encoding system	: PPM (Pulse Position Modulation)
Stop bits	: No

   • Receive format B (supporting repeat code reception)

Guide pulse	: Clock
Data encoding system	: PPM
Stop bits	: Yes

   • Receive format C

Guide pulse	: No
Data encoding system	: PPM
Stop bits	: Yes

   • Receive format D

Guide pulse	: No
Data encoding system	: Manchester coding
Stop bits	: No

   • Receive format E

Guide pulse	: Clock
Data encoding system	: Manchester coding
Stop bits	: No

3) X'tal HOLD mode release function

3.20.2 Functions

1) Remote control receive function
   The REMOREC2 tests the pulses of the remote control signal input from the P73/RMIN pin using the clock output from the prescaler (RM2CKPR) which counts 1 to 128 Tcyc or subclock oscillation source (the RM2CK reference clock is selected out of 8 sources) to identify the data as 0, 1, or error. The data that is found normal is stored in the remote control receive shift register (RM2SFT). Every time 8 bits of data are stored in the register, the 8 bits are transferred to the remote control receive data register (RM2RDT). At this moment, the data transfer flag is set. The end of reception flag is set when the end of receive format condition is detected.

2) Interrupt generation
   An interrupt request to vector address 0013H is generated when an interrupt request occurs in the remote control receiver circuit provided that the interrupt request enable bit is set. The remote control receiver circuit can generate the following four types of interrupt requests:

(1) Guide pulse detection

(2) Receive data test error

(3) Data transfer from RM2SFT to RM2RDT

(4) End of reception

3) X'tal HOLD mode operation and X'tal HOLD mode release function
    The remote control receiver circuit is enabled for operation by setting bits 2 and 1 of the power control register (PCON) after the circuit is started for receive operation with RM2CK being selected as the subclock oscillation source.

    X'tal HOLD mode can be released by making use of the interrupt from the remote control receiver circuit. This function makes it possible to realize low power intermittent current operation.

4) It is necessary to manipulate the following special function registers to control the infrared remote control receiver circuit 2 (REMOREC2).
    • RM2CNT, RM2INT, RM2SFT, RM2RDT, RM2CTPR,

      RM2GPW, RM2DT0W, RM2DT1W, RM2XHW, P7

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE27	0000 0000	R/W	RM2CNT	RM2RUN	RM2FMT2	RM2FMT1	RM2FMT0	RM2DINV	RM2CK2	RM2CK1	RM2CK0
FE28	0000 0000	R/W	RM2INT	RM2GPOK	RM2GPIE	RM2DERR	RM2ERIE	RM2SFUL	RM2SFIE	RM2REND	RM2ENIE
FE29	0000 0000	R	RM2SFT	RM2SFT7	RM2SFT6	RM2SFT5	RM2SFT4	RM2SFT3	RM2SFT2	RM2SFT1	RM2SFT0
FE2A	XXXX XXXX	R	RM2RDT	RM2RDT7	RM2RDT6	RM2RDT5	RM2RDT4	RM2RDT3	RM2RDT2	RM2RDT1	RM2RDT0
FE2B	0000 0000	R/W	RM2CTPR	RM2GPR1	RM2GPR0	RM2DPR1	RM2DPR0	RM2HOLD	RM2BCT2	RM2BCT1	RM2BCT0
FE2C	0000 0000	R/W	RM2GPW	RM2GPH3	RM2GPH2	RM2GPH1	RM2GPH0	RM2GPL3	RM2GPL2	RM2GPL1	RM2GPL0
FE2D	0000 0000	R/W	RM2DT0W	RM2D0H3	RM2D0H2	RM2D0H1	RM2D0H0	RM2D0L3	RM2D0L2	RM2D0L1	RM2D0L0
FE2E	0000 0000	R/W	RM2DT1W	RM2D1H3	RM2D1H2	RM2D1H1	RM2D1H0	RM2D1L3	RM2D1L2	RM2D1L1	RM2D1L0
FE2F	0H00 0000	R/W	RM2XHW	RM2RDIR	-	RM2D1H4	RM2D1L4	RM2D0H4	RM2D0L4	RM2GPH4	RM2GPL4

3.20.3 Circuit Configuration

3.20.3.1 Remote control receive control register (RM2CNT)   (8-bit register)

1) This register controls the remote control receive operation.

3.20.3.2 Remote control receive interrupt control register (RM2INT)   (8-bit register)

1) This register controls the processing of remote control receive interrupts.

2) When the REMOREC2 starts receive operation with RM2CK selected as the subclock oscillation source, X'tal HOLD mode of the microcontroller can be released using the interrupt occurring in the REMOREC2 circuit.

3.20.3.3 Remote control receive shift register (RM2SFT)   (8-bit shift register)

1) This register is an 8-bit shift register used for storing remote control receive data.

2) The direction in which receive data is stored (LSB first or MSB first) is determined by the value of RM2RDIR (RM2XHW, bit 7).

3) Data is transferred from RM2SFT to RM2RDT each time this register is loaded with 8 bits of receive data. This register is also used to read the last less-than-8-bit receive data.

4) RM2SFT is reset when one of the following conditions occurs:

(1) The receive operation is stopped (RM2RUN = 0).

(2) A guide pulse is received normally after the beginning or resumption of a receive operation when RM2FMT2 to RM2FMT0 (RM2CNT, bits 6 to 4) are set to 0, 1, or 4.

(3) The first rising edge (assuming that the input polarity is set to positive phase) is detected after the beginning or resumption of a receive operation when RM2FMT2 to RM2FMT0 are set to 2, or 3.

(4) RM2SFT-to-RM2RDT data transfer occurs.

3.20.3.4   Remote control receive data register (RM2RDT)   (8-bit register)

1) This register is an 8-bit register that holds the data received from the remote control.

2) The initial value of this register is undefined. The contents of the RM2SFT are transferred to this register each time 8 bits of receive data are loaded in the RM2SFT.

3.20.3.5   Remote control receive bit counter & prescaler setup register (RM2CTPR)   (3-bit counter + 5-bit register)

1) This register consists of a 3-bit up-counter (RM2BCT) that counts the number of data bits received from the remote control, a flag (RM2HOLD) that signals the suspension and resumption of the next receive operation, and the bits that defines the count value (RM2GPR1,0/RM2DPR1,0) of RM2CKPR in the guide pulse or data pulse receive mode.

2) The RM2BCT starts counting up when the remote control input signal is identified as 0 or 1. When the receive operation is completed, the number of last less-than-8-bit data bits can be obtained by reading the value of RM2BCT.

The RM2BCT is reset when one of the following conditions occurs.

(1) The receive operation is stopped (RM2RUN set to 0).

(2) A guide pulse is received normally after the beginning or resumption of a receive operation when RM2FMT2 to RM2FMT0 are set to 0, 1, or 4.

(3) The first rising edge (assuming that the input polarity is set to positive phase) is detected after the beginning or resumption of a receive operation when RM2FMT2 to RM2FMT0 are set to 2 or 3.

3) The value of RM2GPR1 and RM2GPR0 exerts no influence on the receive operation if RM2FMT2 to RTM2FMT0 are set to 2 or 3.

3.20.3.6   Remote control receive prescaler (RM2CKPR)   (5-bit counter)

1) This prescaler is a 5-bit up-counter that generates a count clock to the pulse width measuring counter (RM2MJCT).

2) The counter counts up on the RM2CK that is selected by the value of RM2CK2 to RM2CK0 (RM2CNT, bits 2 to 0).

3) The RM2CKPR uses different count setup registers when receiving the guide pulse and the data pulse. The count is set up by RM2GPR1 and RM2GPR0 (RM2CTPR, bits 7 and 6) or RM2DPR1 and RM2DPR0 (RM2CTPR, bits 5 and 4).

A count clock to RM2MJCT is generated every one of the counts listed below.

* Count clock to the RM2MJCT in the guide pulse or data pulse receive mode

When "RM2FMT2 to RM2FMT0 = 0 to 2" is selected.

RM2GPR1 /RM2DPR1	RM2GPR0 /RM2DPR0	RM2CKPR Count Value
0	0	4
0	1	8
1	0	16
1	1	32

When "RM2FMT2 to RM2FMT0 = 3 or 4" is selected.

RM2GPR1 /RM2DPR1	RM2GPR0 /RM2DPR0	RM2CKPR Count Value
0	0	2
0	1	4
1	0	8
1	1	16

3.20.3.7 Remote control receive guide pulse width setup register (RM2GPW)   (8-bit register)
1) This register is an 8-bit register that defines the width of the guide pulse.
2) The value of this register exerts no influence on the receive operation when RM2FMT2 to RM2FMT0 are set to give a value of 2 or 3.

3.20.3.8 Remote control receive data 0 pulse width setup register (RM2DT0W)   (8-bit register)
1) This register is an 8-bit register that defines the width of the data 0 pulse and timings 1 and 2.

3.20.3.9 Remote control receive data 1 pulse width setup register (RM2DT1W)   (8-bit register)
1) This register is an 8-bit register that defines the width of the data 1 pulse and timings 3 and 4.

3.20.3.10 Remote control receive guide pulse & data pulse width high byte setup register
(RM2XHW)   (7-bit register)
1) This register is a 7-bit register that defines the width of the guide pulse and data pulse and sets the highest bit of timings 1 to 4. It is also used to control the direction in which data is loaded in RM2SFT.

3.20.3.11 Remote control receive pulse width measurement counter (RM2MJCT)   (5-bit counter)
1) This counter is a 5-bit up-counter used to measure the pulse width of the remote control input signal and to generate timing signals.
2) It counts up on the count clock output from the RM2CKPR.

Note: See the subsection 3.20.5 entitled "Remote Control Receiver Circuit Operation" for the operation of the REMOREC2 in various receive format mode.

3.20.3.12 Remote control receive noise filter (RM2NFLT)

1) This noise filter rejects remote control input signals whose width is less than a predetermined duration as noises.

2) When the REMOREC2 is running (RM2RUN set to 1), the remote control input signal is always sampled at RM2CK. The input signal is processed by the circuit as a valid signal if its signal levels remain the same while four samples are obtained. If the input signal width is less than RM2CK×4, the remote control input signal is rejected as noise and the REMOREC2 continues operation while preserving the state of the old signal in the circuit.

* Noise cancellation width

Less than RM2CK×4

Note: The noise cancellation width may vary by a maximum factor of ±RM2CK×1 depending on the timing at which the remote control input signal is sampled in the circuit.
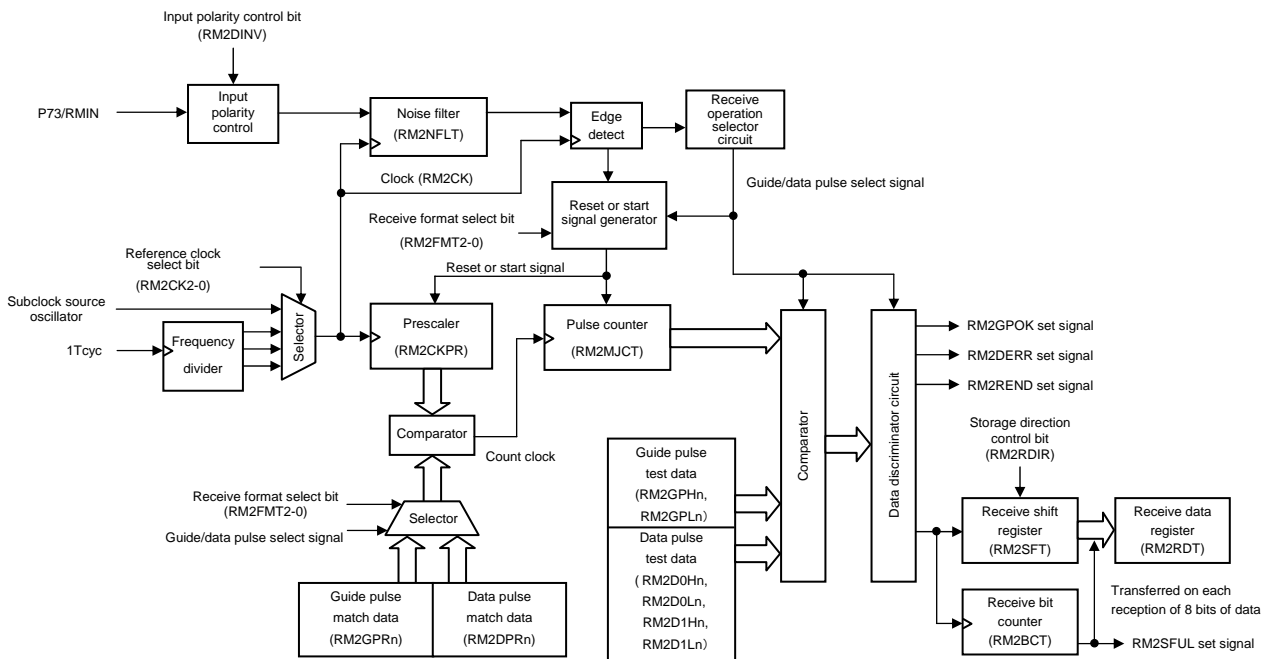
Figure 3.20.1   Infrared Remote Control Receiver Circuit 2 Operation Block Diagram
(RM2FMT2 - 0 = 0 - 2)

3.20.4　Related Registers

3.20.4.1　Remote control receive control register (RM2CNT)

1)　This register is an 8-bit register that controls the operation of the remote control receiver circuit.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE27	0000 0000	R/W	RM2CNT	RM2RUN	RM2FMT2	RM2FMT1	RM2FMT0	RM2DINV	RM2CK2	RM2CK1	RM2CK0

RM2RUN (bit 7): REMOREC2 receive control

Setting this bit to 0 stops the operation of the remote control receiver circuit.

When this bit is set to 1, the remote control receiver circuit starts operation and waits for the remote control input signal.

RM2FMT2 (bit 6):

RM2FMT1 (bit 5): REMOREC2 receive format select

RM2FMT0 (bit 4):

RM2FMT2	RM2FMT1	RM2FMT0	Format
0	0	0	Receive format A ・Guide pulse: Half clock ・Data encoding system: PPM ・Stop bits: No
0	0	1	Receive format B ・Guide pulse: Clock ・Data encoding system: PPM ・Stop bits: Yes
0	1	0	Receive format C ・Guide pulse: No ・Data encoding system: PPM ・Stop bits: Yes
0	1	1	Receive format D ・Guide pulse: No ・Data encoding system: Manchester coding ・Stop bits: No
1	0	0	Receive format E ・Guide pulse: Clock ・Data encoding system: Manchester coding ・Stop bits: No

*　Any value other than those listed above is inhibited.

*　See the subsection 3.20.5 entitled "Remote Control Receiver Circuit Operation" for the operation of the REMOREC2 in various receive format modes.

RM2DINV (bit 3): REMOREC2 receive input polarity control

This bit must be set to 0 when the remote control input signal is a positive phase signal.
This bit must be set to 1 when the input signal is a negative phase signal.

*　The REMOREC2 starts receive processing assuming the detection of a start edge immediately when it is activated if the positive phase input mode is specified for the high level of the remote control input signal or if the negative phase input mode is specified for the low level of the remote control input signal.

RM2CK2 (bit 2):

RM2CK1 (bit 1): **REMOREC2 receive reference clock (RM2CK) select**

RM2CK0 (bit 0):

RM2CK2	RM2CK1	RM2CK0	Reference Clock（RM2CK)
0	0	0	4 Tcyc
0	0	1	8 Tcyc
0	1	0	16 Tcyc
0	1	1	32 Tcyc
1	0	0	64 Tcyc
1	0	1	128 Tcyc
1	1	0	Subclock source oscillation
1	1	1	1 Tcyc

Notes:

• *The registers in the remote control receiver circuit must be set up when RM2RUN is set to 0 (receive operation stopped).*

• *When releasing X'tal HOLD mode, set the RM2CK to subclock source oscillation. The REMOREC2 will not run with any other RM2CK setting mode since the cycle clock is stopped in X'tal HOLD mode.*

3.20.4.2 Remote control receive interrupt control register (RM2INT)

1) This register is an 8-bit register that controls the handling of interrupts occurring in the remote control receiver circuit.

2) This register allows X'tal HOLD mode to be released by an interrupt occurring in the remote control receiver circuit provided that the REMOREC2 is started for receive processing with the RM2CK set to subclock source oscillation.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE28	0000 0000	R/W	RM2INT	RM2GPOK	RM2GPIE	RM2DERR	RM2ERIE	RM2SFUL	RM2SFIE	RM2REND	RM2ENIE

RM2GPOK (bit 7): Guide pulse receive flag

This bit is set when the REMOREC2 receives a guide pulse normally in a receive format that is specified by setting RM2FMT2 to RM2FMT0 to 0, 1, or 4.

This flag must be cleared with an instruction.

RM2GPIE (bit 6): Guide pulse receive interrupt request enable control

When this bit and RM2GPOK are set to 1, a X'tal HOLD mode release signal and an interrupt request to vector address 0013H are generated.

RM2DERR(bit 5): Receive data error flag

This bit is set when an error is detected while testing the received data.
This flag must be cleared with an instruction.

RM2ERIE (bit 4): Receive data error interrupt request enable control

When this bit and RM2DERR are set to 1, a X'tal HOLD mode release signal and an interrupt request to vector address 0013H are generated.

RM2SFUL (bit 3): Receive shift register FULL flag

> This bit is set when the 8 data bits loaded in RM2SFT are transferred from RM2SFT to RM2RDT.
> This flag must be cleared with an instruction.

RM2SFIE (bit 2): Receive shift register FULL interrupt request enable control

> When this bit and RM2SFUL are set to 1, a X'tal HOLD mode release signal and an interrupt request
> to vector address 0013H are generated.

RM2REND (bit 1): End of reception flag

> This bit is set when the end of the receive format conditions are detected.
> This flag must be cleared with an instruction.

RM2ENIE (bit 0): End of reception interrupt request enable control

> When this bit and RM2REND are set to 1, a X'tal HOLD mode release signal and an interrupt request
> to vector address 0013H are generated.

Notes:

 • *RM2GPOK is not set when RM2FMT2 to RM2FMT0 are set to give a value of 2 or 3.*

3.20.4.3 Remote control receive shift register (RM2SFT)

1) This register is an 8-bit shift register used to receive data from the remote control.

2) The data loading direction (LSB first or MSB first) is determined by the value of RM2RDIR.

3) Since the contents of this register are transferred to RM2RDT from RM2SFT each time 8 bits of receive data are loaded in the RM2SFT, this register is also used to read the last less-than-8-bit receive data.

4) RM2SFT is reset when one of the following conditions occurs:

   (1) The receive operation is stopped (RM2RUN set to 0).

   (2) A guide pulse is received normally after the beginning or resumption of a receive operation when RM2FMT2 to RM2FMT0 are set to 0, 1, or 4.

   (3) The first rising edge (assuming that the input polarity is set to positive phase) is detected after the beginning or resumption of a receive operation when RM2FMT2 to RM2FMT0 are set to 2, or 3.

   (4) Data transfer from RM2SFT to RM2RDT occurs.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE29	0000 0000	R	RM2SFT	RM2SFT7	RM2SFT6	RM2SFT5	RM2SFT4	RM2SFT3	RM2SFT2	RM2SFT1	RM2SFT0

Note:

 • *Before reading this register, make sure that the value of RM2REND is set to 1 (End of reception).*

3.20.4.4 Remote control receive data register (RM2RDT)

1) This register is an 8-bit register that holds the data received from the remote control.

2) The initial value of this register is undefined. Each received data block of 8 bits is transferred from RM2SFT to RM2RDT.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE2A	XXXX XXXX	R	RM2RDT	RM2RDT7	RM2RDT6	RM2RDT5	RM2RDT4	RM2RDT3	RM2RDT2	RM2RDT1	RM2RDT0

Note:

 • *Before reading this register, make sure that the value of RM2SFUL is set to 1 (Data transfer detected).*

3.20.4.5 Remote control receive bit counter & prescaler setup register (RM2CTPR)

1) This register consists of a 3-bit up-counter (RM2BCT) that counts the number of data bits received from the remote control, a flag (RM2HOLD) that signals the suspension and resumption of the next receive operation, and the bits that define the count value (RM2GPR1,0/RM2DPR1,0) of RM2CKPR in the guide pulse or data pulse receive mode.

2) The RM2BCT starts counting up when the remote control input signal is identified as 0 or 1. When the receive operation is completed, the number of last less-than-8-bit data bits can be obtained by reading the value of RM2BCT.

   The RM2BCT is reset when one of the following conditions occurs.

   (1) The remote control receive operation is stopped (RM2RUN = 0).

   (2) A guide pulse is received normally after the beginning or resumption of a receive operation when RM2FMT2 to RM2FMT0 are set to give a value of 0, 1, or 4.

   (3) The first rising edge (assuming that the input polarity is set to positive phase) is detected after the beginning or resumption of a receive operation when RM2FMT2 to RM2FMT0 are set to 2 or 3.

3) Bits 3 to 0 of this register is read only.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE2B	0000 0000	R/W	RM2CTPR	RM2GPR1	RM2GPR0	RM2DPR1	RM2DPR0	RM2HOLD	RM2BCT2	RM2BCT1	RM2BCT0

RM2GPR1 (bit 7):
      Guide pulse receive mode RM2CKPR count select
RM2GPR0 (bit 6):

RM2DPR1 (bit 5):
      Data pulse receive mode RM2CKPR count select
RM2DPR0 (bit 4):

When "RM2FMT2 to RM2FMT0 = 0 to 2" is selected.

RM2GPR1 /RM2DPR1	RM2GPR0 /RM2DPR0	RM2CKPR Count Value
0	0	4
0	1	8
1	0	16
1	1	32

When "RM2FMT2 to RM2FMT0 = 3 or 4" is selected.

RM2GPR1 /RM2DPR1	RM2GPR0 /RM2DPR0	RM2CKPR Count Value
0	0	2
0	1	4
1	0	8
1	1	16

RM2HOLD (bit 3): Receive operation suspend/resume flag

This bit is set and the REMOREC2 suspends the receive operation at the end of a receive operation. At this time, the REMOREC2 does not perform another receive operation even when a next remote control signal is input.

This bit is cleared and the REMOREC2 resumes the receive operation when the RM2SFT is read. This bit is also cleared when the receive operation is stopped (RM2RUN set to 0).

RM2BCT2 (bit 2):

RM2BCT1 (bit 1): Receive data counter

RM2BCT0 (bit 0):

> The REMOREC2 allows the number of last less-than-8-bit data block to be read at the end of a receive operation. From this value, the user can identify the number of valid received data bits that are left in the RM2SFT.

Note:

・ *The value that is set in RM2GPR1 and RM2GPR0 exerts no influence on the receive operation when RM2FMT2 to RM2FMT0 are set to give a value of 2 or 3.*

3.20.4.6 Remote control receive guide pulse width setup register (RM2GPW)

1) This register is an 8-bit register that defines the width of the guide pulse.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE2C	0000 0000	R/W	RM2GPW	RM2GPH3	RM2GPH2	RM2GPH1	RM2GPH0	RM2GPL3	RM2GPL2	RM2GPL1	RM2GPL0

Note:

・ *The value of this register exerts no influence on the receive operation when RM2FMT2 to RM2FMT0 are set to give a value of 2 or 3.*

3.20.4.7 Remote control receive data 0 pulse width setup register (RM2DT0W)

1) This register is an 8-bit register that defines the width of the data 0 pulse or timings 1 and 2.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE2D	0000 0000	R/W	RM2DT0W	RM2D0H3	RM2D0H2	RM2D0H1	RM2D0H0	RM2D0L3	RM2D0L2	RM2D0L1	RM2D0L0

3.20.4.8 Remote control receive data 1 pulse width setup register (RM2DT1W)

1) This register is an 8-bit register that defines the width of the data 1 pulse or timings 3 and 4.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE2E	0000 0000	R/W	RM2DT1W	RM2D1H3	RM2D1H2	RM2D1H1	RM2D1H0	RM2D1L3	RM2D1L2	RM2D1L1	RM2D1L0

3.20.4.9 Remote control receive guide pulse & data pulse width high byte setup register (RM2XHW)

1) This register is a 7-bit register that defines the width of the guide pulse and data pulse or sets the highest bit of timings 1 to 4. It is also used to control the direction in which data is loaded in RM2SFT.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE2F	0H00 0000	R/W	RM2XHW	RM2RDIR	-	RM2D1H4	RM2D1L4	RM2D0H4	RM2D0L4	RM2GPH4	RM2GPL4

RM2RDIR (bit 7): Remote control receive shift register loading data direction control

> When this bit is set to 0, the data received by the remote control is loaded into the RM2SFT on an LSB first basis.
> When this bit is set to 1, the data received by the remote control is loaded into the RM2SFT on an MSB first basis.

RM2D1H4 to RM2D1H0 (RM2XHW, bit 5 and RM2DT1W, bits 7 to 4)

> These bits are used to define the higher side of the data 1 pulse width or to generate timing 4.

REMOREC2

RM2D1L4 to RM2D1L0 (RM2XHW, bit 4 and RM2DT1W, bits 3 to 0)

These bits are used to define the lower side of the data 1 pulse width or to generate timing 3.

RM2D0H4 to RM2D0H0 (RM2XHW, bit 3 and RM2DT0W, bits 7 to 4)

These bits are used to define the higher side of the data 0 pulse width or to generate timing 2.

RM2D0L4 to RM2D0L0 (RM2XHW, bit 2 and RM2DT0W, bits 3 to 0)

These bits are used to define the lower side of the data 0 pulse width or to generate timing 1.

RM2GPH4 to RM2GPH0 (RM2XHW, bit 1 and RM2GPW, bits 7 to 4)

These bits are used to define the higher side of the guide pulse width.

RM2GPL4 to RM2GPL0 (RM2XHW, bit 0 and RM2GPW, bits 3 to 0)

These bits are used to define the lower side of the guide pulse width.

Note:

・ *See the subsection 3.20.5 entitled "Remote Control Receiver Circuit Operation" for the operation of the REMOREC2 in various receive format modes.*

3.20.5 Remote Control Receiver Circuit Operation

3.20.5.1 Receive operation when "receive format A" is specified

• Receive format A outline

Guide pulse : Half clock
Data encoding system : PPM
Stop bits : No

* Example of receive format A receive operation (positive phase input)
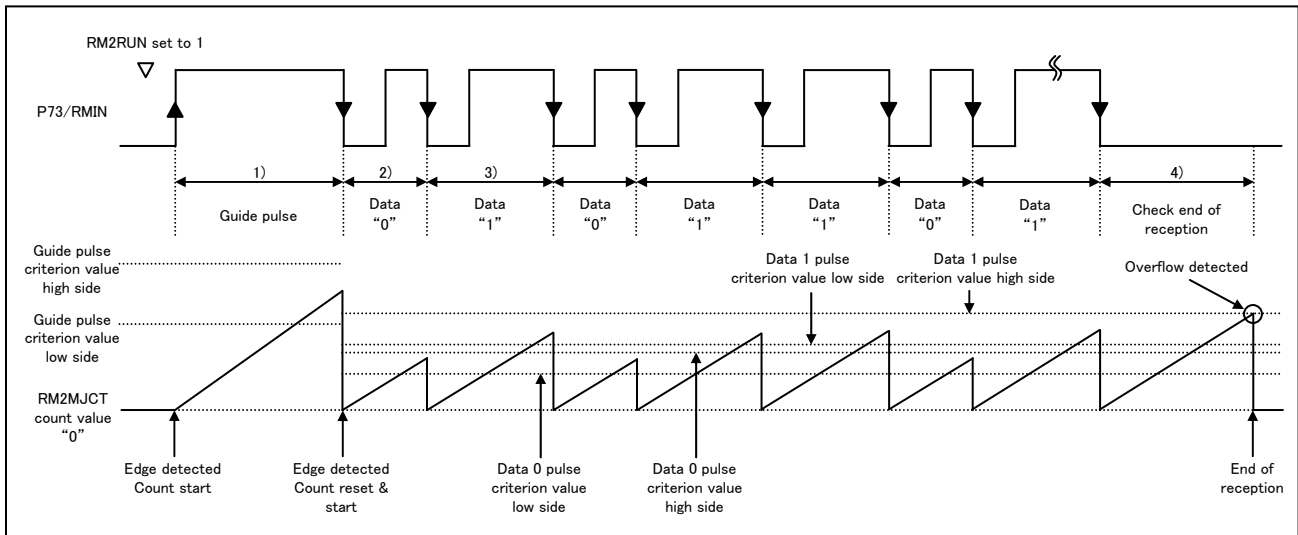
* Setting up the receive format A criterion values

1) Check the pulse width (from rising edge to falling edge) of the guide pulse.

RM2CK in guide pulse receive mode =
(Period selected by RM2CK2 to RM2CK0) × (Count value selected by RM2GPR1, RM2GPR0)

Guide pulse criterion value =
(Value given by RM2GPL4 to RM2GPL0+1) × RM2CK or greater to (Value given by RM2GPH4 to RM2GPH0 + 1) × Less than RM2CK

Note: The register values must be such that value given by RM2GPL4 to RM2GPL0 < value given by RM2GPH4 to RM2GPH0.

2), 3) Check the pulse width (from falling edge to falling edge) of data 0 and 1

RM2CK in data pulse receive mode =
(Period selected by RM2CK2 to RM2CK0) × (Count value selected by RM2DPR1, RM2DPR0)

Data 0 criterion value =
(Value given by RM2D0L4 to RM2D0L0 + 1) × RM2CK or greater to (Value given by RM2D0H4 to RM2D0H0 + 1) × Less than RM2CK

Data 1 criterion value=
(Value give by RM2D1L4 to RM2D1L0 + 1) × RM2CK or greater to (Value given by RM2D1H4 to RM2D1H0 + 1) × Less than RM2CK

Note: The register values must be such that value given by RM2D0L4 to RM2D0L0 < value given by RM2D0H4 to RM2D0H0 ≦ value given by RM2D1L4 to RM2D1L0 < value given by RM2D1H4 to RM2D1H0.

4) Detect an end of reception condition (from falling edge to overflow of data 1 criterion value).

End of reception detection = (Value given by RM2D1H4 to RM2D1H0 + 1) × RM2CK or greater

Note: The minimum criterion value is RM2CK × 8. The interval between the low and high values of guide and data pulses must be set up at intervals of RM2CK × 8 or greater.

REMOREC2

* Receive format A receive operation

(1) The REMOREC2 remains idle in the wait state until it receives a guide pulse normally. When the guide pulse falls within the valid criterion value range, the REMOREC2 resets the RM2MJCT and sets the RM2GPOK flag, then starts checking for the next data pulse. At this time, RM2SFT and RM2BCT are reset.

(2) When the data pulse falls within the valid criterion value range, the REMOREC2 resets the RM2MJCT and loads the data (0/1) into the RM2SFT. The data from the RM2SFT is transferred to the RM2RDT every time the REMOREC2 receives 8 bits of data. At this moment, the REMOREC2 sets the RM2SFUL flag and resets the RM2SFT.

(3) If the data pulse goes out of the valid criterion value range, the REMOREC2 sets the RM2DERR flag and returns into the idle state, waiting for a guide pulse.

(4) The number of received data bits is counted by the RM2BCT. When receiving the number of data bits that is not an integral multiple of 8, the REMOREC2 references this value at the end of reception to determine the number of valid data bits in the RM2SFT.

(5) When the REMOREC2 detects the end of reception condition, it sets the RM2REND and RM2HOLD flags and suspends operation. Subsequently, when the RM2SFT is read, the REMOREC2 clears the RM2HOLD flag and enters the idle state, waiting for a guide pulse (resuming the receive operation).

3.20.5.2 Receive operation when "receive format B" is specified

• Receive format B outline

Guide pulse	: Clock
Data encoding system	: PPM
Stop bits	: Yes

* Example of a receive format B receive operation (positive phase input)
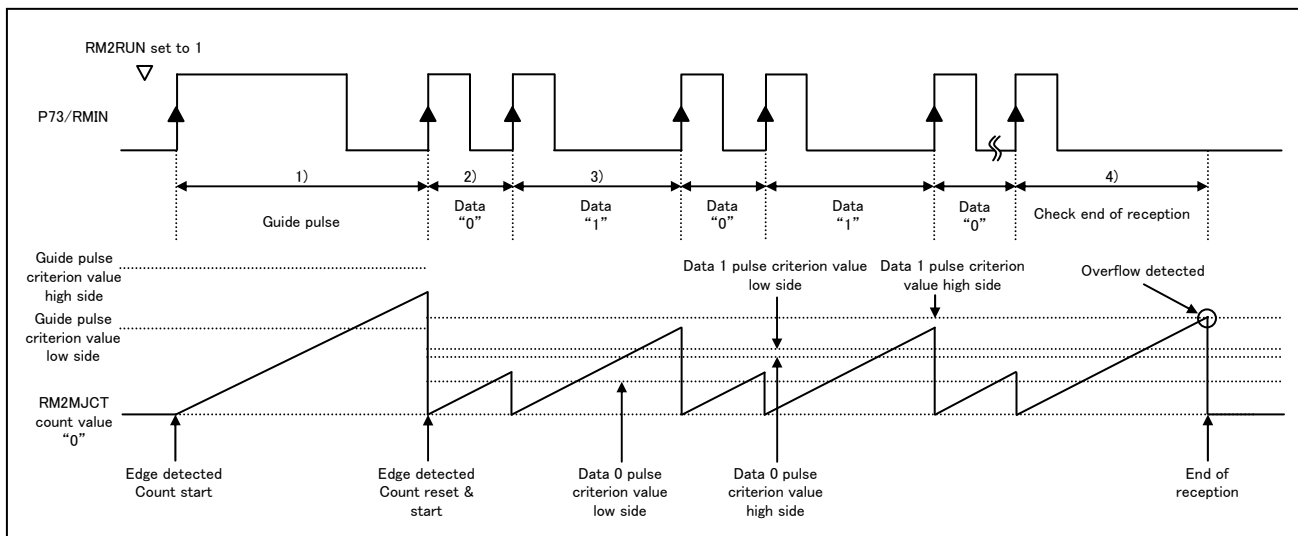
* Setting up the receive format B criterion values

1) Check the pulse width (from rising edge to rising edge) of the guide pulse.

2), 3) Check the pulse width (from rising edge to rising edge) of data 0 and 1

4) Detect an end of reception condition (from rising edge to overflow of data 1 criterion value).

The criterion values are the same as those for the receive format A.

* Receive format B receive operation

The REMOREC2 takes the same actions for receive format B as for receive format A. Refer to "Receive format A receive operation."

3.20.5.3 Receive operation when "receive format C" is specified

• Receive format C outline

Guide pulse : No
Data encoding system : PPM
Stop bits : Yes

* Example of a receive format C receive operation (positive phase input)
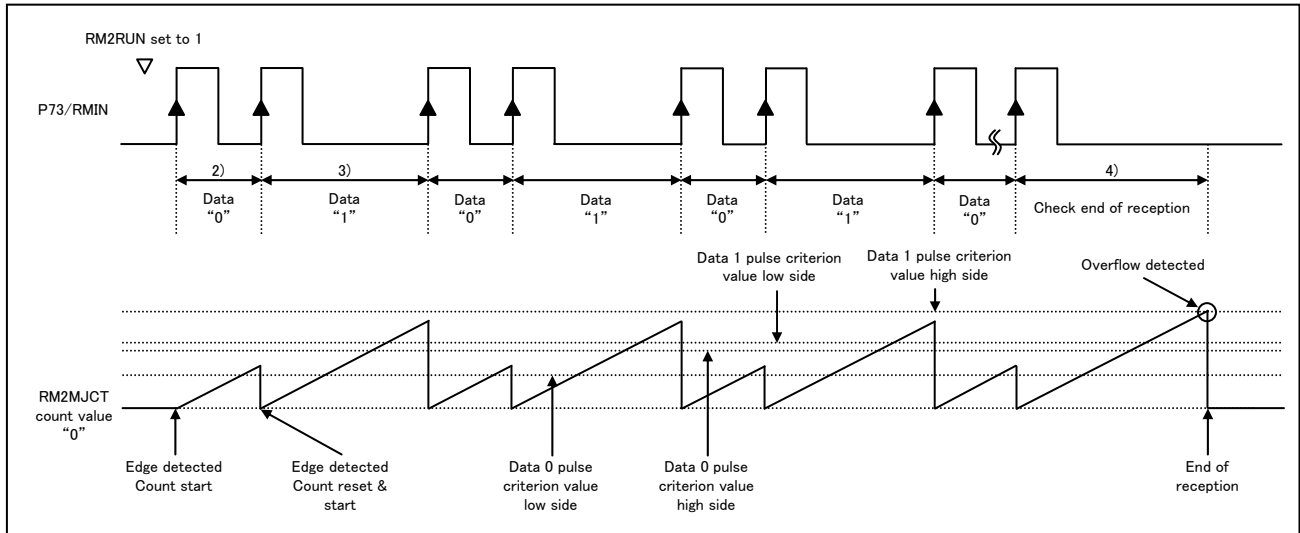
* Setting up the receive format C criterion values

2), 3) Check the pulse width (from rising edge to rising edge) of data 0 and 1

4) Detect an end of reception condition (from rising edge to overflow of data 1 criterion value).

The criterion values are the same as those for the receive format A.

* Receive format C receive operation

(1) When the REMOREC2 detects the first rising edge of the remote control signal at the beginning or resumption of a receive operation, it resets the RM2SFT and RM2BCT.

(2) When the data pulse falls within the valid criterion value range, the REMOREC2 resets the RM2MJCT and loads the data (0/1) into the RM2SFT. The data from the RM2SFT is transferred to the RM2RDT every time the REMOREC2 receives 8 bits of data. At this moment, the REMOREC2 sets the RM2SFUL flag and resets the RM2SFT.

(3) If the data pulse goes out of the valid criterion value range, the REMOREC2 sets the RM2DERR flag and returns into the idle state, waiting for a next rising edge.

(4) The number of received data bits is counted by the RM2BCT. When receiving the number of data bits that is not an integral multiple of 8, the REMOREC2 references this value at the end of reception to determine the number of valid data bits in the RM2SFT.

(5) When the REMOREC2 detects the end of reception condition, it sets the RM2REND and RM2HOLD flags and suspends operation. Subsequently, when the RM2SFT is read, the REMOREC2 clears the RM2HOLD flag and enters the idle state, waiting for a next rising edge (resuming the receive operation).

3.20.5.4 Receive operation when "receive format D" is specified

• Receive format D outline

Guide pulse : No
Data encoding system : Manchester
Stop bits : No

3-171

REMOREC2

* Example of a receive format D receive operation (positive phase input)
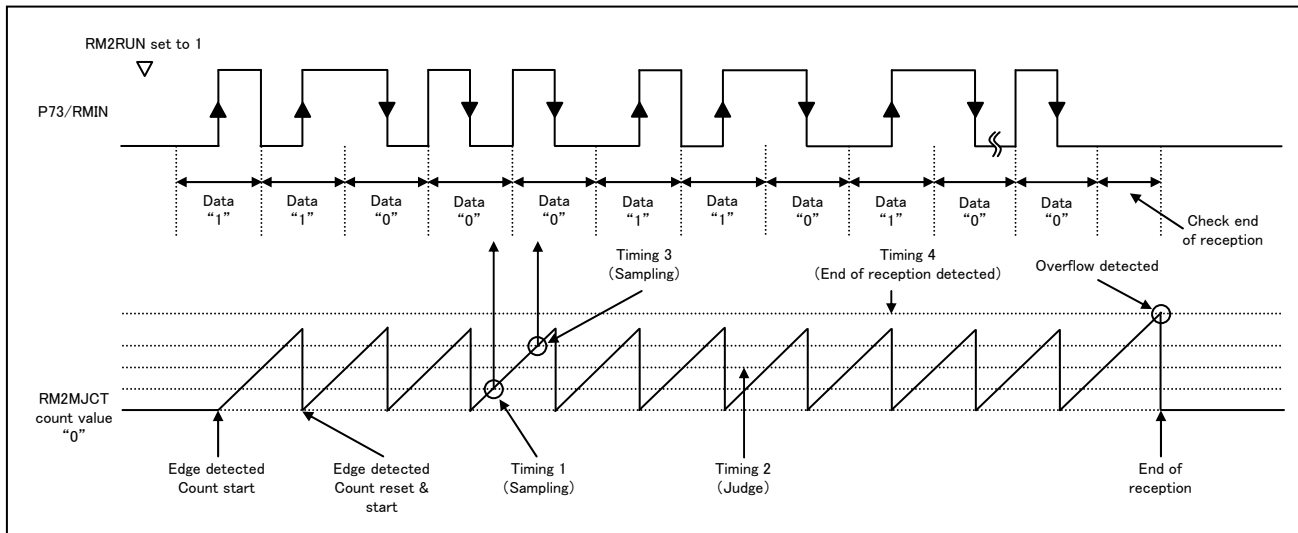
* Setting up the receive format D timings

The REMOREC2 generates four timing signals to check for the reception of a remote control signal.

Timing 1 (sampling) = (Value given by RM2D0L4 to RM2D0L0 + 1) × RM2CK

Timing 2 (data identification) = (Value given by RM2D0H4 to RM2D0H0 + 1) × RM2CK

Timing 3 (sampling) = (Value given by RM2D1L4 to RM2D1L0 + 1) × RM2CK

Timing 4 (detecting end of reception) = (Value given by RM2D1H4 to RM2D1H0 + 1) × RM2CK or greater

The remote control signal is sampled at timings 1 and 3. The resultant two data bits are tested for 0, 1, and error conditions.

Note: The register values must be such that value given by RM2D0L4 to RM2D0L0 < value given by RM2D0H4 to RM2D0H0 < value given by RM2D1L4 to RM2D1L0 < value given by RM2D1H4 to RM2D1H0.

Note: The minimum criterion value is RM2CK × 4. Each interval between timings 1 to 4 must be set up at intervals of RM2CK × 4 or greater.

* Receive format D receive operation

(1) When the REMOREC2 detects the first rising edge of the remote control signal at the beginning or resumption of a receive operation, it resets the RM2SFT and RM2BCT.

(2) At timing 1, the REMOREC2 samples the remote control signal.

(3) At timing 2, the REMOREC2 tests and identifies the data that are sampled in steps (2) and (6). When identifying the first data, the REMOREC2 identifies it as data 1 if an H is sampled at timing 1 (a data error is identified if an L is sampled).

(4) If the data is identified as 0 or 1, it (0/1) is loaded into the RM2SFT. The data from the RM2SFT is transferred to the RM2RDT every time the REMOREC2 receives 8 bits of data. At this moment, the REMOREC2 sets the RM2SFUL flag and resets the RM2SFT.

(5) If the data is identified as error, the REMOREC2 sets the RM2DERR flag and returns into the idle state, waiting for a next rising edge.

(6) At timing 3, the REMOREC2 samples the remote control signal. If the REMOREC2 detects an edge after starting the detection of an edge at this timing and before timing 4, it resets the RM2MJCT and returns to step (2).

(7) When the REMOREC2 detects the end of reception condition, it sets the RM2REND and RM2HOLD flags and suspends operation. Subsequently, when the RM2SFT is read, the REMOREC2 clears the RM2HOLD flag and enters the idle state, waiting for a next rising edge (resuming the receive operation).

3.20.5.5 Receive operation when "receive format E" is specified

• Receive format E outline

Guide pulse	: Yes
Data encoding system	: Manchester
Stop bits	: No

* Example of a receive format E receive operation (positive phase input)
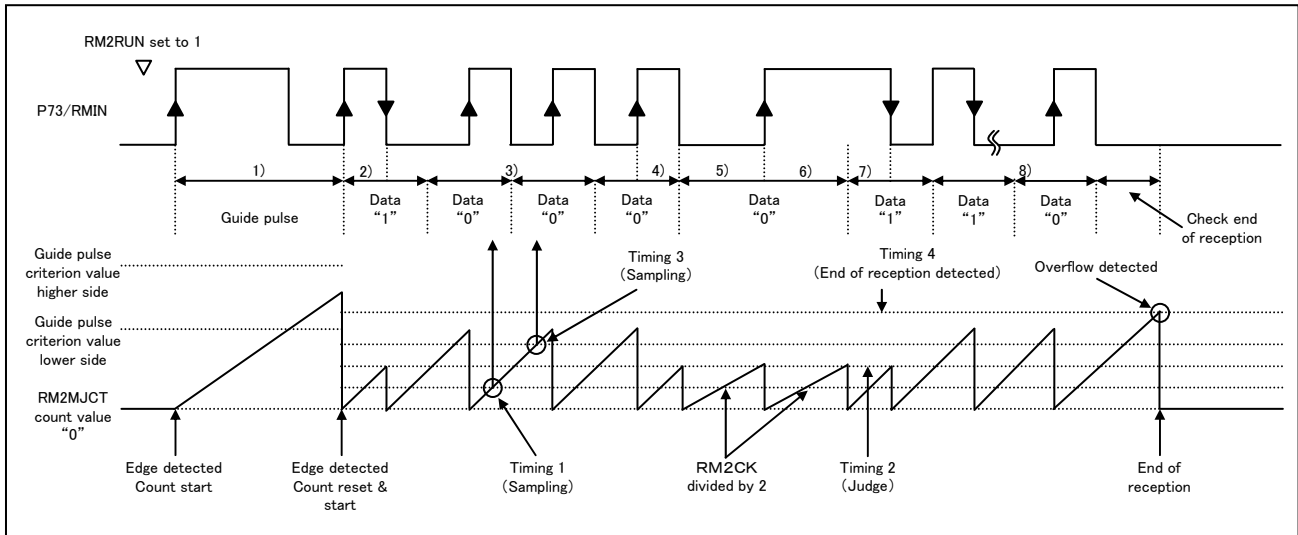
* Setting up the receive format E criterion values / timings

The procedure for setting up the guide pulse criterion values for receive format E is identical to that for receive format B.

The procedure for setting up the data pulse receive timings for receive format E is identical to that for receive format D.

Note: The minimum criterion value is RM2CK × 4. The interval between upper and lower guide pulse must be set up at intervals of RM2CK × 4 or greater.

* Receive format E receive operation

(1) The REMOREC2 remains in the idle state until it receives a guide pulse normally. When the guide pulse falls within the valid criterion value range, the REMOREC2 resets the RM2MJCT and sets the RM2GPOK flag, and tests the next data pulse. At this moment, the RM2SFT and RM2BCT are reset.

(2) At timing 1 in step 2), the REMOREC2 samples the remote control signal. If the REMOREC2 detects an edge after starting the detection of an edge at this timing and before timing 3, it resets the RM2MJCT and proceeds with the next step (a data error is identified if no edge is detected).

(3) At timing 1 in step 3) or 8), the REMOREC2 samples the remote control signal.

(4) At timing 2 in step 3) or 8), the REMOREC2 tests the data that is sampled in step (2) or (7), (3).

(5) If the data is identified as 0 or 1, it (0/1) is loaded into the RM2SFT. The data from the RM2SFT is transferred to the RM2RDT every time the REMOREC2 receives 8 bits of data. At this moment, the REMOREC2 sets the RM2SFUL flag and resets the RM2SFT.

(6) If the data is identified as error, the REMOREC2 sets the RM2DERR flag and returns into the idle state, waiting for a guide pulse.

(7) At timing 3 in step 3) or 8), the REMOREC2 samples the remote control signal. If the REMOREC2 detects an edge after starting the detection of an edge at this timing and before timing 4, it resets the RM2MJCT and returns to operation in step (3).

3-173

(8) When the REMOREC2 detects the end of reception condition, it sets the RM2REND and RM2HOLD flags and suspends operation. Subsequently, when the RM2SFT is read, the REMOREC2 clears the RM2HOLD flag and enters the idle state, waiting for a guide pulse (resuming the receive operation).

(9) After three cycles of steps (3) to (7), the REMOREC2 samples the remote control signal at timing 1 in step 4).

(10) At timing 2 in step 4), the REMOREC2 tests the data that is sampled in step (7) or (9). If the data is identified as 0 or 1, the REMOREC2 performs the same operation as step (5). It also resets the RM2MJCT and divides the frequency of RM2CK by 2. If the data is identified as error, the REMOREC2 performs the same operation as step (6).

(11) At timing1 in step 5), the REMOREC2 samples the remote control signal. If the REMOREC2 detects an edge after starting the detection of an edge at this timing and before timing 3, it resets the RM2MJCT and proceeds with the next step (a data error is identified if no edge is detected).

(12) At timing1 in step 6), the REMOREC2 samples the remote control signal.

(13) At timing 2 in step 6), the REMOREC2 tests the data that is sampled in step (11) or (12). If the data is identified as 0 or 1, the REMOREC2 performs the same operation as step (5). It also resets the RM2MJCT and resets RM2CK to the 1/1 frequency. If the data is identified as error, the REMOREC2 performs the same operation as step (6).

(14) At timing1 in step 7), the REMOREC2 samples the remote control signal. If the REMOREC2 detects an edge after starting the detection of an edge at this timing and before timing 3, it resets the RM2MJCT and proceeds with the next step (a data error is identified if no edge is detected).

(15) In subsequent step 8), the REMOREC2 repeats steps (3) to (7). It performs step (8) when it detects the end of reception condition.

3.21   32-bit Calculation Function

3.21.1   Overview

This series of microprocessors has four 32-bit arithmetic (addition, subtraction, multiplication, and division) functions.

3.21.2   Functions

1)   Operation Types

Operation Type	Results	Execution Time
Addition (32 bits + 32 bits + carry)	Sum: 32 bits	1 Tcyc
Subtraction (32 bits − 32 bits − borrow)	Difference: 32 bits	1 Tcyc
Multiplication (32 bits × 32 bits)	Product: 64 bits	11 Tcyc
Division (32 bits ÷ 32 bits)	Quotient: 32 bits Remainder: 32 bits	11 Tcyc (1 Tcyc if divisor is 0.)

2)   To control the 32-bit calculation function, it is necessary to manipulate the following special function registers.

   • DWC2CT, DWC2A0, DWC2A1, DWC2A2, DWC2A3, DWC2B0, DWC2B1, DWC2B2, WC2B3, DWC2C0, DWC2C1, DWC2C2, DWC2C3

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEE1	HHH0 0000	R/W	DWC2CT	-	-	-	DWCRUN	DWCOV	DWCCY	DWCMD1	DWCMD0
FEE2	0000 0000	R/W	DWC2A0	DWA07	DWA06	DWA05	DWA04	DWA03	DWA02	DWA01	DWA00
FEE3	0000 0000	R/W	DWC2A1	DWA15	DWA14	DWA13	DWA12	DWA11	DWA10	DWA09	DWA08
FEE4	0000 0000	R/W	DWC2A2	DWA23	DWA22	DWA21	DWA20	DWA19	DWA18	DWA17	DWA16
FEE5	0000 0000	R/W	DWC2A3	DWA31	DWA30	DWA29	DWA28	DWA27	DWA26	DWA25	DWA24
FEA0	0000 0000	R/W	DWC2B0	DWB07	DWB06	DWB05	DWB04	DWB03	DWB02	DWB01	DWB00
FEA1	0000 0000	R/W	DWC2B1	DWB15	DWB14	DWB13	DWB12	DWB11	DWB10	DWB09	DWB08
FEA2	0000 0000	R/W	DWC2B2	DWB23	DWB22	DWB21	DWB20	DWB19	DWB18	DWB17	DWB16
FEA3	0000 0000	R/W	DWC2B3	DWB31	DWB30	DWB29	DWB28	DWB27	DWB26	DWB25	DWB24
FEA4	XXXX XXXX	R	DWC2C0	DWC07	DWC06	DWC05	DWC04	DWC03	DWC02	DWC01	DWC00
FEA5	XXXX XXXX	R	DWC2C1	DWC15	DWC14	DWC13	DWC12	DWC11	DWC10	DWC09	DWC08
FEA6	XXXX XXXX	R	DWC2C2	DWC23	DWC22	DWC21	DWC20	DWC19	DWC18	DWC17	DWC16
FEA7	XXXX XXXX	R	DWC2C3	DWC31	DWC30	DWC29	DWC28	DWC27	DWC26	DWC25	DWC24

3.21.3   Related Registers

3.21.3.1   Arithmetic control register   (DWC2CT)

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEE1	HHH0 0000	R/W	DWC2CT	-	-	-	DWCRUN	DWCOV	DWCCY	DWCMD1	DWCMD0

DWCRUN (bit 4): Arithmetic control flag

1)   Setting this bit to 1 starts the arithmetic operation.

2)   This bit is automatically cleared to 0 at the end of the operation.

3)   No correct arithmetic results can be obtained if this flag is cleared to 0 during the arithmetic operation.

DWCALC2

DWCOV (bit 3): Overflow flag

This bit is set to 1 when an overflow occurs as the result of the arithmetic operation and cleared to 0 when no overflow occurs. An overflow occurs when one of the following conditions occurs:

1) The result of a negative number + a negative number or a negative number– a positive number performed with MSB being regarded as the sign bit yields a positive number.

2) The result of a positive number + a positive number or a positive number– a negative number performed with MSB being regarded as the sign bit yields a negative number.

3) The higher-order 32 bits of the 64-bit multiplication results are nonzero.

4) The divisor of a division is 0.

DWCCY (bit 2): Carry flag

1) This flag bit is used as the carry in of additions or subtractions.

2) The bit is set to 1 when a carry or borrow occurs as the result of an addition or subtraction and cleared to 0 if neither carry nor borrow occurs.

DWCMD1 (bit 1): Operation type select

DWCMD0 (bit 0): Operation type select

DWCMD[1:0]	Operation Type
00	Addition
01	Subtraction
10	Multiplication
11	Division

3.21.3.2 The first operand register   (DWC2A)

1) This register holds a 32-bit operand (augend, minuend, multiplicand, or dividend).

2) In a multiplication, the register holds the lower-order 32 bits of the 64-bit product of the results of multiplication.

3) In a division, the register holds the 32-bit quotient of the results of division.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEE2	0000 0000	R/W	DWC2A0	DWA07	DWA06	DWA05	DWA04	DWA03	DWA02	DWA01	DWA00
FEE3	0000 0000	R/W	DWC2A1	DWA15	DWA14	DWA13	DWA12	DWA11	DWA10	DWA09	DWA08
FEE4	0000 0000	R/W	DWC2A2	DWA23	DWA22	DWA21	DWA20	DWA19	DWA18	DWA17	DWA16
FEE5	0000 0000	R/W	DWC2A3	DWA31	DWA30	DWA29	DWA28	DWA27	DWA26	DWA25	DWA24

3.21.3.3 The second operand register (DWC2B)

1) This register holds a 32-bit operand (addend, subtrahend, multiplier, or divisor).

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA0	0000 0000	R/W	DWC2B0	DWB07	DWB06	DWB05	DWB04	DWB03	DWB02	DWB01	DWB00
FEA1	0000 0000	R/W	DWC2B1	DWB15	DWB14	DWB13	DWB12	DWB11	DWB10	DWB09	DWB08
FEA2	0000 0000	R/W	DWC2B2	DWB23	DWB22	DWB21	DWB20	DWB19	DWB18	DWB17	DWB16
FEA3	0000 0000	R/W	DWC2B3	DWB31	DWB30	DWB29	DWB28	DWB27	DWB26	DWB25	DWB24

3.21.3.4 Arithmetic result register (DWC2C)

1) The arithmetic result register holds the 32-bit sum of an addition.

2) The arithmetic result register holds the 32-bit difference of a subtraction.

3) The arithmetic result register holds the higher-order 32 bits of the 64-bit product of a multiplication.

4) The arithmetic result register holds the 32-bit remainder of a division.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA4	XXXX XXXX	R	DWC2C0	DWC07	DWC06	DWC05	DWC04	DWC03	DWC02	DWC01	DWC00
FEA5	XXXX XXXX	R	DWC2C1	DWC15	DWC14	DWC13	DWC12	DWC11	DWC10	DWC09	DWC08
FEA6	XXXX XXXX	R	DWC2C2	DWC23	DWC22	DWC21	DWC20	DWC19	DWC18	DWC17	DWC16
FEA7	XXXX XXXX	R	DWC2C3	DWC31	DWC30	DWC29	DWC28	DWC27	DWC26	DWC25	DWC24

DWCALC2

4.   Control Functions

4.1   Interrupt Function

4.1.1   Overview

This series of microcontrollers has the capabilities to control three levels of multiple interrupts, i.e., low level (L), high level (H), and highest level (X). The master interrupt enable and interrupt priority control registers are used to enable or disable interrupts and determine the priority of interrupts.

4.1.2   Functions

1)   Interrupt processing

- Peripheral modules generate an interrupt request to the predetermined vector address when the interrupt request and interrupt request enable flags are set to 1.

- When the microcontroller receives an interrupt request from a peripheral module, it determines the interrupt level, priority and interrupt enable status of the interrupt. If the interrupt request is legitimate for processing, the microcontroller saves the value of PC in the stack and causes a branch to the predetermined vector address.

- The return from the interrupt routine is accomplished by the RETI instruction, which restores the old state of the PC and interrupt level.

2)   Multilevel interrupt control

- The interrupt function supports three levels of interrupts, that is, the low level (L), high level (H), and highest level (X). The interrupt function will not accept any interrupt request of the same level or lower level than that of the interrupt that is currently being processed.

3)   Interrupt priority

- When interrupt requests to two or more vector addresses occur at the same time, the interrupt request of the highest level takes precedence over the other interrupt requests. For interrupts of the same level, the interrupt into the smallest vector address is given priority.

4)   Interrupt request enable control

- The master interrupt enable register can be used to control the enabling/disabling of H- and L-level interrupt requests.

- Interrupt requests of the X level cannot be disabled.

5)   Interrupt disable period

- Interrupts are held disabled for a period of 2Tcyc after a write is made to the IE (FE08H), IP (FE09H) register, or HOLD mode is released.

- No interrupt can occur during the interval between the execution of an instruction that loads the PCON (FE07H) register and the execution of the next instruction.

- No interrupt can occur during the interval between the execution of a RETI instruction and the execution of the next instruction.

6) Interrupt level control
   • Interrupt levels can be selected on a vector address basis.

Table of Interrupts

No.	Vector	Selectable Level	Interrupt Sources
1	00003H	X or L	INT0
2	0000BH	X or L	INT1
3	00013H	H or L	INT2/T0L/INT4/USB bus active/Remote control receive
4	0001BH	H or L	INT3 /INT5/Base timer
5	00023H	H or L	T0H
6	0002BH	H or L	T1L/T1H
7	00033H	H or L	SIO0/USB bus reset/USB suspend/ UART1 receive/UART2 receive
8	0003BH	H or L	SIO1/USB endpoint/USB-SOF/SIO4/UART1 transmit/UART2 transmit
9	00043H	H or L	ADC/T6/T7
10	0004BH	H or L	Port 0/PWM0/PWM1

• Priority levels: X > H > L

• For interrupts of the same level, an interrupt with a smaller vector address is given priority.

7) To identify interrupt sources, to enable interrupts and to specify their priority, it is necessary to manipulate the following special function registers.
   • IE, IP

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE08	0000 HH00	R/W	IE	IE7	XFLG	HFLG	LFLG	-	-	XCNT1	XCNT0
FE09	0000 0000	R/W	IP	IP4B	IP43	IP3B	IP33	IP2B	IP23	IP1B	IP13

4.1.3   Circuit Configuration

4.1.3.1   Master interrupt enable control register (IE)   (6-bit register)
1) This register enables and disables H- and L-level interrupts.
2) The interrupt level flag of the register can be read.
3) The register selects the level (L or X) of interrupts to vector addresses 00003H and 0000BH.

4.1.3.2   Interrupt priority control register (IP)   (8-bit register)
1) This register selects the level (H or L) of interrupts to vector addresses 00013H to 0004BH.

4.1.4　Related Registers

4.1.4.1　Master interrupt enable control register (IE)

1)　This register is a 6-bit register for controlling the interrupts. Bits 6 to 4 are read only.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE08	0000 HH00	R/W	IE	IE7	XFLG	HFLG	LFLG	-	-	XCNT1	XCNT0

IE7 (bit 7): H-/L-level interrupt enable/disable control
- A 1 in this bit enables H- and L-level interrupt request to be accepted.
- A 0 in this bit disables H- and L-level interrupt request to be accepted.
- X-level interrupt requests are always enabled regardless of the state of this bit

XFLG (bit 6): X-level interrupt flag (R/O)
- This bit is set when an X-level interrupt is accepted and reset when execution returns from the processing of the X-level interrupt.
- This bit is read only. No instruction can rewrite the value of this bit directly.

HFLG (bit 5): H-level interrupt flag (R/O)
- This bit is set when an H-level interrupt is accepted and reset when execution returns from the processing of the H-level interrupt.
- This bit is read only. No instruction can rewrite the value of this bit directly.

LFLG (bit 4): L-level interrupt flag (R/O)
- This bit is set when an L-level interrupt is accepted and reset when execution returns from the processing of the L-level interrupt.
- This bit is read only. No instruction can rewrite the value of this bit directly.

(Bits 3, 2): These bits do not exist. They are always read as 1.

XCNT1 (bit 1): 0000BH interrupt level control flag
- A 1 in this bit sets all interrupts to vector address 0000BH to the L-level.
- A 0 in this bit sets all interrupts to vector address 0000BH to the X-level.

XCNT0 (bit 0): 00003H interrupt level control flag
- A 1 in this bit sets all interrupts to vector address 00003H to the L-level.
- A 0 in this bit sets all interrupts to vector address 00003H to the X-level.

4.1.4.2   Interrupt priority control register (IP)

1) This register is an 8-bit register that selects the interrupt level (H/L) of interrupts to vector addresses 00013H to 0004BH.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE09	0000 0000	R/W	IP	IP4B	IP43	IP3B	IP33	IP2B	IP23	IP1B	IP13

	Interrupt Vector Address	IP Bit	Value	Interrupt Level
7	0004BH	IP4B	0	L
			1	H
6	00043H	IP43	0	L
			1	H
5	0003BH	IP3B	0	L
			1	H
4	00033H	IP33	0	L
			1	H
3	0002BH	IP2B	0	L
			1	H
2	00023H	IP23	0	L
			1	H
1	0001BH	IP1B	0	L
			1	H
0	00013H	IP13	0	L
			1	H

4.2   System Clock Generator Function

4.2.1   Overview

This series of microcontrollers incorporates four systems of oscillator circuits, i.e., the main clock oscillator, subclock oscillator, RC oscillator, and USB-dedicated PLL oscillator, as system clock generator circuits. The RC oscillator circuit has internal resistors and capacitors, so that no external circuit is required.

The system clock can be selected from these four types of clock sources under program control.

4.2.2   Functions

1) System clock select

   • Allows the system clock to be selected under program control from four types of clocks generated by the main clock oscillator, subclock oscillator, RC oscillator, and USB-dedicated PLL oscillator.

2) System clock frequency division

   • Divides the frequency of the oscillator clock selected as the system clock and supplies the resultant clock to the system as the system clock.

   • The frequency divider circuit is made up of two stages:

   The first stage allows the selection of division ratios of $\frac{1}{1}$ and $\frac{1}{2}$.

   The second stage allows the selection of division ratios of $\frac{1}{1}$, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$, and $\frac{1}{128}$.

3) Oscillator circuit control

   • Allows the start/stop control of the four systems of oscillators to be executed independently through microcontroller instructions.

4) Multiplexed I/O pin function

   • The crystal oscillator pins XT1 and XT2 can also be used as input port and I/O port, respectively.

5) Oscillator circuit states and operating modes

Mode/Clock	Main Clock	Subclock	RC Oscillator	USB-dedicated PLL Oscillator	System Clock
Reset	Running	Stopped	Running	Running	RC oscillator
Normal mode	Programmable	Programmable	Programmable	Programmable	Programmable
HALT	State established at entry time	State established at entry time	State established at entry time	State established at entry time	State established at entry time
HOLD	Stopped	Stopped	Stopped	Stopped	Stopped
Immediately after release of HOLD mode	Running	State established at entry time	Running	State established at entry time	RC oscillator
X'tal HOLD	Stopped	State established at entry time	Stopped	Stopped	Stopped
Immediately after release of X'tal HOLD	State established at entry time	State established at entry time	State established at entry time	State established at entry time	State established at entry time

Note: See Section 4.3," Standby Function," for the procedures to enter and exit the microcontroller operating modes.

● Reset
・ Main clock started
・ Subclock stopped
・ RC oscillator started
・ USB-dedicated PLL oscillator
  started

● X'tal HOLD mode
・ Main clock, RC oscillator, and
  PLL oscillator are stopped.
  Subclock retains the state
  established when X'tal HOLD
  mode is entered.
・ Contents of OCR register
  unchanged
・ Contents of PLLCNT register
  unchanged
・ Contents of USBDIV register
  unchanged
・ CPU enters this mode after
  selecting subclock or RC
  oscillator as system clock
  source and stopping main clock.
・ When X'tal HOLD mode is exited,
  the oscillators return to the state
  established when the mode is
  entered.

● Normal operating mode
・ Start/stop of oscillators
  programmable

● HOLD mode
・ All oscillators stopped.
・ Since OCR register, bits 0, 1, 4, and
  5 are cleared, the main clock and
  RC oscillator are started and the RC
  oscillator is designated as system
  clock when HOLD mode is
  released.

● HALT mode
・ All oscillators retain the state
  established when HALT mode
  is entered.

6) To control the system clock, it is necessary to manipulate the following special function registers.
   • USBDIV, PCON, CLKDIV, PLLCNT, OCR

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE04	0000 0000	R/W	USBDIV	CF48ON	DVCKON	DVCKDR	P73NDL	CF12OFF	UDVSEL2	UDVSEL1	UDVSEL0
FE07	HHHH H000	R/W	PCON	-	-	-	-	-	XTIDLE	PDN	IDLE
FE0C	HHHH H000	R/W	CLKDIV	-	-	-	-	-	CLKDV2	CLKDV1	CLKDV0
FE0D	0000 0000	R/W	PLLCNT	SELREF2	SELREF1	SELREF0	PLLTEST	VCOSTP	CMPSTP	PLLCNT1	PLLCNT0
FE0E	0000 XX00	R/W	OCR	CLKSGL	EXTOSC	CLKCB5	CLKCB4	XT2IN	XT1IN	RCSTOP	CFSTOP
FE43	0000 0000	R/W	XT2PC	XT2PCB7	XT2PCB6	XT2PCB5	XT2PCB4	XT2PCB3	XT2PCB2	XT2DR	XT2DT

4.2.3　Circuit Configuration

4.2.3.1　Main clock oscillator circuit

1) The main clock oscillator circuit gets ready for oscillation by connecting a ceramic resonator and a capacitor to the CF1 and CF2 pins.

2) CF1 must be connected to VDD and CF2 must be released when the main clock is not to be used.

4.2.3.2　Subclock oscillator circuit

1) The subclock oscillator circuit gets ready for oscillation by connecting a crystal resonator (32.768 kHz standard), a capacitor, feedback resistor, and a damping resistor to the XT1 and XT2 pins.

2) The state of the XT1 and XT2 pins can be read as bits 2 and 3 of the register OCR.

3) When the subclock is not to be used, XT1 must be connected to VDD, XT2 must be released, and bit 6 of the OCR register must be set.

4.2.3.3　Internal RC oscillator circuit

1) The internal RC oscillator circuit oscillates according to the internal resistors and capacitors.

2) The clock from the RC oscillator circuit is selected as the system clock after the microcontroller is reset or exits HOLD mode.

3) Unlike main clock and subclock oscillators, the RC oscillator circuit starts oscillation from the beginning of oscillation at a normal frequency.

4.2.3.4   USB-dedicated internal PLL oscillator circuit

1) The USB-dedicated PLL oscillator circuit oscillates to drive the main clock by connecting a ceramic resonator and capacitor to the CF1 and CF2 pins.

2) An external circuit (see the data sheet) must be connected to the UFILT/P34 pin.

3) The 48 MHz clock for the USB is generated by the internal multiplier circuit using the main clock as the reference.

4) A 4 to 16 MHz clock which is derived by frequency-dividing the 48 MHz USB clock can be supplied as the system clock. However, it is necessary to configure the frequency of this clock to 8-16 MHz to drive the USB function control circuit.

4.2.3.5   Power control register (PCON)   (3-bit register)

1) This register specifies the operating mode (Normal/HALT/HOLD/X'tal HOLD).

4.2.3.6   Oscillation control register (OCR)   (8-bit register)

1) This register controls the start/stop operation of the oscillator circuits.

2) This register selects the system clock.

3) The register sets the division ratio of the oscillation clock to be used as the system clock to $\frac{1}{1}$ or $\frac{1}{2}$.

4) The state of the XT1 and XT2 pins can be read as bits 2 and 3 of this register.

4.2.3.7   XT2 general-purpose port output control register (XT2PC)   (8-bit register)

1) This register controls the general-purpose output (N-channel open drain type) at the XT2 pin.

4.2.3.8   USB-dedicated PLL oscillator circuit control register (PLLCNT)   (8-bit register)

1) This register controls the start/stop operation of the PLL oscillator circuit.

2) The register defines the frequency of the PLL reference clock (the resonator connected across the CF pins).

4.2.3.9   USB frequency divided clock control register (USBDIV)   (8-bit register)

1) This register is used to select the frequency (out of 4, 4.8, 6, 8, 12, and 16 MHz) that is obtained by dividing the 48 MHz clock for the USB.

2) It is also used to select either CF or USB frequency divided clock as the main clock.

4.2.3.10   System clock division control register (CLKDIV)   (3-bit register)

1) This register controls the operation of the system clock divider circuit.

The division ratios of $\frac{1}{1}$, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$, and $\frac{1}{128}$ are allowed.

System Clock
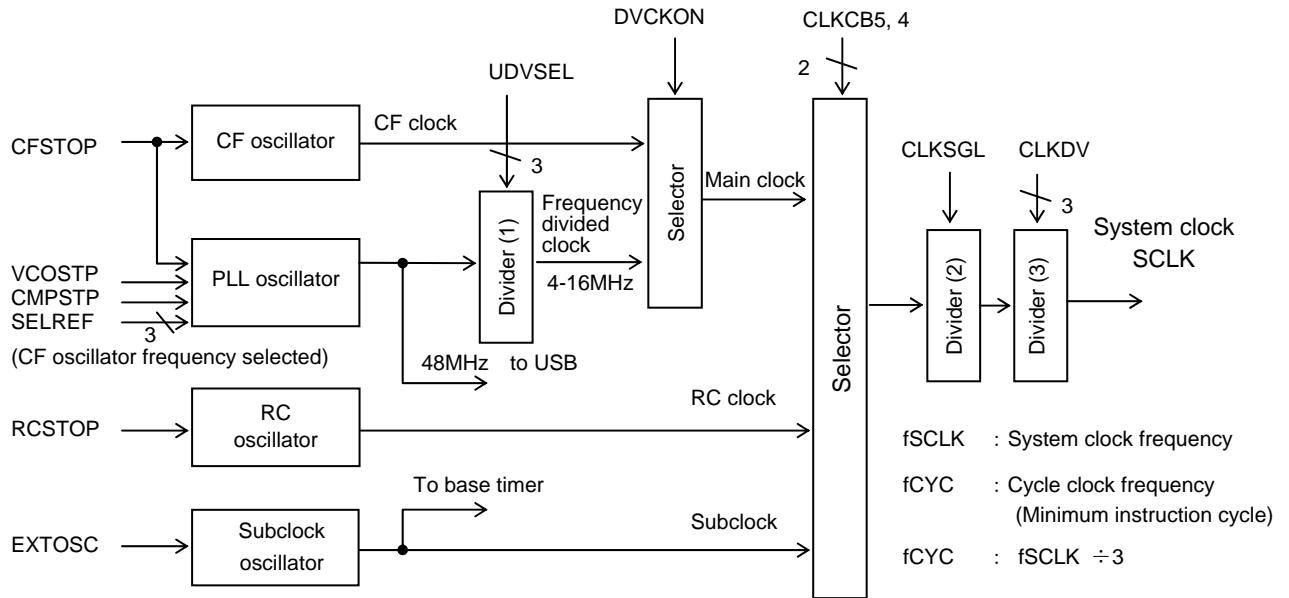
Fig. 4.2.1   System Clock Generator Block Diagram

4.2.4   Related Registers

4.2.4.1   Power control register (PCON)   (3-bit register)

1)   This register is a 3-bit register used to specify the operating mode (normal/HALT/ HOLD/X'tal HOLD).

   • See Section 4.3, "Standby Function," for the procedures to enter and exit the microcontroller operating modes.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE07	HHHH H000	R/W	PCON	-	-	-	-	-	XTIDLE	PDN	IDLE

(Bits 7 to 3): These bits do not exist. They are always read as 1.

XTIDLE (bit 2): X'tal HOLD mode setting flag

PDN (bit 1): HOLD mode setting flag

XTIDLE	PDN	Operating mode
—	0	Normal or HALT mode
0	1	HOLD mode
1	1	X'tal HOLD mode

1)   These bits must be set with an instruction.

   • If the microcontroller enters HOLD mode, all oscillations (main clock, subclock, RC, and PLL) are suspended and bits 0, 1, 4, and 5 of the OCR are set to 0.

   • When the microcontroller returns from HOLD mode, the main clock and RC oscillators resume oscillation. The subclock and PLL oscillators restore the state that is established before HOLD mode is entered and the system clock is set to RC.

   • When the microcontroller enters X'tal HOLD mode, all oscillations except XT (main clock, RC, and PLL) are suspended but the contents of the OCR register remain unchanged.

   • When the microcontroller returns from X'tal HOLD mode, the system clock to be used when X'tal HOLD mode is entered needs to be set to either subclock or RC because it is impossible to reserve the oscillation stabilization time for the main clock.

- Since X'tal HOLD mode is used usually for low-current clock counting, less current will be consumed if the system clock is switched to the subclock and the main clock and RC oscillations are suspended before X'tal HOLD mode is entered.

2) XTIDLE must be cleared with an instruction.

3) PDN is cleared when a HOLD mode release signal (INT0, INT1, INT2, INT4, INT5, P0INT, USB bus active or remote control receive) or a reset signal occurs.

4) Bit 0 is automatically set when PDN is set.

IDLE (bit 0): HALT mode setting flag

1) Setting this bit places the microcontroller into HALT mode.

2) This bit is automatically set when bit 1 is set.

3) This bit is cleared on acceptance of an interrupt request or on receipt of a reset signal.

4.2.4.2   Oscillation control register (OCR)   (8-bit register)

1) This register is an 8-bit register that controls the operation of the oscillator circuits, selects the system clock, and reads data from the XT1 and XT2 pins. Except for read-only bits 3 and 2, all bits of this register can be read or written.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE0E	0000 XX00	R/W	OCR	CLKSGL	EXTOSC	CLKCB5	CLKCB4	XT2IN	XT1IN	RCSTOP	CFSTOP

CLKSGL (bit 7): Clock division ratio select

1) When this bit is set to 1, the clock selected by bits 4 and 5 is used as the system clock as is.

2) When this bit is set to 0, the clock having a clock rate of $\frac{1}{2}$ of the clock selected by bits 4 and 5 is used as the system clock.

EXTOSC (bit 6): XT1/XT2 function control

1) When this bit is set to 1, the XT1 and XT 2 pins serve as the pins for subclock oscillation and get ready for oscillation when a crystal resonator (32.768 kHz standard), capacitors, feedback resistors, and damping resistors are connected. When the OCR register is read in this case, bit 3 reads the data at the XT2 pin and bit 2 reads 0.

2) When this bit is set to 0, the XT1 and XT2 pins serve as input pins. When the OCR register is read in this case, bit 3 reads the data at the XT2 pin and bit 2 reads the data at the XT1 pin.

CLKCB5 (bit 5): System clock select
CLKCB4 (bit 4): System clock select

1) CLKCB5 and CLKCB4 are used to select the system clock.

2) CLKCB5 and CLKCB4 are cleared at reset time or when HOLD mode is entered.

CLKCB5	CLKCB4	System Clock
0	0	Internal RC oscillator
0	1	Main clock
1	0	Subclock
1	1	Main clock

XT2IN (bit 3): XT2 data (read only)
XT1IN (bit 2): XT1 data (read only)

1) Data that can be read via XT1IN varies as summarized below according to the value of EXTOSC (bit 6).

EXTOSC	XT2IN	XT1IN
0	XT2 pin data	XT1 pin data
1	XT2 pin data	0 is read.

RCSTOP (bit 1): Internal RC oscillator control

1) Setting this bit to 1 stops the oscillation of the internal RC oscillator circuit.

2) Setting this bit to 0 starts the oscillation of the internal RC oscillator circuit.

3) When a reset occurs or when HOLD mode is entered, this bit is cleared and the internal RC oscillator circuit is enabled for oscillation.

CFSTOP (bit 0): Main clock oscillator control

1) Setting this bit to 1 stops the oscillation of the main clock oscillator circuit.

2) Setting this bit to 0 starts the oscillation of the main clock oscillator circuit.

3) When a reset occurs or when HOLD mode is entered, this bit is cleared and the main clock oscillator circuit is enabled for oscillation.

4.2.4.3   XT2 general-purpose port output control register (XT2PC)   (8-bit register)

1) This register is an 8-bit register that controls the general- purpose output (N-channel open drain type) at the XT2 pin.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE43	0000 0000	R/W	XT2PC	XT2PCB7	XT2PCB6	XT2PCB5	XT2PCB4	XT2PCB3	XT2PCB2	XT2DR	XT2DT

XT2PCB7-XT2PCB2 (bits 7 to 2): General-purpose flags

These bits can be used as general-purpose flag bits. Any manipulation of these bits exerts no influence on the operation of this function block.

XT2DR (bit 1): XT2 input/output control

XT2DT (bit 0): XT2 output data

Register Data		Port XT2 State	
XT2DT	XT2DR	Input	Output
0	0	Enabled	Open
1	0	Enabled	Open
0	1	Enabled	Low
1	1	Enabled	Open

Note: The XT2 general-purpose output port function is disabled when EXTOSC (OCR register (FE0EH), bit 6) is set to 1. To enable this port as a general-purpose output port, set EXTOSC to 0.

4.2.4.4   USB-dedicated PLL oscillator control register (PLLCNT)   (8-bit register)

1) This register is an 8-bit register that controls the operation of the USB-dedicated PLL oscillator circuit.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE0D	0000 0000	R/W	PLLCNT	SELREF2	SELREF1	SELREF0	PLLTEST	VCOSTP	CMPSTP	PLLCNT1	PLLCNT0

SELREF2 (bit 7): PLL reference clock frequency select

SELREF1 (bit 6): PLL reference clock frequency select

SELREF0 (bit 5): PLL reference clock frequency select

1) Theses bits are used to select the frequency of the ceramic resonator connected to the CF pins (CF1 and CF2).

Ceramic Resonator Frequencies

SELREF[2:0]	Frequency (MHz)
000	8
001	9
010	10
011	11
100	12
101	2
110	3
111	16

PLLTEST (bit 4): Test bit for the PLL circuit. This bit must always be set to 0.

VCOSTP (bit 3): PLLVCO operation control flag

CMPSTP (bit 2): PLL phase comparator operation control flag
1) This bit must be set to 0 together with VCOSTP and CMPSTP to generate the USB 48 MHz clock from the internal PLL circuit. In this case, P3DDR (FE4DH) bit 4 (P34DDR) and P3 (FE4CH) bit 4 (P34) must be set to 0. In addition, it is necessary to connect an external filter circuit to the P34/UFILT pin as shown in Figure 3.19.3.
2) This bit must be set to 1 if the internal PLL circuit is not to be used.

PLLCNT1 (bit 1): Reserved bit.

PLLCNT0 (bit 0): Reserved bit.

4.2.4.5   USB frequency divided clock control register (USBDIV)   (8-bit register)
1) This register is used to select the frequency of the frequency divided clock that is derived from the USB 48 MHz clock.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE04	0000 0000	R/W	USBDIV	CF48ON	DVCKON	DVCKDR	P73NDL	CF12OFF	UDVSEL2	UDVSEL1	UDVSEL0

CF48ON (bit 7): Reserved bit. It must always be set to 0.

DVCKON (bit 6): Frequency divided clock select flag
1) This bit must be set to 1 to select the frequency divided clock derived from the USB 48 MHz clock as the main clock. When the OCR register is configured to select the main clock (CLKCB4 set to 1) in this case, the frequency divided clock is supplied as the system clock.
2) This bit must be set to 0 to select the CF clock as the main clock.
3) When changing the state of this flag, make sure that the OCR register system clock select bit is set to a setting other than the main clock (CLKCB4 set to 0).

DVCKDR (bit 5): Frequency divided clock external output control flag
1) This bit must be set to 1 to generate the frequency divided clock derived from the USB 48 MHz clock from pin P73. The frequency divided clock is transmitted from pin P73 when the P73 output enable bit (P7 register, bit 7) is set to 1 and the P73 data latch (P7 register, bit 3) is set to 0 in this case.
2) If the frequency divided clock is not to be output to the external circuit, this bit must be set to 0.

System Clock

P73NDL (bit 4): Reserved bit. This bit must always be set to 0.

CF12BOFF (bit 3): Reserved bit. This bit must always be set to 0.

UDVSEL2 (bit 2): Frequency divided clock frequency select
UDVSEL1 (bit 1): Frequency divided clock frequency select
UDVSEL0 (bit 0): Frequency divided clock frequency select

1) These bits are used to select the frequency of the frequency divided clock derived from the USB 48 MHz clock.

2) The frequency divided clock must be set to a value between 8 to 16 MHz when the frequency divided clock is supplied as the system clock and USB function control circuit is to be operated.

3) When changing the frequency divided clock setting from a value other than its initial value (UDVSEL = 000), turn on the "frequency clock suspended" state (UDVSEL = 000), then set a new value.

Example: Changing the frequency divided clock frequency from 12 MHz to 8 MHz

UDVSEL=100→000→011

Frequency Divided Clock Frequencies

UDVSEL[2:0]	Frequency (MHz)
000	Frequency divided clock suspended
001	4.8
010	6
011	8
100	12
101	24 (Inhibited)
110	4
111	16

4.2.4.6 System clock divider control register (CLKDIV) (3-bit register)

1) This register controls the frequency division processing of the system clock.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE0C	HHHH H000	R/W	CLKDIV	-	-	-	-	-	CLKDV2	CLKDV1	CLKDV0

(Bits 7 to 3): These bits do not exist. They are always read as 1.

CLKDV2 (bit 2):
CLKDV1 (bit 1): **These bits define the division ratio of the system clock.**
CLKDV0 (bit 0):

CLKDV2	CLKDV1	CLKDV0	Division Ratio
0	0	0	$\frac{1}{1}$
0	0	1	$\frac{1}{2}$
0	1	0	$\frac{1}{4}$
0	1	1	$\frac{1}{8}$
1	0	0	$\frac{1}{16}$
1	0	1	$\frac{1}{32}$
1	1	0	$\frac{1}{64}$
1	1	1	$\frac{1}{128}$

4.3   Standby Function

4.3.1   Overview

This series of microcontrollers supports three standby modes, called HALT, HOLD, and X'tal HOLD modes, that are used to reduce current consumption at power-failure time or in program standby mode. In a standby mode, the execution of all instructions is suspended.

4.3.2   Functions

1)   HALT mode

- The microcontroller suspends the execution of instructions but its peripheral circuits continue processing.
- HALT mode is entered by setting bit 0 of the PCON register to 1.
- Bit 0 of the PCON register is cleared and the microcontroller returns to the normal operating mode when a reset occurs or an interrupt request is accepted.

2)   HOLD mode

- All oscillations are suspended. The microcontroller suspends the execution of instructions and its peripheral circuits stop processing.
- HOLD mode is entered by setting bit 1 of the PCON register to 1 when bit 2 is set to 0. In this case, bit 0 of the PCON register (HALT mode setting flag) is automatically set.
- When a reset occurs or a HOLD mode release signal (INT0, INT1, INT2, INT4, INT5, P0INT, or USB bus active) occurs, bit 1 of the PCON register is cleared and the microcontroller switches into HALT mode.

3)   X'tal HOLD mode

- All oscillations except the subclock oscillation are suspended. The microcontroller suspends the execution of instructions and all the peripheral circuits except the base timer stop processing.
- X'tal HOLD mode is entered by setting bit 1 of the PCON register to 1 when bit 2 is set to 1. In this case, bit 0 of the PCON register (HALT mode setting flag) is automatically set.
- When a reset occurs or a X'tal HOLD mode release signal (base timer interrupt, INT0, INT1, INT2, INT4, INT5, P0INT, USB bus active, or remote control receive) occurs, bit 1 of the PCON register is cleared and the microcontroller switches into HALT mode.

Note 1: Do not allow the microcontroller to enter into HALT, HOLD, or X'tal HOLD mode while AD conversion is in progress. Make sure that ADSTART is set to 0 before placing the microcontroller into one of the above-mentioned standby modes.

4.3.3   Related Register

4.3.3.1   Power control register (PCON)   (3-bit register)

1)   This register is a 3-bit register that specifies the operating mode (normal/HALT/HOLD/ X'tal HOLD).

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE07	HHHH H000	R/W	PCON	-	-	-	-	-	XTIDLE	PDN	IDLE

(Bits 7 to 3): These bits do not exist. They are always read as 1.

Standby

XTIDLE (bit 2): X'tal HOLD mode setting flag

PDN (bit 1): HOLD mode setting flag

XTIDLE	PDN	Operating mode
—	0	Normal or HALT mode
0	1	HOLD mode
1	1	X'tal HOLD mode

1) These bits must be set with an instruction.
   - If the microcontroller enters HOLD mode, all oscillations (main clock, subclock, RC, and PLL) are suspended and bits 0, 1, 4, and 5 of the OCR are set to 0.
   - When the microcontroller returns from HOLD mode, the main clock and RC oscillators resume oscillation. The subclock and PLL oscillators restore the state that is established before HOLD mode is entered and the system clock is set to RC.
   - If the microcontroller enters X'tal HOLD mode, all oscillations except XT (main clock, RC, and PLL) are suspended but the contents of the OCR register remain unchanged.
   - When the microcontroller returns from X'tal HOLD mode, the system clock to be used when X'tal HOLD mode is entered needs to be set to either subclock or RC because it is impossible to reserve the oscillation stabilization time for the main clock.
   - Since X'tal HOLD mode is used usually for low-current clock counting, less current will be consumed if the system clock is switched to the subclock and the main clock and RC oscillations are suspended before X'tal HOLD mode is entered.

2) XTIDLE must be cleared with an instruction.

3) PDN is cleared when a HOLD mode release signal (INT0, INT1, INT2, INT4, INT5, P0INT, USB bus active, or remote control receive) or a reset signal occurs.

IDLE (bit 0): HALT mode setting flag
1) Setting this bit places the microcontroller into HALT mode.
2) This bit is cleared on acceptance of an interrupt request or on receipt of a reset signal.

Table 4.3.1   Standby Mode Operations

Item/mode	Reset State	HALT Mode	HOLD Mode	X'tal HOLD Mode
Entry conditions	• $\overline{\text{RES}}$ applied • Reset from watchdog timer	PCON register Bit 1=0 Bit 0=1	PCON register Bit 2=0 Bit 1=1	PCON register Bit 2=1 Bit 1=1
Data changed on entry	Initialized as shown in separate table.	• WDT bits 2 to 0 are cleared if WDT register (FE0F), bit 4 is set. • WDTCNT bit 5 is cleared if WDTCNT register (FE79), bits 4 and 3 are set to 0 and 1 respectively.	• WDT bits 2 to 0 are cleared if WDT register (FE0F), bit 4 is set. • WDTCNT bit 5 is cleared if WDTCNT register (FE79), bits 4 and 3 are set to 0 and 1 respectively. • PCON, bit 0 turns to 1. • OCR register (FE0E), bits 5, 4, 1, and 0 are cleared.	• WDT bits 2 to 0 are cleared if WDT register (FE0F), bit 4 is set. • WDTCNT bit 5 is cleared if WDTCNT register (FE79), bits 4 and 3 are set to 0 and 1 respectively. • PCON, bit 0 turns to 1.
Main clock oscillation	Running	State established at entry time	Stopped	Stopped
Internal RC oscillation	Running	State established at entry time	Stopped	Stopped
Subclock oscillation	Stopped	State established at entry time	Stopped	State established at entry time
USB-dedicated PLL oscillation	Running	State established at entry time	Stopped	Stopped
CPU	Initialized	Stopped	Stopped	Stopped
I/O pin state	See Table 4.3.2.	←	←	←
RAM	• $\overline{\text{RES}}$: Undefined • When watchdog timer reset: Data preserved	Data preserved	Data preserved	Data preserved
Base timer	Stopped	State established at entry time	Stopped	State established at entry time
Peripheral modules except base timer	Stopped	State established at entry time (Note 2)	Stopped	Stopped
Exit conditions	Entry conditions canceled.	• Interrupt request accepted. • Reset/entry conditions established	• Interrupt request from INT0 to INT2, INT4, INT5, P0INT, or USB bus active • Reset/entry conditions established	• Interrupt request from INT0 to INT2, INT4, INT5, P0INT, USB bus active, base timer, or remote control receiver circuit • Reset/entry conditions established
Returned mode	Normal mode	Normal mode (Note1)	HALT mode (Note1)	HALT mode(Note1)
Data changed on exit	None	PCON register, bit 0=0	PCON register, bit 1=0	PCON register, bit 1=0

Note 1: The microcontroller switches into the reset state if it exits the current mode on the establishment of reset/entry conditions.

Note 2: Part of the serial transfer function and USB function control circuit are stopped.

Standby

Table 4.3.2   Pin States and Operating Modes

Pin Name	Reset Time	Normal Mode	HALT Mode	HOLD Mode	On Exit from HOLD
$\overline{\text{RES}}$	• Input	←	←	←	←
XT1	• Input • X'tal oscillator will not start. • Feedback resistor between XT1 and XT2 is turned off.	• Controlled by register OCR (FE0EH) as X'tal oscillator input pin • XT1 data can be read through a register OCR (FE0EH) (0 is always read in oscillation mode.) • Feedback resistor between XT1 and XT2 is controlled by a program.	←	• Oscillation suspended when used as X'tal oscillator input pin * Oscillation state maintained in X'tal HOLD mode • Feedback resistor between XT1 and XT2 is in the state established at entry time.	• HOLD mode established at entry time
XT2	• Input • X'tal oscillator will not start. • Feedback resistor between XT1 and XT2 is turned off.	• Controlled by register OCR (FE0EH) as X'tal oscillator output pin • XT2 data can be read through a register OCR (FE0EH). • Feedback resistor between XT1 and XT2 is controlled by a program.	←	• Oscillation suspended when used as X'tal oscillator input pin. Always set to VDD level regardless of XT1 state * Oscillation state maintained in X'tal HOLD mode • Feedback resistor between XT1 and XT2 is in the state established at entry time.	• HOLD mode established at entry time
CF1	• CF oscillator inverter input • Feedback resistor present between CF1 and CF2.	• CF oscillator inverter input • Enabled/disabled by a register OCR (FE0EH) • Feedback resistor present between CF1 and CF2.	←	• CF oscillator inverter input • Oscillation enabled • Feedback resistor present between CF1 and CF2.	• Same as reset time
CF2	• CF oscillator inverter output • Oscillation enabled	• CF oscillator inverter output • Enabled/disabled by a register OCR (FE0EH) • Always set to VDD level regardless of CF1 state when oscillation is suspended.	←	• CF oscillator inverter output • Oscillation suspended • Always set to VDD level regardless of CF1 state	• Same as reset time
P00 to P07 P10 to P17 P20 to P27 P30 to P34	• Input mode • Pull-up resistor off	• Input/output/pull-up resistor controlled by a program	←	←	←
P70	• Input mode • Pull-up resistor off	• Input/output/pull-up resistor controlled by a program • N-channel output transistor for watchdog timer controlled by a program (on time is automatic)	• Input mode • Pull-up resistor off • N-channel output transistor for watchdog timer is off. (automatic on-time extension function reset)	←	• Same as in normal mode
P71 to P73	• Input mode • Pull-up resistor off	• Input/output/pull-up resistor controlled by a program	←	←	←

4-16

♦ Reset state entry conditions
• Low level applied to $\overline{RES}$ pin.
• Reset signal generated by
  watchdog timer

• All modes

• Reset
  • Main clock started
  • Subclock stopped
  • RC oscillator started
  • USB-dedicated PLL oscillator
    started
  • All registers initialized

♦ Reset state cancellation conditions
• Lapse of predetermined time after
  resetting conditions are set.

♦ HOLD mode entry conditions
• PCON register (FE07), bit 2 set to
  0 and bit 1 to 1

♦ X'tal HOLD mode entry conditions
• PCON register (FE07), bit 2 set to
  1 and bit 1 to 1

• Normal operating mode
  • Start/stop of oscillators
    programmable
  • CPU and peripheral modules run
    normally.

• HOLD mode
  • All oscillators stopped
  • Since OCR register, bits 0, 1, 4,
    and 5 are cleared, the main clock
    and RC oscillator are started and
    RC oscillator is designated as
    system clock source when HOLD
    mode is released.
  • CPU and peripheral modules are
    stopped.

• X'tal HOLD
  • Main clock, RC, and PLL oscillators
    stopped. Subclock retains the state
    established when X'tal HOLD
    mode is entered.
  • Contents of OCR register remain
    unchanged
  • Contents of PLLCNT and USBDIV
    registers remain unchanged
  • CPU enters this mode after
    selecting subclock or RC oscillator
    clock as system clock and stopping
    main clock.
  • CPU and all peripheral modules
    except base timer stop operation
    and base timer retains the state
    established when X'tal HOLD
    mode is entered.
  • When X'tal HOLD mode is exited,
    the oscillators return to the state
    established when the mode is
    entered

• HALT mode
  • All oscillators retain the state
    established when HALT mode is
    entered.
  • CPU stopped. Peripheral
    modules operate.

♦ HOLD mode release conditions
• INT0 or INT1 level interrupt
  request generated
• Request for INT2, INT4, INT5,
  USB bus active, or port 0 interrupt
  generated

♦ HALT mode entry conditions
• PCON register (FE07), bit 1 set to
  0 and bit 0 to 1

♦ HALT mode release conditions
• Interrupt request accepted (Note 2)
• Resetting conditions established (Note 1)

♦ X'tal HOLD mode release
  conditions
• Base timer interrupt request
  generated.
• INT0 or INT1 level interrupt request
  generated.
• Request for INT2, INT4, INT5, USB
  bus active, port 0 interrupt, or
  remote control receive generated.
• Resetting condition established .
  (Note 1)

Note 1: The CPU enters the reset state when the resetting conditions are established.
Note 2: The CPU cannot return from HALT mode since no interrupt request can be accepted unless its interrupt level is
        higher than the interrupt level that placed the CPU into HALT, HOLD, or X'tal HOLD mode.

Interrupt level at which the CPU entered HALT, HOLD, or X'tal HOLD mode	Interrupt request level that can release HALT mode
No interrupt request present	X, H and L levels
L level	X and H levels
H level	X level
X level	None (unable to release with interrupt)

Fig. 4.3.1   Standby Mode State Transition Diagram

4.4 Reset Function

4.4.1 Overview

The reset function initializes the microcontroller when it is powered on or while it is running.

4.4.2 Functions

This series of microcontrollers provides the following three types of resetting function:

1) External reset via the $\overline{\text{RES}}$ pin

   The microcontroller is reset without fail by applying and holding a low level to the $\overline{\text{RES}}$ pin for 200 μs or longer. Note, however, that a low level of a small duration (less than 200 μs) is likely to trigger a reset.

   The $\overline{\text{RES}}$ pin can serve as a power-on reset pin when it is provided with an appropriate external time constant.

2) Runaway detection/reset function using an external RC watchdog timer

   The external RC watchdog timer can be used to detect and reset runaway conditions by connecting a resistor and a capacitor to its external interrupt pin (P70/INT0/T0LCP) and making an appropriate time constant element.

3) Runaway detection/reset function using an internal counter watchdog timer

   The internal counter watchdog timer can generate internal reset signals at certain time intervals on an internal low-speed RC oscillator clock.

An example of a resetting circuit is shown in Figure 4.4.1.
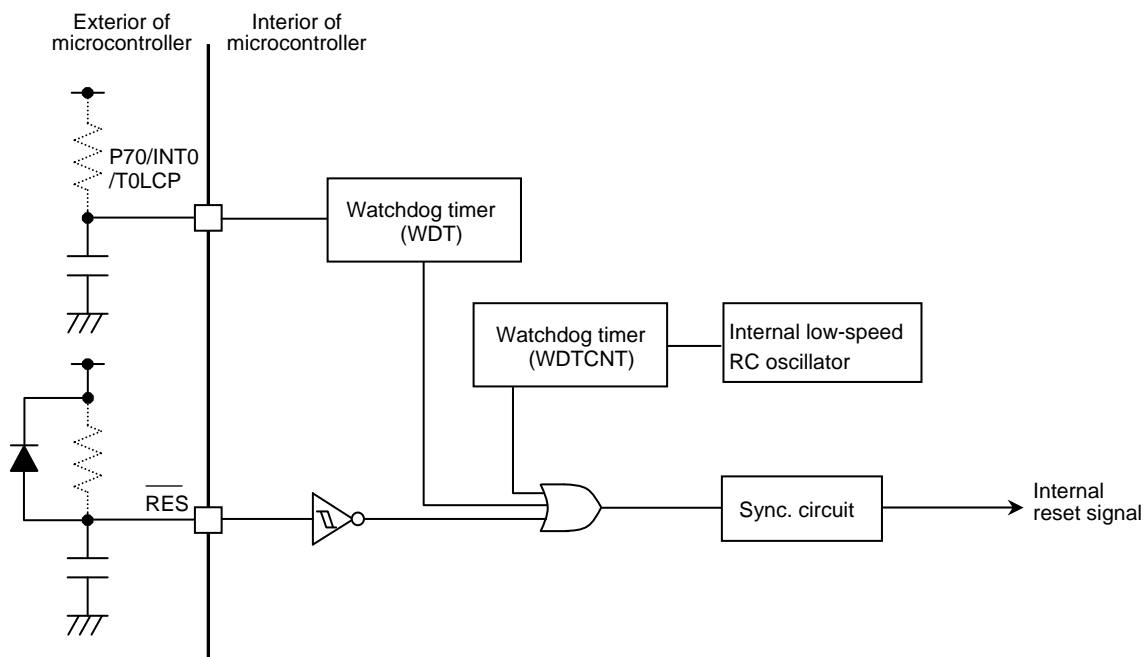
Fig. 4.4.1   Reset Circuit Block Diagram

4.4.3   Reset State

When a reset is generated by the $\overline{RES}$ pin or watchdog timer, the hardware functional blocks of the microcontroller are initialized by a reset signal that is in synchronization with the system clock.

Since the system clock is switched to the internal RC oscillator when a reset occurs, hardware initialization is also carried out immediately even at power-on time. To use the main clock subsequently, enable the oscillation of the main clock, wait until the oscillation gets stabilized, then switch the system clock source to the main clock.

On reset, the program counter is initialized to the program start address that is selected through a user option. The special function registers (SFRs) are also initialized to the values that are listed in the Special Function Register (SFR) Map shown in Appendix A-I.

<Notes and precautions>

- *The stack pointer is initialized to 0000H.*
- *Data RAM is never initialized by a reset. Consequently, the contents of RAM are undefined at power-on time.*
- *Be sure to set the $\overline{RES}$ pin to the low level when turning on the CPU. Otherwise, the CPU will be out of control during the period from power-on till the time the $\overline{RES}$ pin goes to the low level.*

4.5 Watchdog Timer Function

This series of microcontrollers incorporates two types of watchdog timer functions:

1) Watchdog timer that uses an external RC circuit
2) Watchdog timer that runs on the internal counter

4.5.1 Overview (with an External RC)

This series of microcontrollers incorporates a watchdog timer that, with an external RC circuit, detects program runaway conditions.

The watchdog timer charges the external RC circuit that is connected to the P70/INT0/T0LCP pin and, when the level at the pin reaches the high level, triggers a reset or interrupt, regarding that a program runaway occurred.

4.5.2 Functions

1) Detection of a runaway condition

   A program that discharges the RC circuit periodically needs to be prepared. If a program runaways, it will not execute instructions to discharge the RC circuit. Consequently, the P70/INT0/T0LCP pin goes to the high level, causing the watchdog timer to detect a program runaway condition.

2) Actions to be taken following the detection of a runaway condition

   The microcontroller can take one of the following actions when the watchdog timer detects a program runaway condition:

   • Reset (program reexecution)
   • External interrupt INT0 generation (program continuation)

     The priority of the external interrupt INT0 can be changed using the master interrupt enable control register (IE).

4.5.3 Circuit Configuration

The watchdog timer is made up of a high-threshold buffer, a pulse stretcher circuit, and a watchdog timer control register. Its configuration diagram is shown in Figure 4.5.1.

   • High-threshold buffer

   The high-threshold buffer detects the charging voltage of the external capacitor.

   • Pulse stretcher circuit

   The pulse stretcher circuit discharges the external capacitor for longer than the specified time to ensure reliable discharging. The stretching time is from 1920 to 2048 Tcyc.

   • Watchdog timer control register (WDT)

   This register controls the operation of the watchdog timer.
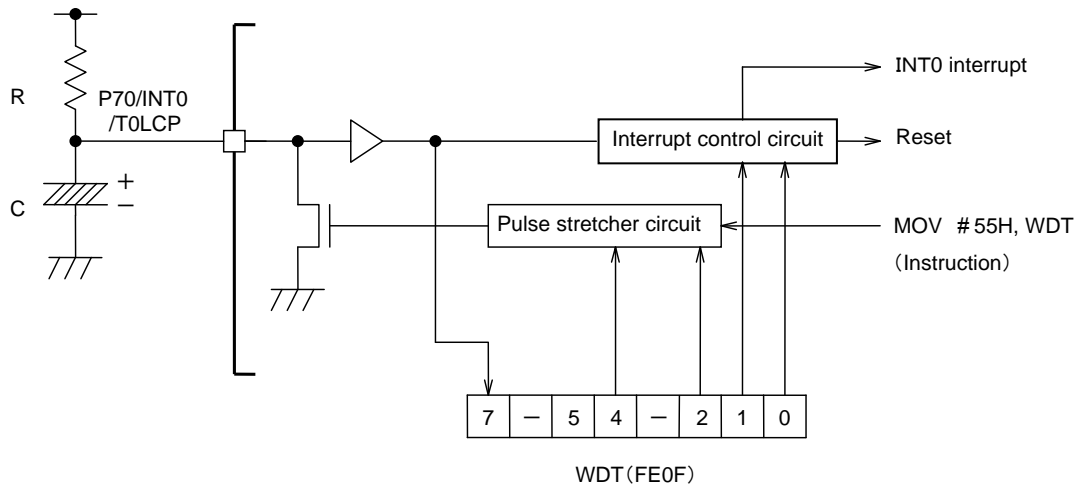
Figure 4.5.1 Watchdog Timer Circuit

4.5.4 Related Registers

4.5.4.1 Watchdog timer control register (WDT)

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE0F	0H00 H000	R/W	WDT	WDTFLG	-	WDTB5	WDTHLT	-	WDTCLR	WDTRST	WDTRUN

Bit Name	Function
WDTFLG (bit 7)	Runaway detection flag
	0: No runaway 1: Runaway
WDTB5 (bit 5)	General-purpose flag
	Can be used as a general-purpose flag.
WDTHLT (bit 4)	HALT/HOLD mode function control
	0: Enables the watchdog timer. 1: Disables the watchdog timer.
WDTCLR (bit 2)	Watchdog timer clear control
	0: Disables clearing of the watchdog timer. 1: Enables clearing of the watchdog timer.
WDTRST (bit 1)	Runaway-time reset control
	0: Disables a reset when a runaway condition is detected. 1: Enables a reset when a runway condition is detected.
WDTRUN (bit 0)	Watchdog timer operation control
	0: Maintains the state of the watchdog timer. 1: Starts the watchdog timer operation.

WDTFLG (bit 7): Runaway detection flag

This bit is set when a program runaway condition is detected by the watchdog timer. The application can identify the occurrence of a program runaway condition by monitoring this bit (provided that WDTRST is set to 1).

This bit is not reset automatically. It must be reset with an instruction.

Watchdog Timer

WDTB5 (bit 5): General-purpose flag

This bit can be used as a general-purpose flag. Manipulating this bit exerts no influence on the operation of the functional block.

WDTHLT (bit 4): HALT/HOLD mode function control

This bit enables (0) or disables (1) the watchdog timer when the microcontroller is in the HALT or HOLD state. When this bit is set to 1, WDTCLR, WDTRST and WDTRUN are reset and the watchdog timer is stopped in the HALT or HOLD state. When this bit is set to 0, WDTCLR, WDTRST and WDTRUN remain unchanged and the watchdog timer continues operation even when the microcontroller enters the HALT or HOLD state.

To use the watchdog timer function after the microcontroller returns to the normal operating mode from HALT or HOLD mode with this bit set to 1, initialize and set up the watchdog timer again for starting the operation.

WDTCLR (bit 2): Watchdog timer clear control

This bit enables (1) or disables (0) the discharge of capacitance from the external capacitor when the watchdog timer is running (WDTRUN=1). If a watchdog timer clear instruction is executed when this bit is set to 1, the pin P70/INT0/T0LCP N-channel transistor is turned on, discharging the external capacitors and clearing the watchdog timer. The pulse stretcher also functions during this process. Setting the bit to 0 disables operation of the N-channel transistors and clearing of the watchdog timer.
If this bit is set to 1 when the watchdog timer is inactive (WDTRUN=0), the N-channel transistor at the pin P70/INT0/T0LCP is turned on, discharging the external capacitors and clearing the watchdog timer.

WDTRST (bit 1): Runaway-time reset control

This bit enables (1) or disables (0) the reset sequence that is to be executed when the watchdog timer detects a program runaway condition. If this bit is set to 1, the microcontroller reexecutes the program starting at the program start address which is selected through a user option, when a program runaway condition is detected and reset is triggered. If this bit is set to 0, no reset occurs. Instead, an external interrupt INT0 is generated and a call is made to vector address 0003H.

WDTRUN (bit 0): Watchdog timer operation control

This bit starts (1) or maintains (0) the state of the watchdog timer. A 1 in this bit starts the watchdog timer function and a 0 exerts no influence on the operation of the watchdog timer. This means, that once the watchdog timer is started, a program will not be able to stop the watchdog timer (stopped by a reset).

Caution

If WDTRST is set to 1, a reset is triggered when the P70/INT0/T0LCP pin goes to the high level even if the watchdog timer is not running.

The N-channel transistor at the P70/INT0/T0LCP pin is turned on if the watchdog timer clear control bit (WDTCLR) is set to 1 when the watchdog timer is not running (WDTRUN = 0).

Keep this in mind when programming if the watchdog timer function is not to be used.   More current than usual may be consumed depending on the program or application circuit.

4.5.4.2 Master interrupt enable control register (IE)

See subsubsection 4.1.4.1, "Master interrupt enable control register," for details.

4.5.4.3 Port 7 control register (P7)

See subsubsection 3.5.3.1, "Port 7 control register," for details.

4.5.5 Using the Watchdog Timer

Code a program so that instructions for clearing the watchdog timer periodically are executed. Select the resistance R and the capacitance C such that the time constant of the external RC circuit is greater than the time interval required to clear the watchdog timer.

1) Initializing the watchdog timer

All bits of the watchdog timer control register (WDT) are reset when an external reset is triggered through the $\overline{RES}$ pin. If the P70/INT0/T0LCP pin has been charged up to the high level, discharge it down to the low level before starting the watchdog timer. The internal N-channel transistor is used for discharging. Since it has an on-resistance, a discharging time equal to the time constant of the external capacitance is required.
Set bits 0 and 4 of the Port 7 control register (P7:FE5C) to 0, 0 or 1, 1 to make the P70 port output open.

Starting discharge
Load WDT with 04H to turn on the N-channel transistor at the P70/INT0/T0LCP pin to start discharging the capacitor.

Checking the low level
Checking for data at the P70/INT0/T0LCP pin
Read the data at the P70/INT0/T0LCP pin with an LD or similar instruction. A 0 indicates that the P70/INT0/T0LCP pin is at the low level.

2) Starting the watchdog timer

(1) Set bit 2 (WDTCLR) and bit 0 (WDTRUN) to 1.

(2) Also set bit 1 (WDTRST) to 1 at the same time when a reset is to be triggered when a runaway condition is detected.

(3) To suspend the operation of the watchdog timer in HOLD or HALT mode, set bit 4 (WDTHLT) at the same time.

The watchdog timer starts functioning when bit 0 (WDTRUN) is set to 1. Once the watchdog timer starts operation, the watchdog timer control register (WDT) is disabled for write; it is allowed only to clear the watchdog timer and read the watchdog timer control register (WDT). Consequently, the watchdog timer can never be stopped with an instruction. The function of the watchdog timer is stopped only when a reset occurs or when the microcontroller enters HALT or HOLD mode with WDTHLT being set. In this case, bits WDTCLR, WDTRST, and WDTRUN are reset.

Watchdog Timer

3) Clearing the watchdog timer

Immediately when power is turned on, charging the external RC circuit that is connected to the P70/INT0/T0LCP pin is started. When voltage at this pin reaches the high level, a reset or interrupt is generated as specified in the watchdog timer control register (WDT). To run the program in the normal mode, it is necessary to periodically discharge the RC circuit before the voltage at the P70/INT0/ T0LCP pin reaches the high level (clearing the watchdog timer). Execute the following instruction to clear the watchdog timer while it is running:

MOV   #55H,WDT

This instruction turns on the N-channel transistor at the P70/INT0/T0LCP pin. Due to the pulse stretcher function (keeps the transistor on after the MOV instruction is executed), the capacitor keeps discharging for a period from a minimum of 1920 cycle times to a maximum of 2048 cycle times.

4) Detecting a runaway condition

Unless the above-mentioned instruction is executed periodically, the external RC circuit keeps charging because the watchdog timer is not cleared. As charging proceeds and the voltage at the P70/INT0/T0LCP pin reaches the high level, the watchdog timer considers that a program runaway has occurred and triggers a reset or interrupt. In this case, the runaway detection flag (WDTFLG) is set (provided that WDTRST is set to 1).

If WDTRST is found to be 1 in this case, a reset occurs and execution restarts at the program start address which is selected through a user option. If WDTRST is 0, an external interrupt (INT0) is generated and control is transferred to vector address 0003H.

● **Hints on Use**

1) To realize ultra-low-power operation using HOLD mode, it is necessary not to use the watchdog timer at all or to disable the watchdog timer from running in HOLD mode by setting WDTHLT to 1.

Be sure to set WDTCLR to 0 when the watchdog timer is not to be used.

2) The P70/INT0/T0LCP pin has two input levels. The threshold level of the input pins of the watchdog timer circuit is higher than that of the port inputs and the interrupt detection level.

Refer to the latest "SANYO Semiconductors Data Sheet" for the input levels.
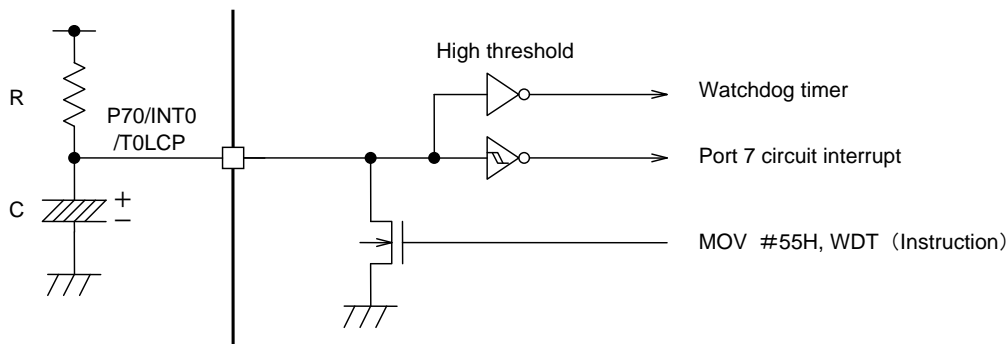
Fig. 4.5.2   P70/INT0/T0LCP Pin (without an Optional Pull-up Resistor)

3) The external resistor to be connected to the watchdog timer can be omitted by setting bits 4 and 0 of the Port 7 control register (P7:FE5C) to 0 and 1 and connecting a <u>pull-up resistor</u> to the P70/INT0/T0LCP pin (see Figure 4.5.3).

   The resistance of the pull-up resistor used in this case varies according to the power source voltage VDD. Calculate the time constant of the watchdog timer while referring to the latest "SANYO Semiconductor Data Sheet" to confirm the pull-up resistance.
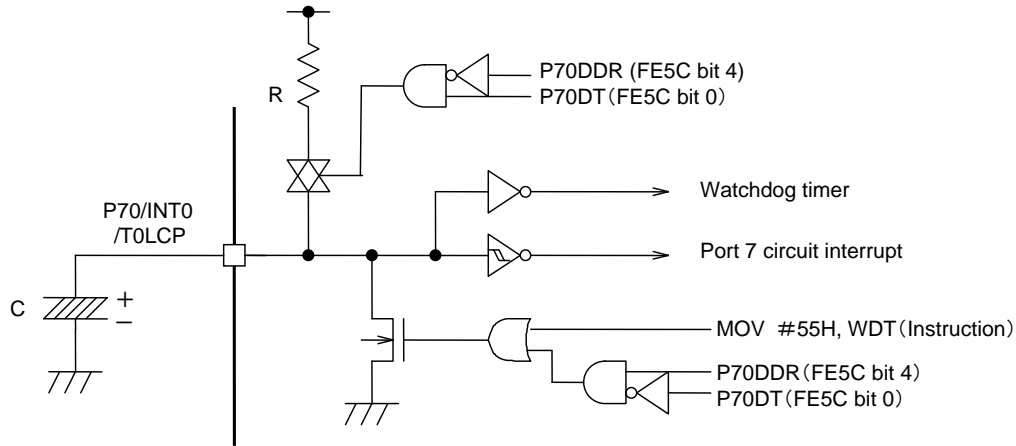
Fig. 4.5.3   Sample Application Circuit with a Pull-up Resistor

4) When the microcontroller enters HALT or HOLD mode with WDTHLT being set to 1, bits WDTCLR, WDTRST, and WDTRUN are reset. To use the watchdog timer function after the microcontroller returns to the normal operating mode from HALT or HOLD mode, initialize and set up the watchdog timer again for starting the operation.

4.5.6   Overview (with an Internal Counter)

This series of microcontrollers is provided with a watchdog timer (WDT) that has the following functions:

1) Generates an internal reset on an overflow of a timer that runs on a WDT-dedicated low-speed RC oscillator clock.

2) The continuation, termination, or holding (count value) of the WDT operation on entry into the standby mode is selectable.

4.5.7   Functions

1) Watchdog timer function

- The 16-bit up-counter (WDTCT) runs on a low-speed RC oscillator clock and generates a WDT reset signal (internal reset signal) when it reaches the count equivalent to the overflow time (one selected out of 8 levels) selected through the watchdog timer control register (WDTCNT). Then the reset detection flag (RSTFLG) is set.
  Since the WDTCT is cleared under program control, it is necessary to code the program so that the WDTCT be cleared periodically.

- Since the WDT used in this series of microcontrollers has a dedicated low-speed RC oscillator, the system continues operation even when the system clock is stopped due to a program runaway, making it possible to detect any system runaway conditions.

- The WDT operation mode on entry into the standby mode can be selected from three modes, i.e., "continuation of operation," "termination of operation," and "holding of WDTCT count value and resume WDT operation at the holding count value when the standby mode is exited." In the "continuation of operation" mode, the low-speed RC oscillator circuit continues oscillation even in the standby mode, allowing an operating current of several µA to flow at all times. (For details, refer to the latest "SANYO Semiconductors Data Sheet").

2) To control the watchdog timer (WDT), it is necessary to manipulate the following special function register.

- WDTCNT

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE79	0000 0000	R/W	WDTCNT	RSTFLG	FIX0	WDTRUN	IDLOP1	IDLOP0	WDTSL2	WDTSL1	WDTSL0

4.5.8   Circuit Configuration

4.5.8.1   WDT control register (WDTCNT)   (8-bit register)

1) This register is used to manipulate the reset detection flag, to select operations in the standby mode, to select the overflow time, and to control the operation of WDT.

Note: The WDTCNT is initialized to"00H" when a reset is generated by applying a low-level signal to the external RES pin. Bits 4 to 0 of the WDTCNT are not initialized, however, when a WDT-triggered reset occurs.

Note: The WDTCNT is disabled for writes once the WDT is started (WDTRUN set to 1). If the instruction "MOV #55H, WDTCNT" is executed in this case, the WDTCT is cleared and counting is restarted at a count value of 0 (WDTCT is not cleared if it is loaded with 55H with any other instruction).

Note: The low-speed RC oscillator circuit is started and stopped by setting bit WDTRUN (WDTCNT, bit 5) to 1 and 0, respectively. Once the oscillator starts oscillation, an operating current of several μA flows at all times (For details, refer to the latest "SANYO Semiconductors Data Sheet").

4.5.8.2 WDT counter (WDTCT) (16-bit counter)

1) Operation start/stop : Enters the standby mode when WDTRUN=0/WDTRUN=1, or when WDTRUN=1 and IDLOP1 and IDLOP0 (WDTCNT, bits 4 and 3)=1.

2) Count clock : Low-speed RC oscillator clock

3) Overflow : Generated when the WDTCT count value matches the count value designated by WDTSL2 to WDTSL0 (WDTCNT, bits 2 to 0).

    * Generates a signal to set the RSTFLG flag bit (WDTCNT, bit 7).

    * Generates a WDT reset signal and WDTRUN clear signal.

4) Reset : Enters into the standby mode when WDTRUN=0, overflow occurs, WDTRUN=1 and instruction **"MOV #55H, WDTCNT"** is executed, or WDTRUN=1 and IDLOP1 and IDLOP0=1.

* See Figure 4.5.5 for details on the WDT operation.
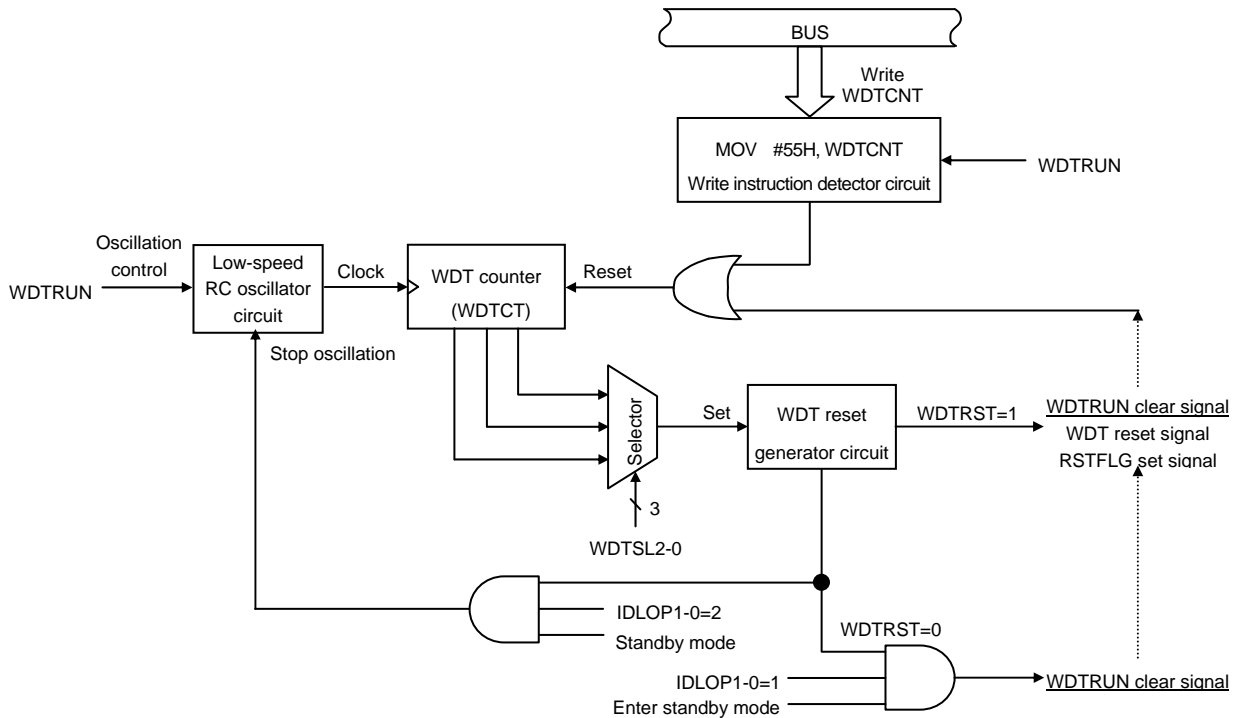
Figure 4.5.4 Watchdog Timer Operation Block Diagram
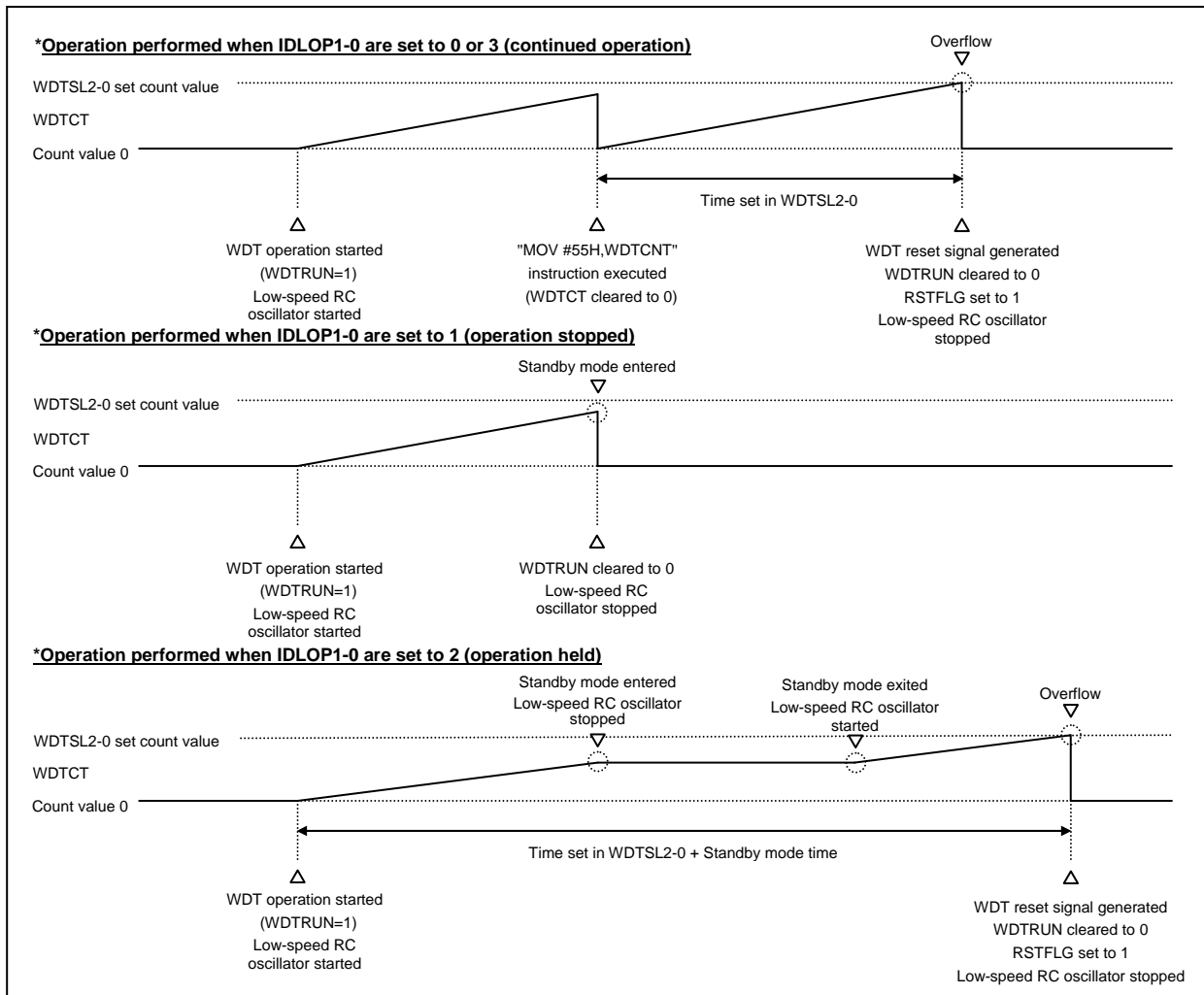
Figure 4.5.5   Sample Watchdog Timer Operation Waveforms

4.5.9   Related Register

4.5.9.1   WDT control register (WDTCNT)

1)   This register is used to manipulate the reset detection flag, to select the standby mode operation, to select the overflow time, and to control the operation of the WDT.

Address	Initial value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE79	0000 0000	R/W	WDTCNT	RSTFLG	FIX0	WDTRUN	IDLOP1	IDLOP0	WDTSL2	WDTSL1	WDTSL0

RSTFLG (bit 7): WDT reset detection flag

This bit is cleared when a reset is generated by applying a low level to the external $\overline{\text{RES}}$ pin.

This bit is set when a WDT-triggered reset occurs. This flag can be rewritten with an instruction.

FIX0 (bit 6): Test bit

This bit is for test. It must always be set to 0.

WDTRUN (bit 5): WDT operation control

Setting this bit to 0 stops the WDT operation.

Setting this bit to 1 starts the WDT operation.

IDLOP1 (bit 4):

IDLOP0 (bit 3):   **WDT standby mode operation selection**

IDLOP1	IDLOP0	WDT Standby Mode Operation
0	0	Continue operation
0	1	Stop operation
1	0	Hold operation
1	1	Continue operation

* See Figure 4.5.5 for details of the WDT operating modes.

* There are notes to be taken when running WDT by specifying "Hold operation." See Section 4.5.10, "Notes on the Use of the Watchdog Timer."

WDTSL2 (bit 2):

WDTSL1 (bit 1):   **WDT counter (WDTCT) control**

WDTSL0 (bit 0):

WDTSL2	WDTSL1	WDTSL0	WDT Counter Count Value	
0	0	0	512	(17.06 ms)
0	0	1	1024	(34.13 ms)
0	1	0	2048	(68.26 ms)
0	1	1	4096	(136.53 ms)
1	0	0	8192	(273.06 ms)
1	0	1	16384	(546.13 ms)
1	1	0	32768	(1092.26 ms)
1	1	1	65536	(2184.53 ms)

* Time values enclosed in parentheses refer to the time for a WDTCT overflow to occur when the low-speed RC oscillation frequency is 30 kHz (typical). The low-speed RC oscillation frequency may deviate from the target frequency. For details, refer to the latest "SANYO Semiconductors Data Sheet."

Note: The WDTCNT is initialized to "00H" when a low-level signal is applied to the external $\overline{RES}$ pin . Bits 4 to 0 of the WDTCNT are not initialized, however, when a WDT-triggered reset occurs.

*Note: The WDTCNT is disabled for writes once the WDT is started (WDTRUN set to 1). If the instruction **"MOV   #55H, WDTCNT"** is executed in this case, the WDTCT is cleared and counting is restarted at a count value of 0 (the WDTCT is not cleared if it is loaded with 55H with any other instruction).*

Note: The low-speed RC oscillator circuit is started and stopped by setting bit WDTRUN to 1 and 0, respectively. Once the oscillator starts oscillation, an operating current of several µA flows at all times (For details, refer to the latest "SANYO Semiconductors Data Sheet").

4.5.10   Notes on the Use of the Watchdog Timer

1)   When "Hold operation" is selected in the standby mode operation (IDLOP1-IDLOP0 = 2)

   • When the microcontroller is placed in standby mode (HALT/HOLD) after the watchdog timer is started with "Hold operation" selected, the low-speed RC oscillator circuit stops oscillation and the watchdog timer stops counting and retains the count value. When the CPU subsequently exits the standby mode, the low-speed RC oscillator circuit resumes oscillation and the watchdog timer starts counting. If the period between the release of the standby mode and the next entry into a standby mode is less than **"low-speed RC oscillator clock $\times$ 4**," however, the low-speed RC oscillator circuit may not stop oscillation when the microcontroller enters a standby mode. In such a case (a standby mode is on), several μA of operating current flows because the low-speed RC oscillator circuit is active though the watchdog timer stops counting.

   To minimize the standby power requirement of the set, code the program so that an interval of **"low-speed RC oscillator clock $\times$ 4''** or more be provided between release from a standby mode and entry into the next standby mode (Note that the oscillation frequency of the low-speed RC oscillator may deviate from the target frequency. Refer to the latest "SANYO Semiconductors Data Sheet" for details).

Appendixes

Table of Contents

Appendix I

- Special Functions Register (SFR) Map

Appendix II

- Port 0 Block Diagram
- Port 1 Block Diagram
- Port 2 Block Diagram
- Port 3 Block Diagram
- Port 7 Block Diagram
- PWM0, PWM1 Block Diagram

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0-FFF	XXXX XXXX	R/W	RAM4KB	9 bits long									
FE00	0000 0000	R/W	AREG		–	AREG7	AREG6	AREG5	AREG4	AREG3	AREG2	AREG1	AREG0
FE01	0000 0000	R/W	BREG		–	BREG7	BREG6	BREG5	BREG4	BREG3	BREG2	BREG1	BREG0
FE02	0000 0000	R/W	CREG		–	CREG7	CREG6	CREG5	CREG4	CREG3	CREG2	CREG1	CREG0
FE03	HHHH HHHH		None										
FE04	0000 0000	R/W	USBDIV		–	CF480N	DVCKON	DVCKDR	P73NDL	CF120FF	UDVSEL2	UDVSEL1	UDVSEL0
FE05	HHHH HHHH		None										
FE06	0000 0000	R/W	PSW		–	CY	AC	PSWB5	PSWB4	LDCBNK	OV	P1	PARITY
FE07	HHHH H000	R/W	PCON		–	–	–	–	–	–	XTIDLE	PDN	IDLE
FE08	0000 HH00	R/W	IE		–	IE7	XFLG	HFLG	LFLG	–	–	XCNT1	XCNT0
FE09	0000 0000	R/W	IP		–	IP4B	IP43	IP3B	IP33	IP2B	IP23	IP1B	IP13
FE0A	0000 0000	R/W	SPL		–	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
FE0B	0000 0000	R/W	SPH		–	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8
FE0C	HHHH H000	R/W	CLKDIV		–	–	–	–	–	–	CLKDV2	CLKDV1	CLKDV0
FE0D	0000 0000	R/W	PLLCNT			SELREF2	SELREF1	SELREF0	PLLTEST	VCOSTP	CMPSTP	PLLCNT1	PLLCNT0
FE0E	0000 XX00	R/W	OCR	XT1 and XT2 read at bits 2 and 3	–	CLKSGL	EXTOSC	CLKCB5	CLKCB4	XT2IN	XT1IN	RCSTOP	CFSTOP
FE0F	0H00 H000	R/W	WDT		–	WDTFLG	–	WDTB5	WDTHLT	–	WDTCLR	WDTRST	WDTRUN
FE10	0000 0000	R/W	T0CNT		–	T0HRUN	T0LRUN	T0LONG	T0LEXT	T0HCMP	T0HIE	T0LCMP	T0LIE
FE11	0000 0000	R/W	T0PRR	Prescaler is 8 bits long. (max. 256 Tcyc)	–	T0PRR7	T0PRR6	T0PRR5	T0PRR4	T0PRR3	T0PRR2	T0PRR1	T0PRR0
FE12	0000 0000	R	T0L		–	T0L7	T0L6	T0L5	T0L4	T0L3	T0L2	T0L1	T0L0
FE13	0000 0000	R	T0H		–	T0H7	T0H6	T0H5	T0H4	T0H3	T0H2	T0H1	T0H0
FE14	0000 0000	R/W	T0LR		–	T0LR7	T0LR6	T0LR5	T0LR4	T0LR3	T0LR2	T0LR1	T0LR0
FE15	0000 0000	R/W	T0HR		–	T0HR7	T0HR6	T0HR5	T0HR4	T0HR3	T0HR2	T0HR1	T0HR0
FE16	XXXX XXXX	R	T0CAL	Timer 0 capture register L	–	T0CAL7	T0CAL6	T0CAL5	T0CAL4	T0CAL3	T0CAL2	T0CAL1	T0CAL0
FE17	XXXX XXXX	R	T0CAH	Timer 0 capture register H	–	T0CAH7	T0CAH6	T0CAH5	T0CAH4	T0CAH3	T0CAH2	T0CAH1	T0CAH0
FE18	0000 0000	R/W	T1CNT		–	T1HRUN	T1LRUN	T1LONG	T1PWM	T1HCMP	T1HIE	T1LCMP	T1LIE
FE19	0000 0000	R/W	T1PRR		–	T1HPRE	T1HPRC2	T1HPRC1	T1HPRC0	T1LPRE	T1LPRC2	T1LPRC1	T1LPRC0
FE1A	0000 0000	R	T1L		–	T1L7	T1L6	T1L5	T1L4	T1L3	T1L2	T1L1	T1L0
FE1B	0000 0000	R	T1H		–	T1H7	T1H6	T1H5	T1H4	T1H3	T1H2	T1H1	T1H0
FE1C	0000 0000	R/W	T1LR		–	T1LR7	T1LR6	T1LR5	T1LR4	T1LR3	T1LR2	T1LR1	T1LR0
FE1D	0000 0000	R/W	T1HR		–	T1HR7	T1HR6	T1HR5	T1HR4	T1HR3	T1HR2	T1HR1	T1HR0

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE1E	HHHH HHHH		None										
FE1F	HHHH HHHH		None										
FE20	0000 HHHH	R/W	PWM0L	PWM0 compare L (additional)	–	PWM0L3	PWM0L2	PWM0L1	PWM0L0	–	–	–	–
FE21	0000 0000	R/W	PWM0H	PWM0 compare H (base)	–	PWM0H7	PWM0H6	PWM0H5	PWM0H4	PWM0H3	PWM0H2	PWM0H1	PWM0H0
FE22	0000 HHHH	R/W	PWM1L	PWM1 compare L (additional)	–	PWM1L3	PWM1L2	PWM1L1	PWM1L0	–	–	–	–
FE23	0000 0000	R/W	PWM1H	PWM1 compare H (base)	–	PWM1H7	PWM1H6	PWM1H5	PWM1H4	PWM1H3	PWM1H2	PWM1H1	PWM1H0
FE24	0000 0000	R/W	PWM0C	Controls PWM0 and PWM1.	–	PWM0C7	PWM0C6	PWM0C5	PWM0C4	ENPWM1	ENPWM0	PWM0OV	PWM0IE
FE25	HHHH HHXX	R	PWM01P		–	–	–	–	–	–	–	PWM1IN	PWM0IN
FE26	HHHH HHHH		None										
FE27	0000 0000	R/W	RM2CNT		–	RM2RUN	RM2FMT2	RM2FMT1	RM2FMT0	RM2DINV	RM2CK2	RM2CK1	RM2CK0
FE28	0000 0000	R/W	RM2INT		–	RM2GPOK	RM2GPIE	RM2DERR	RM2ERIE	RM2SFUL	RM2SFIE	RM2REND	RM2ENIE
FE29	0000 0000	R	RM2SFT		–	RM2SFT7	RM2SFT6	RM2SFT5	RM2SFT4	RM2SFT3	RM2SFT2	RM2SFT1	RM2SFT0
FE2A	XXXX XXXX	R	RM2RDT		–	RM2RDT7	RM2RDT6	RM2RDT5	RM2RDT4	RM2RDT3	RM2RDT2	RM2RDT1	RM2RDT0
FE2B	0000 0000	R/W	RM2CTPR	RM2CTPR[3:0] is read-only.	–	RM2GPR1	RM2GPR0	RM2DPR1	RM2DPR0	RM2HOLD	RM2BCT2	RM2BCT1	RM2BCT0
FE2C	0000 0000	R/W	RM2GPW		–	RM2GPH3	RM2GPH2	RM2GPH1	RM2GPH0	RM2GPL3	RM2GPL2	RM2GPL1	RM2GPL0
FE2D	0000 0000	R/W	RM2DT0W		–	RM2D0H3	RM2D0H2	RM2D0H1	RM2D0H0	RM2D0L3	RM2D0L2	RM2D0L1	RM2D0L0
FE2E	0000 0000	R/W	RM2DT1W		–	RM2D1H3	RM2D1H2	RM2D1H1	RM2D1H0	RM2D0H3	RM2D0H2	RM2D0H1	RM2D0H0
FE2F	0H00 0000	R/W	RM2XHW		–	RM2RDIR	–	RM2D1H4	RM2D1L4	RM2D0H4	RM2D0L4	RM2GPH4	RM2GPL4
FE30	0000 0000	R/W	SCON0		–	SIOBNK	SIOWRT	SIORUN	SIOCTR	SIODIR	SIO0VR	SIOEND	SIOIE
FE31	0000 0000	R/W	SBUF0		–	SBUF07	SBUF06	SBUF05	SBUF04	SBUF03	SBUF02	SBUF01	SBUF00
FE32	0000 0000	R/W	SBR0		–	SBRG07	SBRG06	SBRG05	SBRG04	SBRG03	SBRG02	SBRG01	SBRG00
FE33	0000 0000	R/W	SCTR0		–	SCTR07	SCTR06	SCTR05	SCTR04	SCTR03	SCTR02	SCTR01	SCTR00
FE34	0000 0000	R/W	SCON1		–	SI1M1	SI1M0	SI1RUN	SI1REC	SI1DIR	SI10VR	SI1END	SI1IE
FE35	0000 0000	R/W	SBUF1	9-bit REG	SBUF18	SBUF17	SBUF16	SBUF15	SBUF14	SBUF13	SBUF12	SBUF11	SBUF10
FE36	0000 0000	R/W	SBR1		–	SBRG17	SBRG16	SBRG15	SBRG14	SBRG13	SBRG12	SBRG11	SBRG10
FE37	0000 0000	R/W	SWCON0	S0XBYT[4:0] is read-only.	–	S0WSTP	SWCONB6	SWCONB5	S0XBYT4	S0XBYT3	S0XBYT2	S0XBYT1	S0XBYT0
FE38	HHHH HHHH		None		–								
FE39	HHHH HHHH		None		–								
FE3A	HHHH HHHH		None										
FE3B	HHHH HHHH		None										
FE3C	HHHH HHHH		None										
FE3D	HHHH HHHH		None										

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE3E	HHHH HHHH		None										
FE3F	HHHH HHHH		None										
FE40	0000 0000	R/W	P0		–	P07	P06	P05	P04	P03	P02	P01	P00
FE41	HH00 0000	R/W	P0DDR		–	–	–	P0FLG	P0IE	P0HPU	P0LPU	P0HDDR	P0LDDR
FE42	HHHH HHHH		None										
FE43	0000 0000	R/W	XT2PC		–	XT2PCB7	XT2PCB6	XT2PCB5	XT2PCB4	XT2PCB3	XT2PCB2	XT2DR	XT2DT
FE44	0000 0000	R/W	P1		–	P17	P16	P15	P14	P13	P12	P11	P10
FE45	0000 0000	R/W	P1DDR		–	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
FE46	0000 0000	R/W	P1FCR		–	P17FCR	P16FCR	P15FCR	P14FCR	P13FCR	P12FCR	P11FCR	P10FCR
FE47	0HHH H0H0	R/W	P1TST		–	FIX0	–	–	–	–	DSNK0T	–	FIX0
FE48	0000 0000	R/W	P2		–	P27	P26	P25	P24	P23	P22	P21	P20
FE49	0000 0000	R/W	P2DDR		–	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
FE4A	0000 0000	R/W	I45CR		–	INT5HEG	INT5LEG	INT5IF	INT5IE	INT4HEG	INT4LEG	INT4IF	INT4IE
FE4B	0000 0000	R/W	I45SL		–	I5SL3	I5SL2	I5SL1	I5SL0	I4SL3	I4SL2	I4SL1	I4SL0
FE4C	HHH0 0000	R/W	P3		–	–	–	–	P34	P33	P32	P31	P30
FE4D	HHH0 0000	R/W	P3DDR		–	–	–	–	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
FE4E	HHHH HHHH		None										
FE4F	0000 0000	R/W	P0FCRU		–	T70E	T60E	SCK0SL5	SCK0SL4	CLK0EN	CK0DV2	CK0DV1	CK0DV0
FE50	HHHH HHHH		None										
FE51	HHHH HHHH		None										
FE52	HHHH HHHH		None										
FE53	HHHH HHHH		None										
FE54	HHHH HHHH		None										
FE55	HHHH HHHH		None										
FE56	HHHH HHHH		None										
FE57	HHHH HHHH		None										
FE58	0000 0000	R/W	ADCRC		–	ADCHSEL3	ADCHSEL2	ADCHSEL1	ADCHSEL0	ADCR3	ADSTART	ADENDF	ADIE
FE59	0000 0000	R/W	ADMRC		–	ADMD4	ADMD3	ADMD2	ADMD1	ADMD0	ADMR2	ADTM1	ADTM0
FE5A	0000 0000	R/W	ADRLC		–	DATAL3	DATAL2	DATAL1	DATAL0	ADRL3	ADRL2	ADRL1	ADTM2
FE5B	0000 0000	R/W	ADRHC		–	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FE5C	0000 0000	R/W	P7	4-bit IO (7-4:DDR 3:0:DATA)	–	P73DDR	P72DDR	P71DDR	P70DDR	P73DT	P72DT	P71DT	P70DT
FE5D	0000 0000	R/W	I01CR		–	INT1LH	INT1LV	INT1IF	INT1IE	INT0LH	INT0LV	INT0IF	INT0IE

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE5E	0000 0000	R/W	I23CR		−	INT3HEG	INT3LEG	INT3IF	INT3IE	INT2HEG	INT2LEG	INT2IF	INT2IE
FE5F	0000 0000	R/W	ISL		−	STOHCP	STOLCP	BTIMC1	BTIMC0	BUZON	NFSEL	NFON	STOIN
FE60	HHHH HHHH		None										
FE61	HHHH HHHH		None										
FE62	HHHH HHHH		None										
FE63	HHHH HHHH		None										
FE64	HHHH HHHH		None										
FE65	HHHH HHHH		None										
FE66	HHHH HHHH		None										
FE67	HHHH HHHH		None										
FE68	HHHH HHHH		None										
FE69	HHHH HHHH		None										
FE6A	HHHH HHHH		None										
FE6B	HHHH HHHH		None										
FE6C	HHHH HHHH		None										
FE6D	HHHH HHHH		None										
FE6E	HHHH HHHH		None										
FE6F	HHHH HHHH		None										
FE70	HHHH HHHH		None										
FE71	HHHH HHHH		None										
FE72	HHHH HHHH		None										
FE73	HHHH HHHH		None										
FE74	HHHH HHHH		None										
FE75	HHHH HHHH		None										
FE76	HHHH HHHH		None										
FE77	HHHH HHHH		None										
FE78	0000 0000	R/W	T67CNT		−	T7C1	T7C0	T6C1	T6C0	T70V	T7IE	T60V	T6IE
FE79	0000 0000	R/W	WDTCNT		−	RSTFLG	FIX0	WDTRUN	IDLOP1	IDLOP0	WDTSL2	WDTSL1	WDTSL0
FE7A	0000 0000	R/W	T6R		−	T6R7	T6R6	T6R5	T6R4	T6R3	T6R2	T6R1	T6R0
FE7B	0000 0000	R/W	T7R		−	T7R7	T7R6	T7R5	T7R4	T7R3	T7R2	T7R1	T7R0
FE7C	HHHH HHHH		None										

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE7D	0000 0000	R/W	NKREG		–	NKEN	NKCMP2	NKCMP1	NKCMP0	NKCOV	NKCAP2	NKCAP1	NKCAP0
FE7E	0000 0000	R/W	FSR0	Flash control (bit 4 is R/O)	–	FSR0B7 Fix to 0	FSR0B6 Fix to 0	FSAERR	FSWOK	INTHIGH	FSLDAT	FSPGL	FSWREQ
FE7F	0000 0000	R/W	BTCR	Base timer control	–	BTFST	BTON	BTC11	BTC10	BTIF1	BTIE1	BTIF0	BTIE0
FE80	0000 0000	R/W	USCTRL		–	USBON	USBRUN	VD30EN	VD3KIL	IDLFG	IDLEN	DPIEZ	DMIEZ
FE81	0000 0000	R/W	USPORT		–	DDRSOF	VD30E2	USBSIO	SUSPND	DDRDP	DDRDM	PORTDP	PORTDM
FE82	0000 0000	R/W	USBINT		–	BRSFG	BRSEN	BACFG	BACEN	SOFFG	SOFEN	USBINT1	ENPEN
FE83	0000 0000	R/W	EP0INT		–	AK0FG	AK0EN	NK0FG	NK0EN	ER0FG	ER0EN	ST0FG	ST0EN
FE84	0000 0000	R/W	EP1INT		–	AK1FG	AK1EN	NK1FG	NK1EN	ER1FG	ER1EN	ST1FG	ST1EN
FE85	0000 0000	R/W	EP2INT		–	AK2FG	AK2EN	NK2FG	NK2EN	ER2FG	ER2EN	ST2FG	ST2EN
FE86	0000 0000	R/W	EP3INT		–	AK3FG	AK3EN	NK3FG	NK3EN	ER3FG	ER3EN	ST3FG	ST3EN
FE87	0000 0000	R/W	EP4INT		–	AK4FG	AK4EN	NK4FG	NK4EN	ER4FG	ER4EN	ST4FG	ST4EN
FE88	HHHH HHHH		None										
FE89	HHHH HHHH		None										
FE8A	0000 0000	R	FRAMEL		–	FRM07	FRM06	FRM05	FRM04	FRM03	FRM02	FRM01	FRM00
FE8B	HHHH H000	R	FRAMEH		–	–	–	–	–	–	FRM10	FRM09	FRM08
FE8C	0000 0000	R/W	USBADR		–	ADREN	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
FE8D	0000 0000	R/W	EPINF0		–	EPNO3	EPNO2	EPNO1	EPNO0	TKN1	TKN0	CTKN1	CTKN0
FE8E	0000 0000	R/W	EP0STA		–	E0EN	E0TGL	E0OVR	E0STL	E0ACK	E0CSU	E0CST	E0CRW
FE8F	H000 0000	R/W	EP0MP		–	–	E0MP6	E0MP5	E0MP4	E0MP3	E0MP2	E0MP1	E0MP0
FE90	H000 0000	R/W	EP0RX		–	–	E0RX6	E0RX5	E0RX4	E0RX3	E0RX2	E0RX1	E0RX0
FE91	H000 0000	R/W	EP0TX		–	–	E0TX6	E0TX5	E0TX4	E0TX3	E0TX2	E0TX1	E0TX0
FE92	0000 0000	R/W	EP1STA		–	E1EN	E1TGL	E1OVR	E1STL	E1ACK	E1DIR	E1IS0	E1BNK
FE93	0000 0000	R/W	EP2STA		–	E2EN	E2TGL	E2OVR	E2STL	E2ACK	E2DIR	E2IS0	E2BNK
FE94	0000 0000	R/W	EP3STA		–	E3EN	E3TGL	E3OVR	E3STL	E3ACK	E3DIR	E3IS0	E3BNK
FE95	0000 0000	R/W	EP4STA		–	E4EN	E4TGL	E4OVR	E4STL	E4ACK	E4DIR	E4IS0	E4BNK
FE96	HHHH HHHH		None										
FE97	HHHH HHHH		None										
FE98	H000 0000	R/W	EP1CNT		–	–	E1CN6	E1CN5	E1CN4	E1CN3	E1CN2	E1CN1	E1CN0
FE99	H000 0000	R/W	EP1RX		–	–	E1RX6	E1RX5	E1RX4	E1RX3	E1RX2	E1RX1	E1RX0
FE9A	H000 0000	R/W	EP2CNT		–	–	E2CN6	E2CN5	E2CN4	E2CN3	E2CN2	E2CN1	E2CN0
FE9B	H000 0000	R/W	EP2RX		–	–	E2RX6	E2RX5	E2RX4	E2RX3	E2RX2	E2RX1	E2RX0

AI-5

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE9C	H000 0000	R/W	EP3CNT		–	–	E3CN6	E3CN5	E3CN4	E3CN3	E3CN2	E3CN1	E3CN0
FE9D	H000 0000	R/W	EP3RX		–	–	E3RX6	E3RX5	E3RX4	E3RX3	E3RX2	E3RX1	E3RX0
FE9E	H000 0000	R/W	EP4CNT		–	–	E4CN6	E4CN5	E4CN4	E4CN3	E4CN2	E4CN1	E4CN0
FE9F	H000 0000	R/W	EP4RX		–	–	E4RX6	E4RX5	E4RX4	E4RX3	E4RX2	E4RX1	E4RX0
FEA0	0000 0000	R/W	DWC2B0		–	DWB07	DWB06	DWB05	DWB04	DWB03	DWB02	DWB01	DWB00
FEA1	0000 0000	R/W	DWC2B1		–	DWB15	DWB14	DWB13	DWB12	DWB11	DWB10	DWB09	DWB08
FEA2	0000 0000	R/W	DWC2B2		–	DWB23	DWB22	DWB21	DWB20	DWB19	DWB18	DWB17	DWB16
FEA3	0000 0000	R/W	DWC2B3		–	DWB31	DWB30	DWB29	DWB28	DWB27	DWB26	DWB25	DWB24
FEA4	XXXX XXXX	R	DWC2C0		–	DWC07	DWC06	DWC05	DWC04	DWC03	DWC02	DWC01	DWC00
FEA5	XXXX XXXX	R	DWC2C1		–	DWC15	DWC14	DWC13	DWC12	DWC11	DWC10	DWC09	DWC08
FEA6	XXXX XXXX	R	DWC2C2		–	DWC23	DWC22	DWC21	DWC20	DWC19	DWC18	DWC17	DWC16
FEA7	XXXX XXXX	R	DWC2C3		–	DWC31	DWC30	DWC29	DWC28	DWC27	DWC26	DWC25	DWC24
FEA8	HHHH HHHH		None										
FEA9	HHHH HHHH		None										
FEAA	HHHH HHHH		None										
FEAB	HHHH HHHH		None										
FEAC	0000 0000	R/W	TESTR0		–	DPLLTEST	CMPTST	CMPKIL	TTXCLK	TTXREQ	TADR2	TADR1	TADR0
FEAD	0000 0000	R	TESTR1		–	TDAT7	TDAT6	TDAT5	TDAT4	TDAT3	TDAT2	TDAT1	TDAT0
FEAE	0000 0000	R/W	EPADOF		–	E40F1	E40F0	E30F1	E30F0	E20F1	E20F0	E10F1	E10F0
FEAF	HHHH HHHH		None										
FEB0	HHHH HHHH		None										
FEB1	HHHH HHHH		None										
FEB2	HHHH HHHH		None										
FEB3	HHHH HHHH		None										
FEB4	HHHH HHHH		None										
FEB5	HHHH HHHH		None										
FEB6	HHHH HHHH		None										
FEB7	HHHH HHHH		None										
FEB8	HHHH HHHH		None										
FEB9	HHHH HHHH		None										
FEBA	HHHH HHHH		None										

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEBB	HHHH HHHH		None										
FEBC	HHHH HHHH		None										
FEBD	HHHH HHHH		None										
FEBE	HHHH HHHH		None										
FEBF	0000 0000	R/W	WPLLC		–	WPLLC7	WPLLC6	WPLLC5	WPLLC4	WPLLC3	WPLLC2	WPLLC1	WPLLC0
FEC0	0000 0000	R/W	PCMCNT		–	PCMCNT7	PCMCNT6	PCMCNT5	PCMCNT4	PCMCNT3	PCMCNT2	PCMCNT1	PCMCNT0
FEC1	00HH H000	R/W	PCMDIV		–	PCMDIV7	PCMDIV6	–	–	–	PCMDIV2	PCMDIV1	PCMDIV0
FEC2	HHHH HHHH		None										
FEC3	HHHH HHHH		None										
FEC4	HHHH HHHH		None										
FEC5	HHHH HHHH		None										
FEC6	HHHH HHHH		None										
FEC7	HHHH HHHH		None										
FEC8	HHHH HHHH		None										
FEC9	HHHH HHHH		None										
FECA	HHHH HHHH		None										
FECB	HHHH HHHH		None										
FECC	HHHH HHHH		None										
FECD	HHHH HHHH		None										
FECE	HHHH HHHH		None										
FECF	HHHH HHHH		None										
FED0	0000 0000	R/W	UCON0		–	UBRSEL	STRDET	RECRUN	STPERR	U0B3	RBIT8	RECEND	RECIE
FED1	0000 0000	R/W	UCON1		–	TRUN	8/9BIT	TDDR	TCMOS	8/7BIT	TBIT8	TEPTY	TRNSIE
FED2	0000 0000	R/W	UBR		–	UBRG7	UBRG6	UBRG5	UBRG4	UBRG3	UBRG2	UBRG1	UBRG0
FED3	0000 0000	R/W	TBUF		–	T1BUF7	T1BUF6	T1BUF5	T1BUF4	T1BUF3	T1BUF2	T1BUF1	T1BUF0
FED4	0000 0000	R/W	RBUF		–	R1BUF7	R1BUF6	R1BUF5	R1BUF4	R1BUF3	R1BUF2	R1BUF1	R1BUF0
FED5	HHHH HHHH		None										
FED6	0000 0000	R/W	S4ADRL		–	S4ADL7	S4ADL6	S4ADL5	S4ADL4	S4ADL3	S4ADL2	S4ADL1	S4ADL0
FED7	0000 0000	R/W	S4BYTH		–	S4STPWD	S4BYTRD	S4BYTH5	S4BYTH4	S4BYTH3	S4BYTH2	S4BYTH1	S4BYTH0
FED8	0000 0000	R/W	CRCL		–	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
FED9	0000 0000	R/W	CRCH		–	CRC15	CRC14	CRC13	CRC12	CRC11	CRC10	CRC9	CRC8

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEDA	0000 0000	R/W	CRCCNT		–	CRCON	CRCLRZ	CRCRD	1/0SEL	S4STPCEN	S4STPCHI	S4STPSL1	S4STPSL0
FEDB	0000 0000	R/W	SI4CN0		–	SI4RUN	SBITON	MSBSEL	S4RAM	S4CKPL	SI4WRT	SI4END	SI4IE
FEDC	0000 0000	R/W	SI4CN1		–	PARA	P1/P0	P22/P23	P24OUT	P23MOS	P23OUT	P22MOS	P22OUT
FEDD	0000 0000	R/W	SI4BUF		–	S4BUF7	S4BUF6	S4BUF5	S4BUF4	S4BUF3	S4BUF2	S4BUF1	S4BUF0
FEDE	0000 0000	R/W	S4BAUD		–	S4BAU7	S4BAU6	S4BAU5	S4BAU4	S4BAU3	S4BAU2	S4BAU1	S4BAU0
FEDF	0000 0000	R/W	S4ADDR		–	S4WSTP	S4PTSEL	S4ADR5	S4ADR4	S4ADR3	S4ADR2	S4ADR1	S4ADR0
FEE0	0000 0000	R/W	S4BYTE		–	S4BYT7	S4BYT6	S4BYT5	S4BYT4	S4BYT3	S4BYT2	S4BYT1	S4BYT0
FEE1	HHH0 0000	R/W	DWC2CT		–	–	–	–	DWCRUN	DWCOV	DWCCY	DWCMD1	DWCMD0
FEE2	0000 0000	R/W	DWC2A0		–	DWA07	DWA06	DWA05	DWA04	DWA03	DWA02	DWA01	DWA00
FEE3	0000 0000	R/W	DWC2A1		–	DWA15	DWA14	DWA13	DWA12	DWA11	DWA10	DWA09	DWA08
FEE4	0000 0000	R/W	DWC2A2		–	DWA23	DWA22	DWA21	DWA20	DWA19	DWA18	DWA17	DWA16
FEE5	0000 0000	R/W	DWC2A3		–	DWA31	DWA30	DWA29	DWA28	DWA27	DWA26	DWA25	DWA24
FEE6	HHHH HHHH		None										
FEE7	HHHH HHHH		None										
FEE8	0000 0000	R/W	UCON2		–	UBRSEL2	STRDET2	RECRUN2	STPERR2	U2B3	RBIT82	RECEND2	RECIE2
FEE9	0000 0000	R/W	UCON3		–	TRUN2	8/9BIT2	TDDR2	TCOMS2	8/7BIT2	TBIT82	TEPTY2	TRNSIE2
FEEA	0000 0000	R/W	UBR2		–	U2BRG7	U2BRG6	U2BRG5	U2BRG4	U2BRG3	U2BRG2	U2BRG1	U2BRG0
FEEB	0000 0000	R/W	TBUF2		–	T2BUF7	T2BUF6	T2BUF5	T2BUF4	T2BUF3	T2BUF2	T2BUF1	T2BUF0
FEEC	0000 0000	R/W	RBUF2		–	R2BUF7	R2BUF6	R2BUF5	R2BUF4	R2BUF3	R2BUF2	R2BUF1	R2BUF0
FEED	HHHH HHHH		None										
FEEE	HHHH HHHH		None										
FEEF	HHHH HHHH		None										
FEF0	HHHH HHHH		None										
FEF1	HHHH HHHH		None										
FEF2	HHHH HHHH		None										
FEF3	HHHH HHHH		None										
FEF4	HHHH HHHH		None										
FEF5	HHHH HHHH		None										
FEF6	HHHH HHHH		None										
FEF7	HHHH HHHH		None										
FEF8	HHHH HHHH		None										
FEF9	HHHH HHHH		None										

Address	Initial value	R/W	LC871D00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEFA	HHHH HHHH		None										
FEFB	HHHH HHHH		None										
FEFC	HHHH HHHH		None										
FEFD	HHHH HHHH		None										
FEFE	HHHH HHHH		None										
FEFF	HHHH HHHH		None										

Port 0 Block Diagram

Option: Output type (CMOS or N-channel OD) selectable in 1-bit units

Port	Multiplexed port pin function
P07	Timer 7 toggle output
P06	Timer 6 toggle output
P05	Clock output (system/subclock selectable)

Pull-up resistor

Option Nch-OD selected : None

Option CMOS selected   : Programmable

Port block diagram

Port	Special Input	Function Output
P17	None	Timer 1HPWM output
P16	None	Timer 1LPWM output
P15	SIO1clock input	SIO1 clock output
P14	SIO1data input	SIO1 data output
P13	None	SIO1 data output
P12	SIO0 clock input	SIO0 clock output
P11	SIO0 data input	SIO0 data output
P10	None	SIO0 data output

Table of Port 1 Multiplexed Functions
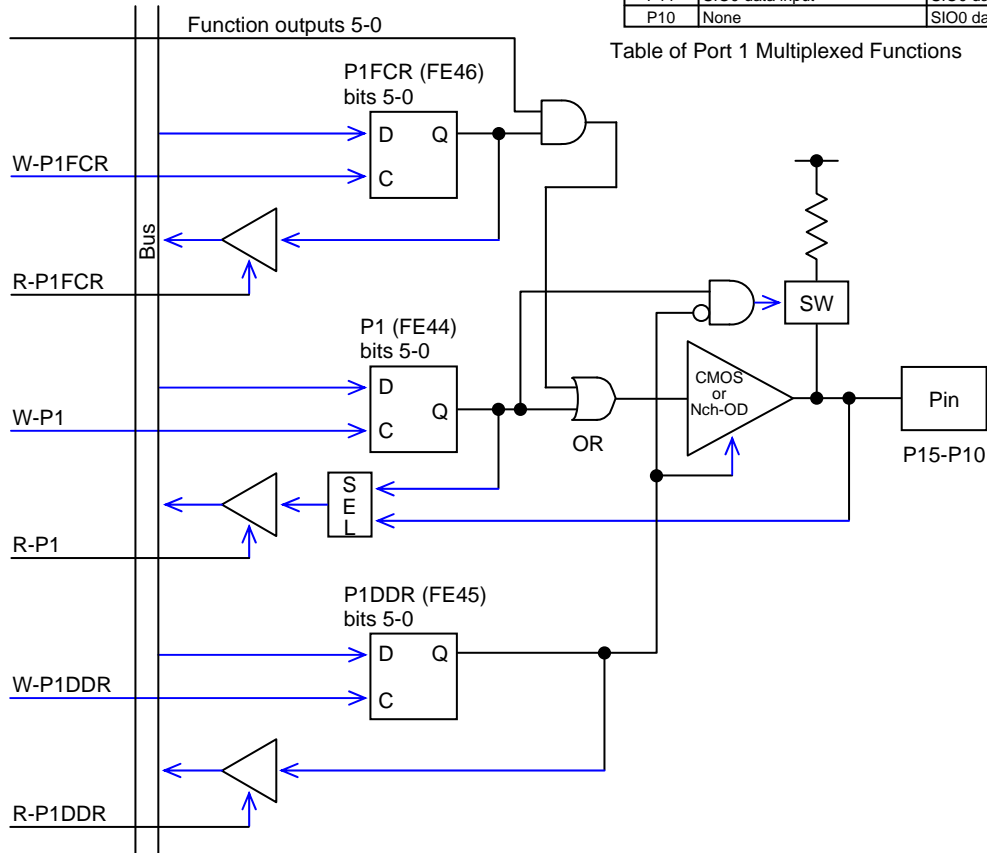
Port 1 Block Diagram

Option: Output type (CMOS or N-channel OD) selectable in 1-bit units

AII-2

| Port	Special input	Function output
P27	None	None
P26	UART2 data input	None
P25	None	UART2 data output
P24	SIO4 clock input	SIO4 clock output
P23	SIO4 data input	SIO4 data output
P22	SIO4 data input	SIO4 data output
P21	UART1 data input	None
P20	None	UART1 data output
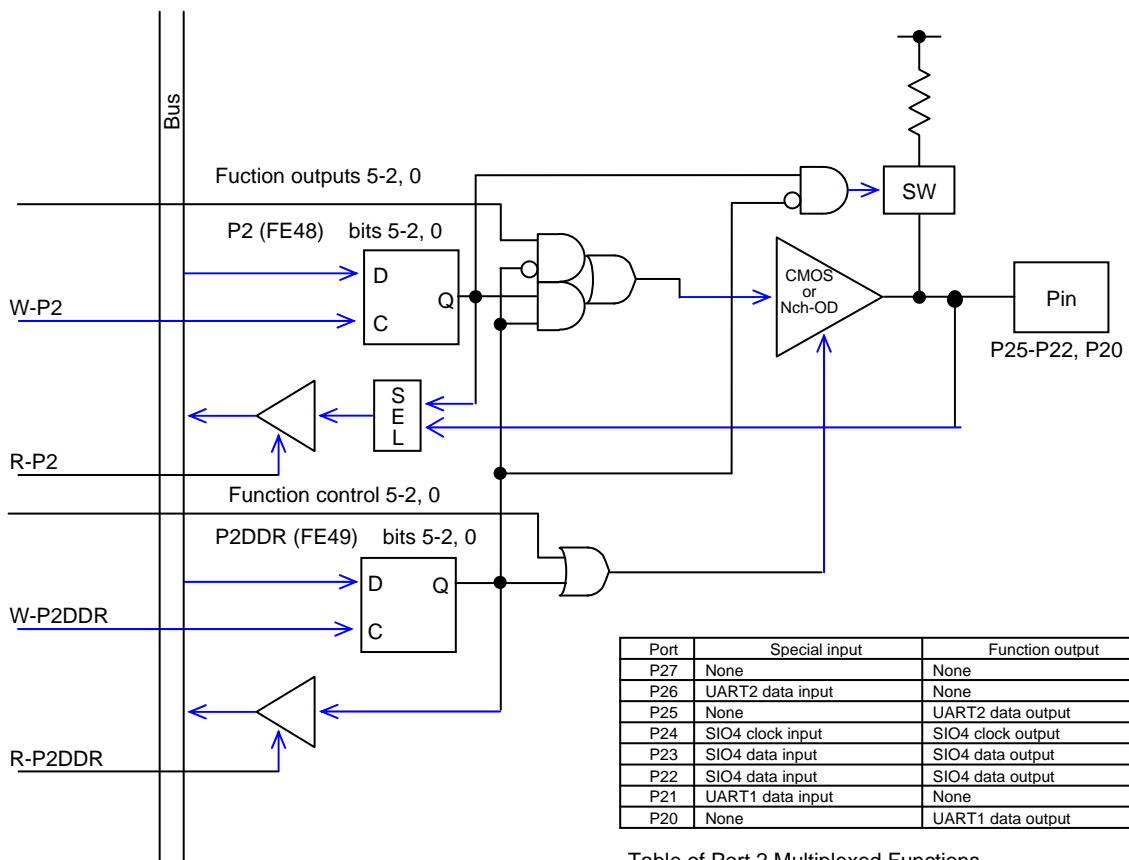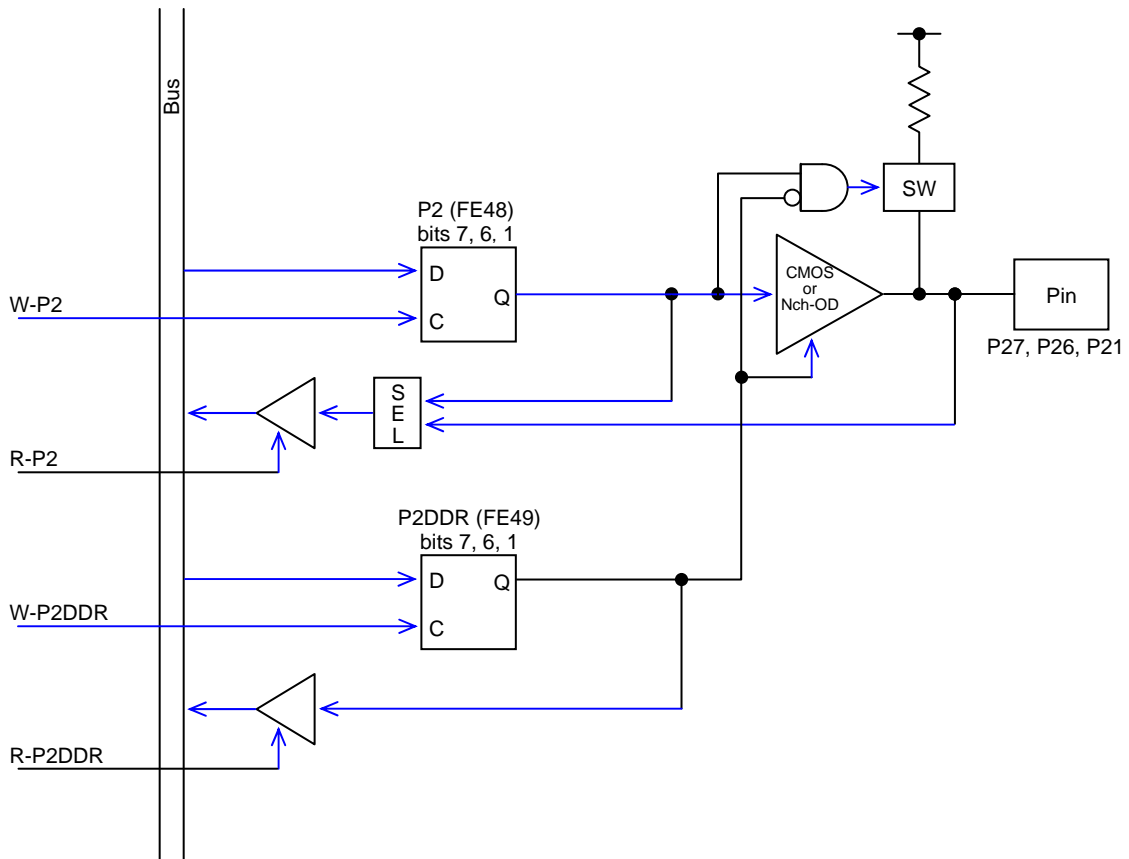
Table of Port 2 Multiplexed Functions

Port 2(Pin) Block Diagram

Option: Output type (CMOS or N-channel OD) selectable in 1-bit units

AII-3

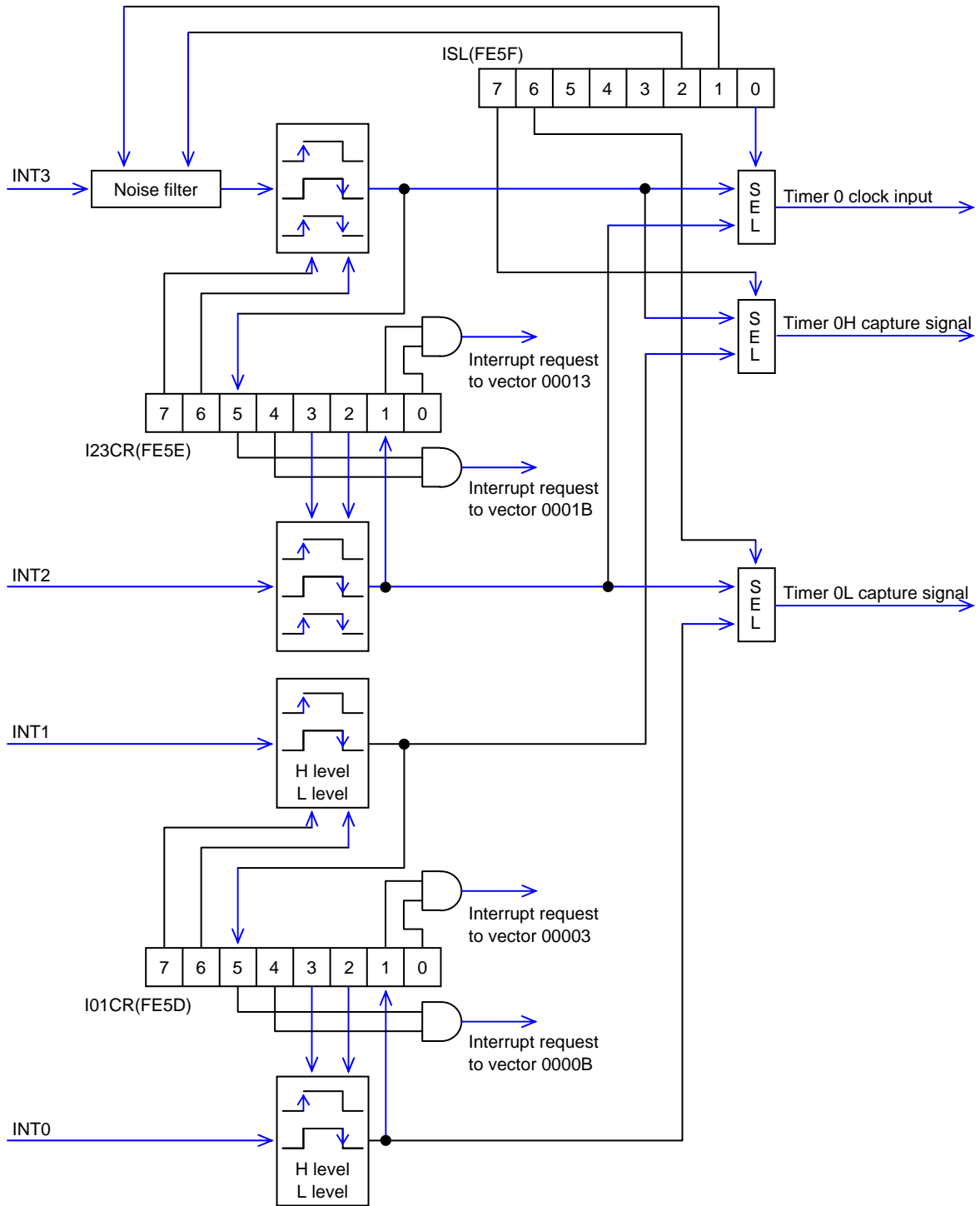## Port block diagram

Port 2 (Interrupt) Block Diagram
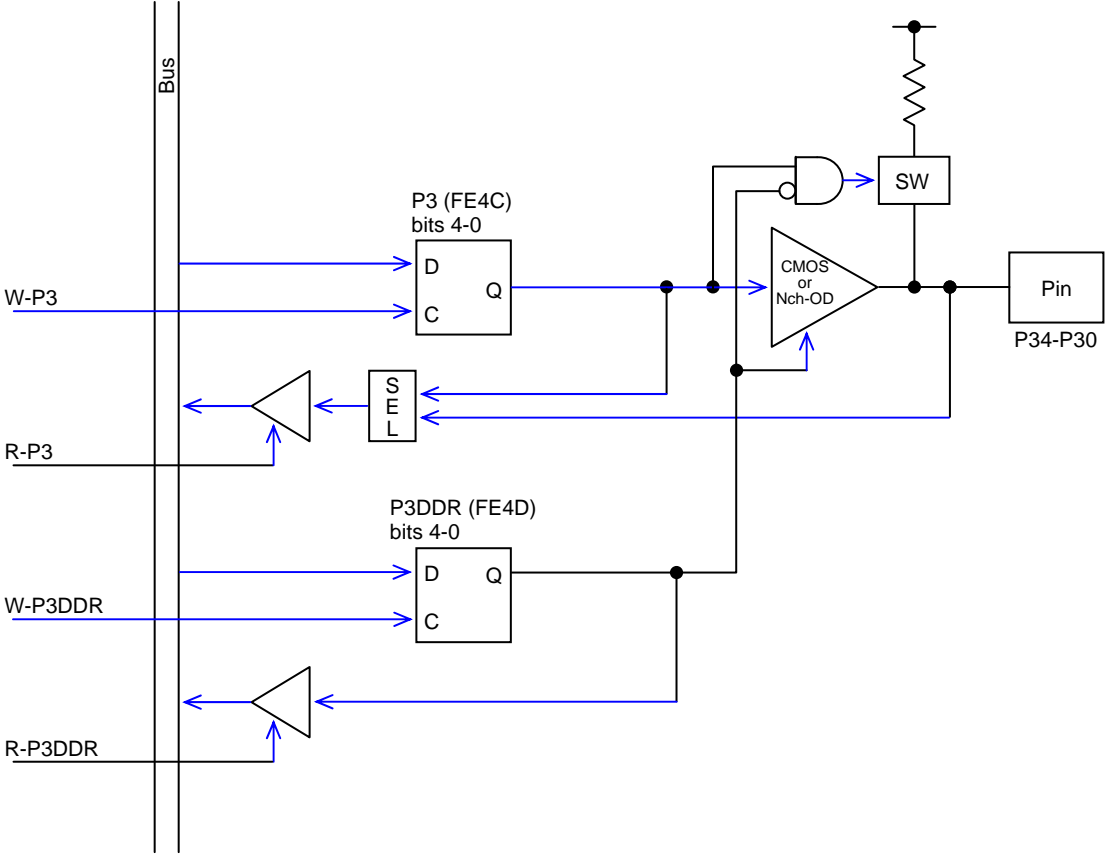
Port 3 Block Diagram
Option: Output type (CMOS or N-channel OD) selectable in 1-bit units
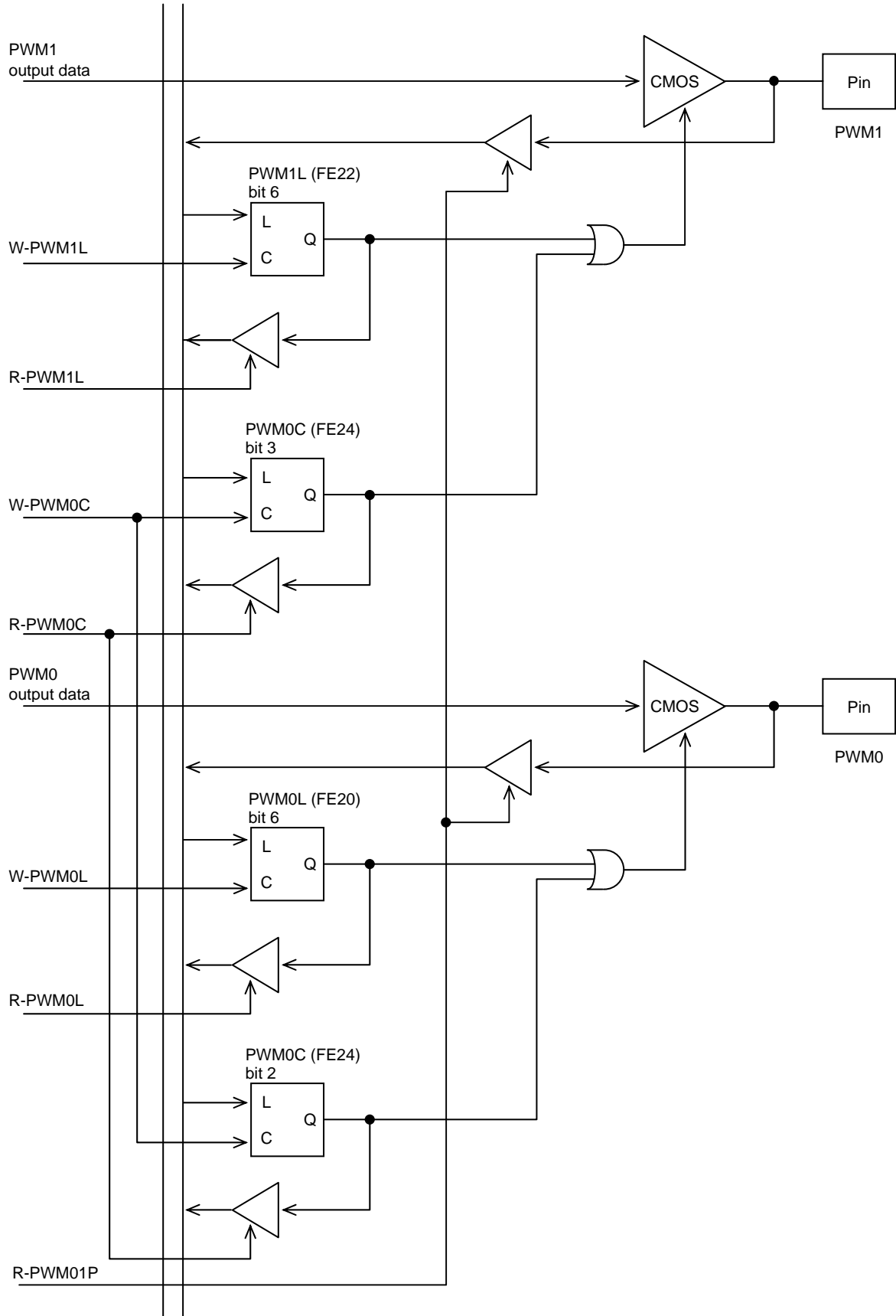
Port block diagram
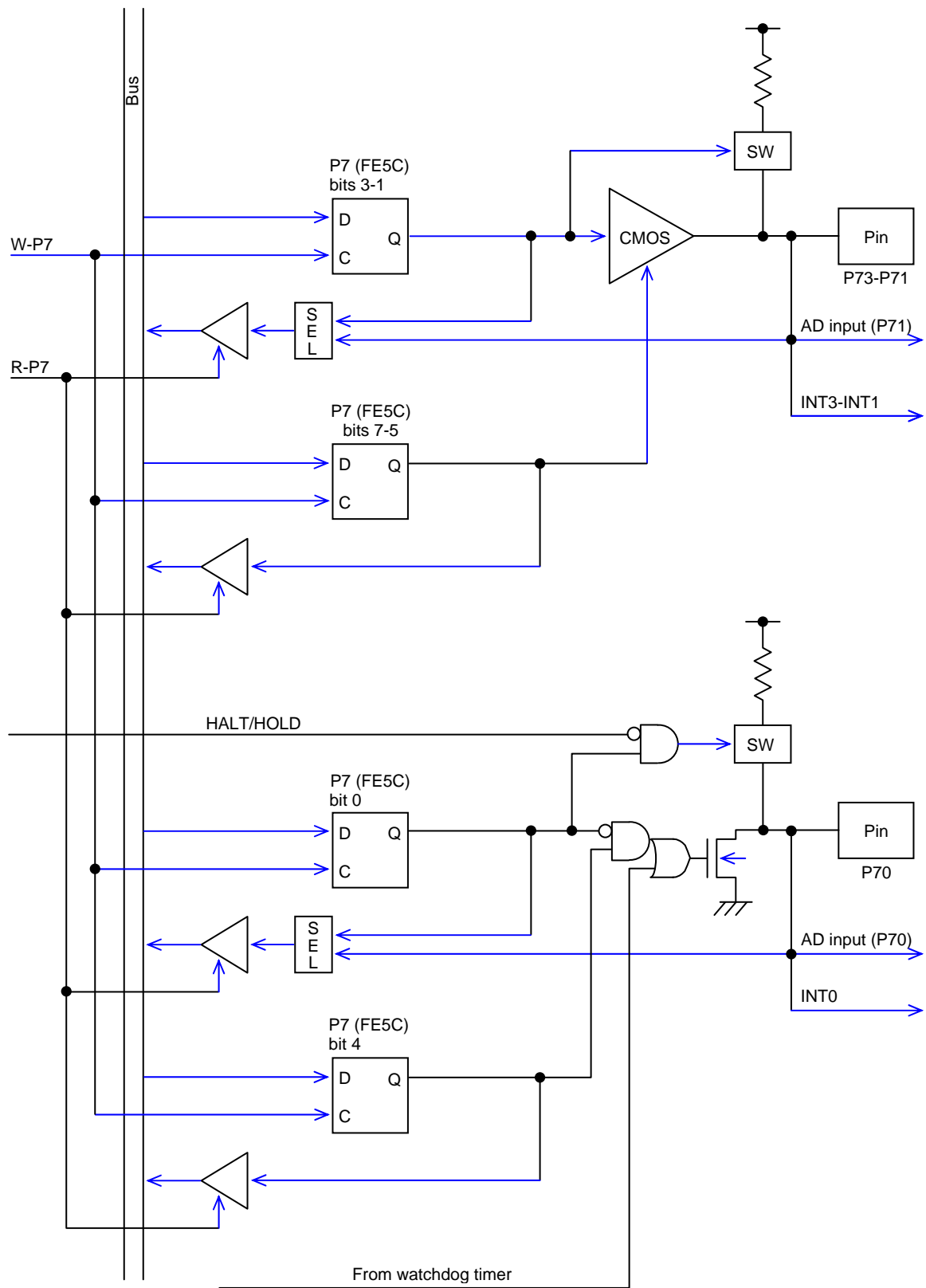
Port 7 (Pin) Block Diagram
Option: None

Port 7 (Interrupt) Block Diagram

Port block diagram

PWM0, PWM1 Block Diagram
Option: None

Important Note

This document is designed to provide the reader with accurate information in easily understandable form regarding the device features and the correct device implementation procedures.

The sample configurations included in the various descriptions are intended for reference only and should not be directly incorporated in user product configurations.

ON Semiconductor shall bear no responsibility for obligations concerning patent infringements, safety or other legal disputes arising from prototypes or actual products created using the information contained herein.