ON Semiconductor

Is Now

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

CMOS 8-BIT MICROCONTROLLER

LC871K00 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

Appendixes

1. Overview

1.1 Overview

The LC871K00 series is an 8-bit microcontroller that, centered around a CPU running at a minimum bus cycle time of 83.3 ns, integrates on a single chip a number of hardware features such as 64K-byte flash ROM (onboard programmable), 8192-byte RAM, an on-chip debugger, a sophisticated 16-bit timer/ counter (may be divided into 8-bit timers), 16-bit timer (may be divided into 8-bit timers or 8-bit PWMs), four 8-bit timers with a prescaler, a base timer serving as a time-of-day clock, two synchronous SIO interfaces (with an automatic transfer function), an asynchronous/synchronous SIO interface, a single master I^2C/synchronous SIO interface, a UART interface (full duplex), two full-/low-speed USB interfaces (with host control function), a 12-bit 12-channel AD converter, two 12-bit PWM channels, a system clock frequency divider, an infrared remote control receive function, an internal reset circuit, and 44-source 10-vector interrupt function.

1.2 Features

● Flash ROM
- Capable of onboard programming with a wide supply voltage range of 3.0 to 5.5V
- 128-byte block erase
- Writing in two-byte units
- 65536 × 8 bits

● RAM
- 8192 × 9 bits

● Bus cycle time
- 83.3 ns (at CF=12 MHz)

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

● Minimum instruction cycle time (Tcyc)
- 250 ns (at CF=12 MHz)

● Ports
- I/O ports

  Ports whose input/output can be specified in 1-bit units: 34 (P00 to P07, P10 to P17, P20 to P25, P30 to P34, P70 to P73, PWM0, PWM1, XT2)
- USB ports 4 (UHAD+, UHAD-, UHBD+, UHBD-)
- 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 two capture registers

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

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

  Mode 2: 16-bit timer with an 8-bit programmable prescaler (with two 16-bit capture registers)

  Mode 3: 16-bit counter (with two 16-bit capture registers)

- 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 low-order 8 bits)

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

- Timer 4: 8-bit timer with a 6-bit prescaler
- Timer 5: 8-bit timer with a 6-bit prescaler
- 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 the subclock (32.768 kHz crystal oscillation), low-speed RC oscillator, system clock, and timer 0 prescaler output.
  2) Interrupts can be generated at five specified time intervals.

● Serial interface

- SIO0: Synchronous serial interface

  1) LSB first/MSB first selectable
  2) Transfer clock cycle: 4/3 to 512/3 Tcyc
  3) Automatic continuous data communication (1 to 256 bits can be specified 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 clock)

  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 clock)

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

- SIO4: Synchronous serial interface

  1) LSB first/MSB first selectable
  2) Transfer clock cycle: 4/3 to 1020/3 Tcyc
  3) Automatic continuous data communication (1 to 8192 bytes can be specified in 1-byte units) (Suspension and resumption of data transfer possible in 1-byte or word units)
  4) Clock polarity can be selected.
  5) Built-in CRC16 computation circuit

- SMIIC0: Single master I^2C/synchronous 8-bit serial I/O

  Mode 0: Single-master master mode communication

  Mode 1: Synchronous 8-bit serial I/O (data MSB first)

- Full duplex UART
  1) Data length: 7/8/9 bits
  2) Stop bits: 1 bit (2 bits in continuous transmission mode)
  3) Parity bit: None/even parity/odd parity (data length 8 bits only)
  4) Baudrate: 16/3 to 8192/3Tcyc

- AD converter: 12 bits × 12 channels

- PWM: Multifrequency 12-bit PWM × 2 channels

- Infrared remote control receiver circuit
  1) Noise rejection function

     (Noise rejection filter time constant: approx. 120 μs when selecting a 32.768 kHz crystal oscillator as a reference clock.)
  2) Supports data encoding formats, including PPM (Pulse Position Modulation), Manchester encoding, and so on.
  3) X'tal HOLD mode release function

- USB interface (with host control function) × 2 ports
  - Supports full-speed (12 Mbps) and low-speed (1.5 Mbps).
  - Supports four types of transfer (control transfer, bulk transfer, interrupt transfer, and isochronous transfer)

- Audio interface
  1) Sampling frequencies (fs): 8 kHz/11.025 kHz/12 kHz/16 kHz/22.05 kHz/24 kHz/32 kHz/44.1 kHz/48 kHz
  2) Master clock: 256 fs/384 fs
  3) Bit clock: 48 fs/64 fs
  4) Data bit length: 16/18/20/24 bits
  5) LSB first/MSB first selectable
  6) Left-justified/right-justified/I^2S format selectable

- Watchdog timer
  - External RC time constant type
    1) Interrupt generation/reset generation can be selected
    2) WDT operation on entry into HALT or HOLD mode can be selected from operation continue mode or operation stop mode.
  - Internal timer type
    1) Capable of generating an internal reset on an overflow of a timer running on the low-speed RC oscillator clock or the subclock.
    2) WDT operation on entry into HALT or HOLD mode can be selected from three modes (operation continue, operation stop, and operation stop while retaining the count value).

- Clock output function
  1) Capable of generating a clock with a frequency of 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, or 1/64 of the source oscillator clock selected as the system clock.
  2) Capable of generating the source oscillator clock for the subclock.

- Interrupts
  - 44 sources, 10 vector addresses
    1) 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 with the lowest vector address has 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/UHC-A bus active/UHC-B bus active/remote control receive
4	0001BH	H or L	INT3/INT5/base timer
5	00023H	H or L	T0H/ INT6/UHC-A device connection/UHC-A device disconnection/UHC-A resume
6	0002BH	H or L	T1L/T1H/INT7/AIF start/SMIIC0/UHC-B device connection/UHC-B device disconnection/ UHC-B resume
7	00033H	H or L	SIO0/UART1 receive end
8	0003BH	H or L	SIO1/SIO4/UART1 buffer empty/UART1 transmit end/AIF end
9	00043H	H or L	ADC/T6/T7/UHC-ACK/UHC-NAK/UHC error/UHC-STALL
10	0004BH	H or L	Port 0/PWM0/PWM1/T4/T5/UHC-SOF

- Priority levels: X > H > L
- When interrupts of the same level occur at the same time, the interrupt with the lowest vector address is processed first.

● Subroutine stack levels: Up to 4096 levels (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
- Medium-speed RC oscillator circuit (internal): For system clock (approx. 1 MHz)
- Low-speed RC oscillator circuit (internal): For system clock, timers, and watchdog timer (approx. 30 kHz)
- CF oscillator circuit: For system clock
- Crystal oscillator circuit: For system clock and time-of-day clock
- PLL circuit (internal): For USB interface and audio interface

● Internal reset circuit
- Power-on reset (POR) function
  1) POR is generated only when power is turned on.
  2) The POR release level can be selected from 8 levels (1.67, 1.97, 2.07, 2.37, 2.57V, 2.87V, 3.86V, and 4.35V) by setting options.
- Low-voltage detection reset (LVD) function
  1) LVD and POR functions are combined to generate resets when power is turned on and when the power voltage falls below a certain level.
  2) The use/non-use of the LVD function and the low voltage detection level can be selected from 7 levels (1.91, 2.01, 2.31, 2.51, 2.81V, 3.79V, and 4.28V) by setting options.

● Standby function

- HALT mode: Halts instruction execution while allowing the peripheral circuits to continue operation.
  1) Oscillators do not stop automatically.
  2) There are three ways of releasing HALT mode.
     <1> Low level input to the reset pin
     <2> Generating a reset by the watchdog timer or low-voltage detection
     <3> Generating an interrupt
- HOLD mode: Suspends instruction execution and the operation of the peripheral circuits.
  1) The PLL, CF, RC, and crystal oscillators automatically stop operation.

  Note: The low-speed RC oscillation is also controlled directly by the watchdog timer. Its oscillation in the standby mode is also controlled by the watchdog timer.

  2) There are five ways of releasing HOLD mode.
     <1> Low level input to the reset pin
     <2> Generating a reset by the watchdog timer or low-voltage detection
     <3> Establishing an interrupt source at either of 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 host controller circuit.
- X'tal HOLD mode: Suspends instruction execution and the operation of the peripheral circuits except the base timer and infrared remote control receiver circuit.
  1) The PLL, CF and RC oscillators automatically stop operation.

  Note: The low-speed RC oscillation is also controlled directly by the watchdog timer. Its oscillation in the standby mode is also controlled by the watchdog timer.

  Note: In an environment in which the base timer is run with its input clock source set to the low-speed RC oscillator and X'tal HOLD mode is entered, the low-speed RC oscillator retains the state that is established when this mode is entered.

  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> Low level input to the reset pin
     <2> Generating a reset by the watchdog timer or low-voltage detection.
     <3> Establishing an interrupt source at either of 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 host controller circuit
     <7> Establishing an interrupt source in the infrared remote control receiver circuit

● Package form

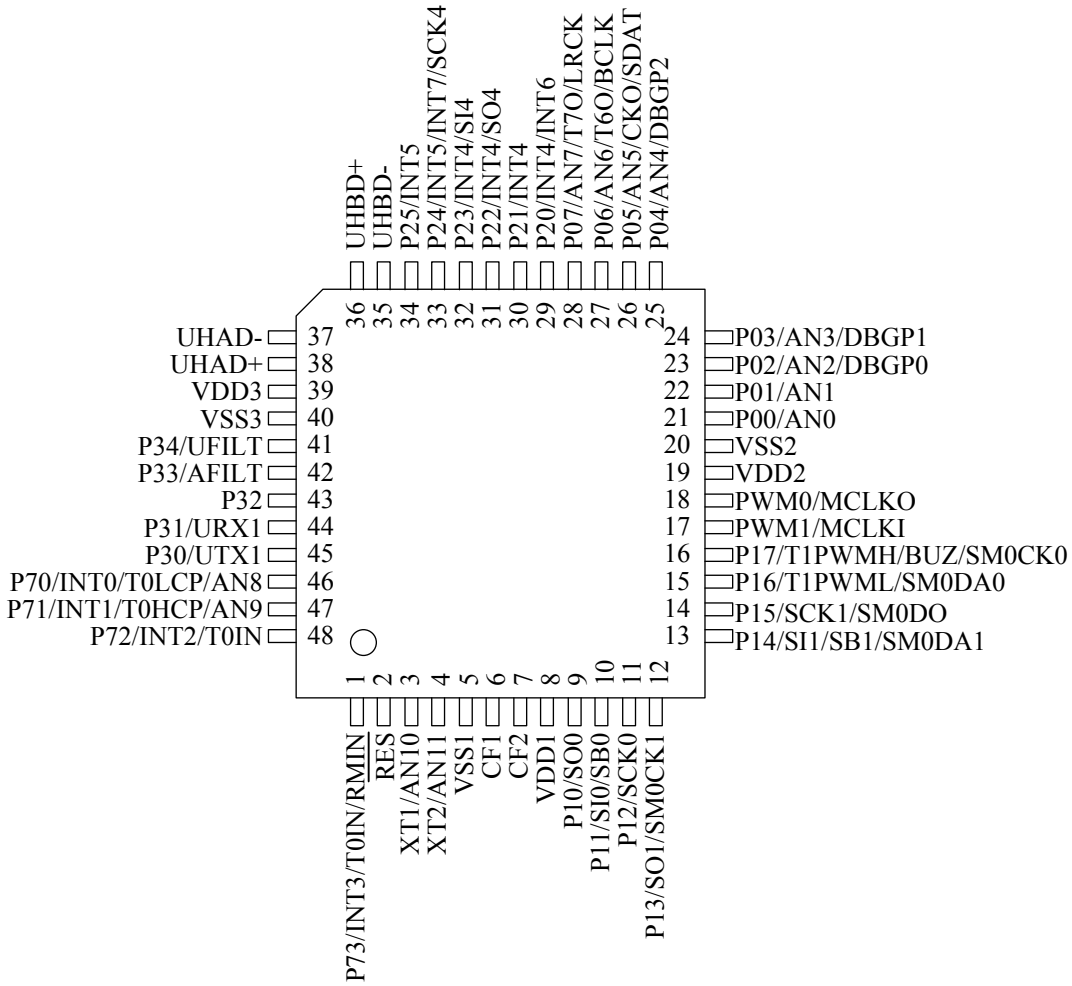
- SQFP48 (7×7) (Lead-free and halogen-free product)

● Development tools

- On-chip debugger: TCB87-Type B + LC87F1K64A or
  TCB87-Type C (3-wire communication cable) + LC87F1K64A

1.3 Pinout

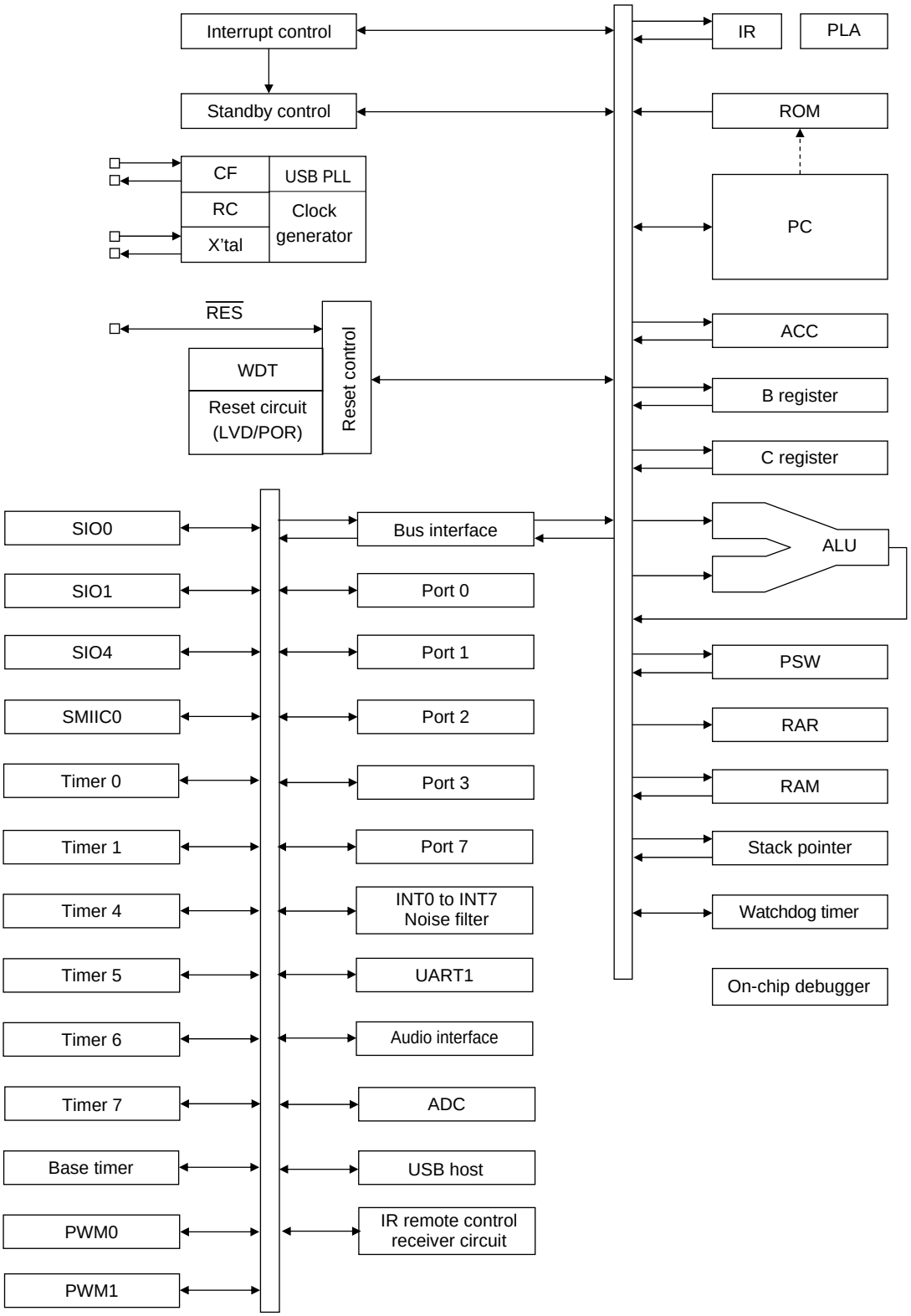
SQFP48 (7×7)



SQFP48	NAME
1	P73/INT3/T0IN/RMIN
2	$\overline{\text{RES}}$
3	XT1/AN10
4	XT2/AN11
5	VSS1
6	CF1
7	CF2
8	VDD1
9	P10/SO0
10	P11/SI0/SB0
11	P12/SCK0
12	P13/SO1/SM0CK1
13	P14/SI1/SB1/SM0DA1
14	P15/SCK1/SM0DO
15	P16/T1PWML/SM0DA0
16	P17/T1PWMH/BUZ/SM0CK0
17	PWM1/MCLKI
18	PWM0/MCLKO
19	VDD2
20	VSS2
21	P00/AN0
22	P01/AN1
23	P02/AN2/DBGP0
24	P03/AN3/DBGP1

SQFP48	NAME
25	P04/AN4/DBGP2
26	P05/AN5/CKO/SDAT
27	P06/AN6/T6O/BCLK
28	P07/AN7/T7O/LRCK
29	P20/INT4/INT6
30	P21/INT4
31	P22/INT4/SO4
32	P23/INT4/SI4
33	P24/INT5/INT7/SCK4
34	P25/INT5
35	UHBD-
36	UHBD+
37	UHAD-
38	UHAD+
39	VDD3
40	VSS3
41	P34/UFILT
42	P33/AFILT
43	P32
44	P31/URX1
45	P30/UTX1
46	P70/INT0/T0LCP/AN8
47	P71/INT1/T0HCP/AN9
48	P72/INT2/T0IN

1.4 System Block Diagram



1.5 Pin Functions

<table>
<tr><th>Name</th><th>I/O</th><th>Description</th><th>Option</th></tr>
<tr><td>VSS1, VSS2, VSS3</td><td>−</td><td>Power supply pin (-)</td><td>No</td></tr>
<tr><td>VDD1,VDD2</td><td>−</td><td>Power supply pin (+)</td><td>No</td></tr>
<tr><td>VDD3</td><td>−</td><td>USB reference power supply pin</td><td>Yes</td></tr>
<tr><td>Port 0
P00 to P07</td><td>I/O</td><td>• 8-bit I/O port
• I/O can be specified in 1-bit units
• Pull-up resistors can be turned on and off in 1-bit units
• HOLD release input
• Port 0 interrupt input
• Pin functions
AD converter input port: AN0 to AN7 (P00 to P07)
On-chip debugger pin: DBGP0 to DBGP2 (P02 to P04)
P05: System clock output/audio interface SDAT I/O
P06: Timer 6 toggle output/audio interface BCLK I/O
P07: Timer 7 toggle output/audio interface LRCK I/O</td><td>Yes</td></tr>
<tr><td>Port 1
P10 to P17</td><td>I/O</td><td>• 8-bit I/O port
• I/O can be specified 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/SMIIC0 clock I/O
P14: SIO1 data input/ bus I/O / SMIIC0 bus I/O / data input
P15: SIO1 clock I/O /SMIIC0 data output (3-wire SIO mode)
P16: Timer 1 PWML output/SMIIC0 bus I/O / data input
P17: Timer 1 PWMH output/buzzer output/SMIIC0 clock I/O</td><td>Yes</td></tr>
<tr><td>Port 2
P20 to P25</td><td>I/O</td><td>• 6-bit I/O port
• I/O can be specified 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 P25: INT5 input/HOLD release input/timer 1 event input/timer 0L capture input/timer 0H capture input
P20: INT6 input/ timer 0L capture 1 input
P22: SIO4 data I/O
P23: SIO4 data I/O
P24: INT7 input/timer 0H capture 1 input/SIO4 clock I/O
Interrupt detection mode
<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>INT4</td><td>○</td><td>○</td><td>○</td><td>×</td><td>×</td></tr>
<tr><td>INT5</td><td>○</td><td>○</td><td>○</td><td>×</td><td>×</td></tr>
<tr><td>INT6</td><td>○</td><td>○</td><td>○</td><td>×</td><td>×</td></tr>
<tr><td>INT7</td><td>○</td><td>○</td><td>○</td><td>×</td><td>×</td></tr>
</table></td><td>Yes</td></tr>
</table>

Continued on next page

Continued from preceding page

<table>
<tr><th>Name</th><th>I/O</th><th>Description</th><th>Option</th></tr>
<tr><td>Port 3
P30 to P34</td><td>I/O</td><td>• 5-bit I/O port
• I/O can be specified in 1-bit units
• Pull-up resistors can be turned on and off in 1-bit units
• Pin functions
P30: UART1 transmit
P31: Uart1 receive
P33: Audio interface PLL filter circuit connection pin
P34: USB interface PLL filter circuit connection pin</td><td>Yes</td></tr>
<tr><td>Port 7
P70 to P73</td><td>I/O</td><td>• 4-bit I/O port
• I/O can be specified 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
P71: INT1 input/HOLD release input/timer 0H 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 receive input
AD converter input port: AN8 (P70), AN9 (P71)
Interrupt detection mode
<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></td><td>No</td></tr>
<tr><td>PWM0
PWM1</td><td>I/O</td><td>• PWM0 and PWM1 output port
• General-purpose input port
• Pin functions
PWM0: Audio interface master clock output
PWM1: Audio interface master clock input</td><td>No</td></tr>
<tr><td>UHAD-
UHAD+</td><td>I/O</td><td>USB-A port data I/O pin /general-purpose I/O port</td><td>No</td></tr>
<tr><td>UHBD-
UHBD+</td><td>I/O</td><td>USB-B port data I/O pin /general-purpose I/O port</td><td>No</td></tr>
<tr><td>$\overline{\text{RES}}$</td><td>I/O</td><td>External reset input/internal reset output pin</td><td>No</td></tr>
<tr><td>XT1</td><td>I</td><td>• 32.768 kHz crystal resonator input pin
• Pin functions
General-purpose input port
AD converter input port: AN10</td><td>No</td></tr>
<tr><td>XT2</td><td>I/O</td><td>• 32.768 kHz crystal resonator output pin
• Pin functions
General-purpose I/O port
AD converter input port: AN11</td><td>No</td></tr>
<tr><td>CF1</td><td>I</td><td>Ceramic/crystal resonator input pin</td><td>No</td></tr>
<tr><td>CF2</td><td>O</td><td>Ceramic/crystal resonator output pin</td><td>No</td></tr>
</table>

1.6 On-chip Debugger Pin Connection Requirements

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

1.7 Recommended Unused Pin Connections

Pin	Recommended Unused Pin Connections	
	Board	Software
P00 to P03, P05 to P07	Open	Output low
P04	Pulled down with a resistor of 100kΩ	-
P10 to P17	Open	Output low
P20 to P25	Open	Output low
P30 to P34	Open	Output low
P70 to P73	Open	Output low
PWM0, PWM1	Open	Output low
UHAD+, UHAD-	Open	Output low
UHBD+, UHBD-	Open	Output low
XT1	Pulled down with a resistor of 100kΩ or less	-
XT2	Open	Output low

Note: Since P34 is multiplexed with UFILT, it must be configured for input when using the USB function.

Since P33 is multiplexed with AFILT, it must be configured for input when using the PLL circuit for the audio interface.

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	Option Selected in Units of	Option Type	Output Type	Pull-up Resistor
P00 to P07 P10 to P17 P20 to P25 P30 to P34	1 bit	1	CMOS	Programmable
		2	N-channel open drain	Programmable
P70	−	No	N-channel open drain	Programmable
P71 to P73	−	No	CMOS	Programmable
PWM0, PWM1	−	No	CMOS	No
UHAD+, UHAD- UHBD+, OHBD-	−	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

1.9 User Option Table

Option	Option to be Applied on	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 P25	○	1 bit	CMOS
				N-channel open drain
	P30 to P34	○	1 bit	CMOS
				N-channel open drain
Program start address	–	○	–	00000h
				0FE00h
USB regulator	USB regulator	○	–	Use
				Non-use
	USB regulator (HOLD mode)	○	–	Use
				Non-use
	USB regulator (HALT mode)	○	–	Use
				Non-use
Main clock 8 MHz select	–	○	–	Enable
				Disable
Low-voltage detection reset function	Detection function	○	–	Enable: Use
				Disable: Non-use
	Detection level	○	–	7 levels
Power-on reset function	Power-on reset level	○	–	8 levels

1.10 USB Reference Power Supply 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 setting	USB regulator	Use	Use	Use	Non-use
	USB regulator in HOLD mode	Use	Non-use	Non-use	Non-use
	USB regulator in HALT mode	Use	Non-use	Use	Non-use
Reference voltage circuit operation	Normal operating mode	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 level.

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

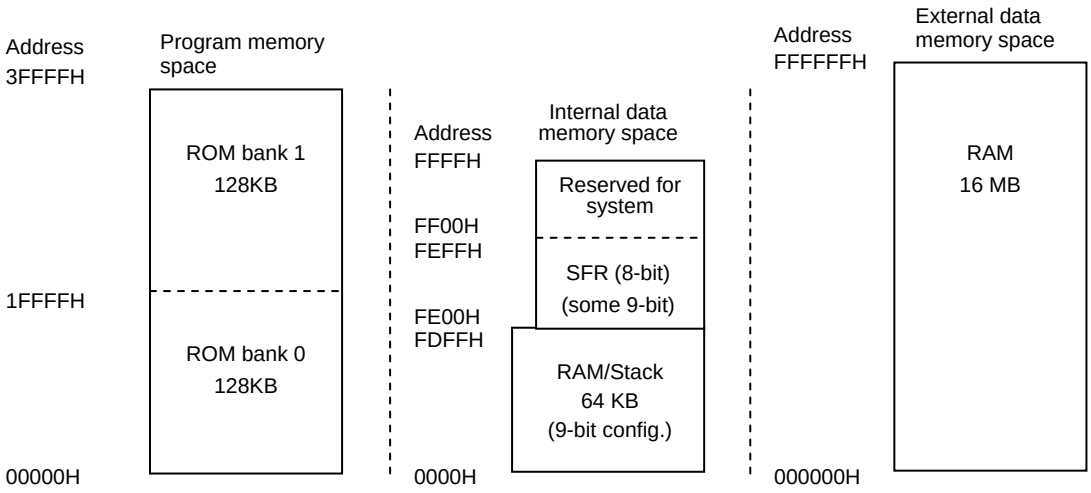
Do not apply a voltage of 3.6V or more to UHAD+, UHAD-, UHBD+, and UHBD- pins when the reference voltage generator circuit is active.

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 Appendix 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 low-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/UHC-A bus active/UHC-B bus active/ remote control receive		00013H	0
	INT3/INT5/base timer		0001BH	0
	T0H/INT6/UHC-A device connect/UHC-A device disconnect/UHC-A resume		00023H	0
	T1L/T1H/INT7/AIF start/SMIIC0/UHC-B device connect/UHC-B device disconnect/UHC-B resume		0002BH	0
	SIO0/UART1 receive end		00033H	0
	SIO1/SIO4/UART1 buffer empty/UART1 transmit end/ AIF end		0003BH	0
	ADC/T6/T7/UHC-ACK/UHC-NAK/UHC error/UHC-STALL		00043H	0
	Port 0/PWM0/ PWM1/T4/T5/UHC-SOF		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: For the flash version, the program start addresson on reset can be selected by setting options.

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 to 1FFFFH for ROM size of 64K and above, and 0FF00H to 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 high-order bits + 8 low-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.

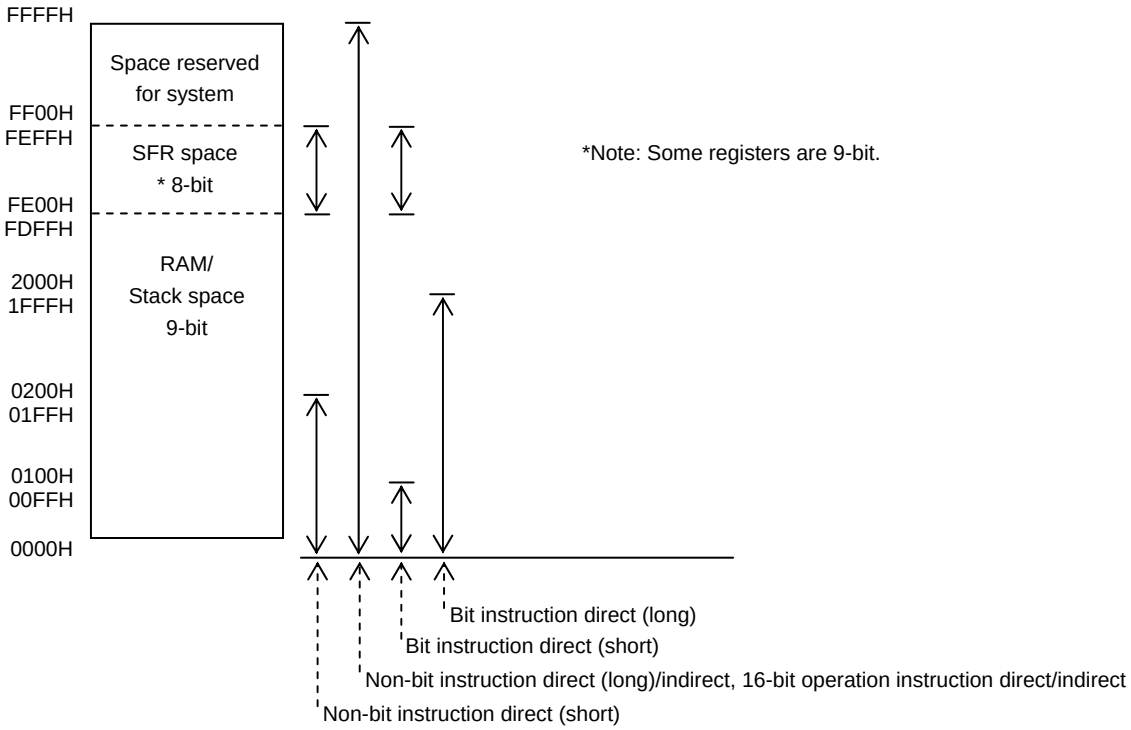


Figure 2.4.1 RAM Addressing Map

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 low-order 8 bits of the (17-bit) PC are stored in RAM address SP+1 and the high-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 high-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 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 high-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 number.
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 high-order 8 bits of a 16 bits × 8 bits multiplication is nonzero
4) When the high-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 1's in the A register. It is cleared (to 0) when there is an even number of 1's 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 (in 1-byte (9 bits) units) 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.

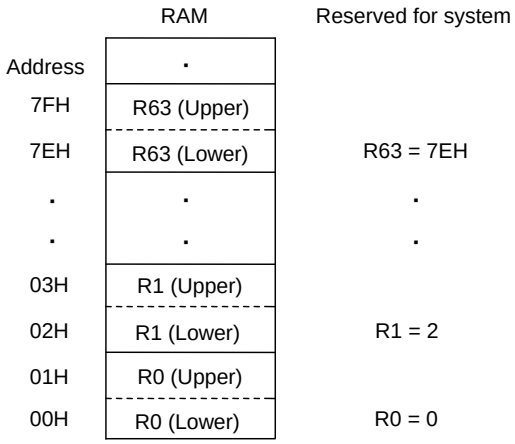


Figure 2.10.1 Allocation of Indirect Registers

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 ($0 \leq n \leq 63$)
3) Indirect register (Rn) + C register indirect ($0 \leq n \leq 63$)
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 designates 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 result 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 low-order bytes of the address.

Examples:

```
LDW     #3456H;    Sets up the low-order 16 bits.
STW     R0;        Loads the indirect register R0 with the low-order 16 bits of the address.
MOV     #12H, B;   Sets up the high-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 automatically suspend the execution of instructions in the following cases:

1) When continuous data transfer is performed on the SIO0 with SI0CTR (SCON0, bit 4) set, a wait request occurs and 1 cycle of wait operation (RAM data transfer) is performed before each 8-bit data transfer.
2) When continuous data transfer is performed on the SIO4, a wait request occurs and 1 cycle of wait operation (RAM data transfer) is carried out on each 8-bit data transfer.
3) When transmission or reception of a data packet is performed in the USB host controller circuit, a wait request occurs and 1 cycle of wait operation (RAM data transfer) is carried out on each 4-byte data transfer.
4) When transmission or reception of data is performed in the audio interface circuit, 2 (or 3) wait requests occur and 2 (or 3) cycles of wait operation (RAM data transfer) are carried out for each of the channels (Lch/Rch).

2.12.2 What is a Wait Sequence?

1) When a wait request occurs according to the event explained in Subsection 2.12.1, the CPU suspends the execution of the instruction for a predetermined cycle period, during which the required data is transferred. This is called a wait sequence.
2) Peripheral circuits such as timers and PWMs continue processing during the wait sequence.
3) The microcontroller does not perform a wait sequence when it is in HALT or HOLD mode.
4) 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	Bit 8 (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 high-order address of PSW is popped
POP_P	—	P1←RAMl (bit l)	Bit 8 is 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←low-order 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← low-order byte of CY inverted	P1←REGH8 after computation	DEC 17 bits
DBNZ	DEC 9 bits	P1←REG8	DEC 9 bits, check low-order 8 bits
DBZ	DEC 9 bits	P1←REG8	DEC 9 bits, check low-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 high-order byte of a RAM location or SFR/bit 8 of the low-order byte
RAM8: Bit 8 of a RAM location
RAMH8/RAML8: Bit 8 of the high-order byte of a RAM location/bit 8 of the low-order byte

3. Peripheral System Configuration

This chapter describes the internal functional blocks (peripheral system) of 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. The I/O direction is determined by the data direction register in 1-bit units.

This port can also be used 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 with a programmable pull-up resistor can be selected as the output type in 1-bit units.

<Note> Port P02 is temporarily set low when the microcontroller is reset.

3.1.2 Functions

1) Input/output port (8 bits: P00 to P07)
   - The port output data is controlled by the port 0 data latch (P0: FE40) and the I/O direction is controlled by the port 0 data direction register (P0DDR: FE41).
   - Each port is equipped with a programmable pull-up resistor.

2) Interrupt pin function

   P0FLG (P0FCR: FE42, bit 5) is set when the low level data is input to one of the ports whose port 0 interrupt select register (P0INTE: FE66) bit is set to 1.

   In this case, if P0IE (P0FCR: FE42, bit 4) is 1, HOLD mode is released and an interrupt request to vector address 004BH is generated.

3) Multiplexed pin functions

   P05 also serves as the system clock output, P06 as the timer 6 toggle output, P07 as the timer 7 toggle output, P00, P05 to P07 as the audio interface I/O function, and P00 to P07 as analog input channel 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	0000 0000	R/W	P0DDR	P07DDR	P06DDR	P05DDR	P04DDR	P03DDR	P02DDR	P01DDR	P00DDR
FE42	0000 0000	R/W	P0FCR	P0FCR7	P0FCR6	P0FLG	P0IE	P0FCR3	P0FCR2	P0FCR1	P0FCR0
FE4F	0000 0000	R/W	P0FCRU	T7OE	T6OE	SCKOSL5	SCKOSL4	CLKOEN	CKODV2	CKODV1	CKODV0
FE66	0000 0000	R/W	P0INTE	P07INTE	P06INTE	P05INTE	P04INTE	P03INTE	P02INTE	P01INTE	P00INTE

3.1.3 Related Registers

3.1.3.1 Port 0 data latch (P0)

1) This latch is an 8-bit register that controls the port 0 output data and pull-up resistors.
2) When this register is read with an instruction, data at pins P00 to P07 is read in. However, if P0 (FE40) is manipulated using a NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced instead of the data at the 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 an 8-bit register that controls the I/O direction of port 0 data in 1-bit units. A 1 in bit P0nDDR places port P0n into output mode, and a 0 places it into input mode.
2) When bit P0nDDR is set to 0 and bit P0n of the port 0 data latch is set to 1, port P0n becomes an input with a pull-up resistor.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE41	0000 0000	R/W	P0DDR	P07DDR	P06DDR	P05DDR	P04DDR	P03DDR	P02DDR	P01DDR	P00DDR

Register Data		Port P0n State		Internal Pull-up Resistor
P0n	P0nDDR	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

3.1.3.3 Port 0 interrupt select register (P0INTE)

1) This register is an 8-bit register that specifies the low level detection port for port 0 interrupt in 1-bit units.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE66	0000 0000	R/W	P0INTE	P07INTE	P06INTE	P05INTE	P04INTE	P03INTE	P02INTE	P01INTE	P00INTE

3.1.3.4 Port 0 interrupt control register (P0FCR)

1) This register is an 8-bit register that controls port 0 interrupt.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE42	0000 0000	R/W	P0FCR	P0FCR7	P0FCR6	P0FLG	P0IE	P0FCR3	P0FCR2	P0FCR1	P0FCR0

P0FCR (bit 7): Fixed bit

This bit must always be set to 0.

F0FCR6 (bit 6) Fixed bit

This bit must always be set to 0.

P0FLG (bit 5): P0 interrupt source flag

This flag is set when a low level is applied to either one of the ports to which a port 0 interrupt select register (P0INTE: FE66) 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

A HOLD mode release signal and an interrupt request to vector address 004BH are generated when this bit and P0FLG are set to 1.

P0FCR3 (bit 3): Fixed bit

This bit must always be set to 0.

P0FCR2: Fixed bit

This bit must always be set to 0.

P0FCR1: Fixed bit

This bit must always be set to 0.

P0FCR0: Fixed bit

This bit must always be set to 0.

3.1.3.5 Port 0 function control register (P0FCRU)

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

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 at pin P07.

This bit is disabled when P07 is in input mode (P07DDR=0).

When P07 is in output mode (P07DDR=1):

0: Carries the value of the port data latch.

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

T6OE (bit 6):

This bit controls the output data at pin P06.

This bit is disabled when P06 is in input mode (P06DDR=0).

When P06 is in output mode (P06DDR=1):

0: Carries the value of the port data latch.

1: Carries the OR of the waveform that toggles at a period 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 output to P05.

SCKOSL5	SCKOSL4	P05 Output Clock Source
0	0	Source oscillator clock selected as the system clock
0	1	Internal medium-speed RC clock
1	0	USB frequency-divided clock
1	1	Main clock (CF)

CLKOEN (bit 3):

This bit controls the output data at pin P05.

This bit is disabled when P05 is in input mode (P05DDR=0).

When P05 is in output mode (P05DDR=1):

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 output to P05.

000: Frequency of source oscillator clock to be output to P05

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

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

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

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

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

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

111: Frequency of source oscillator clock selected as subclock

<Notes on the use of the clock output function>

Follow 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 SCKOSL5 and SCKOSL4 (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 that loads the P0FCRU register with the data that sets the state of CLKOEN (bit 3) from 1 to 0. CLKOEN is set to 0 at the end of the clock that is being output (on detection of the falling edge of the clock). Accordingly, when changing the clock frequency division 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.

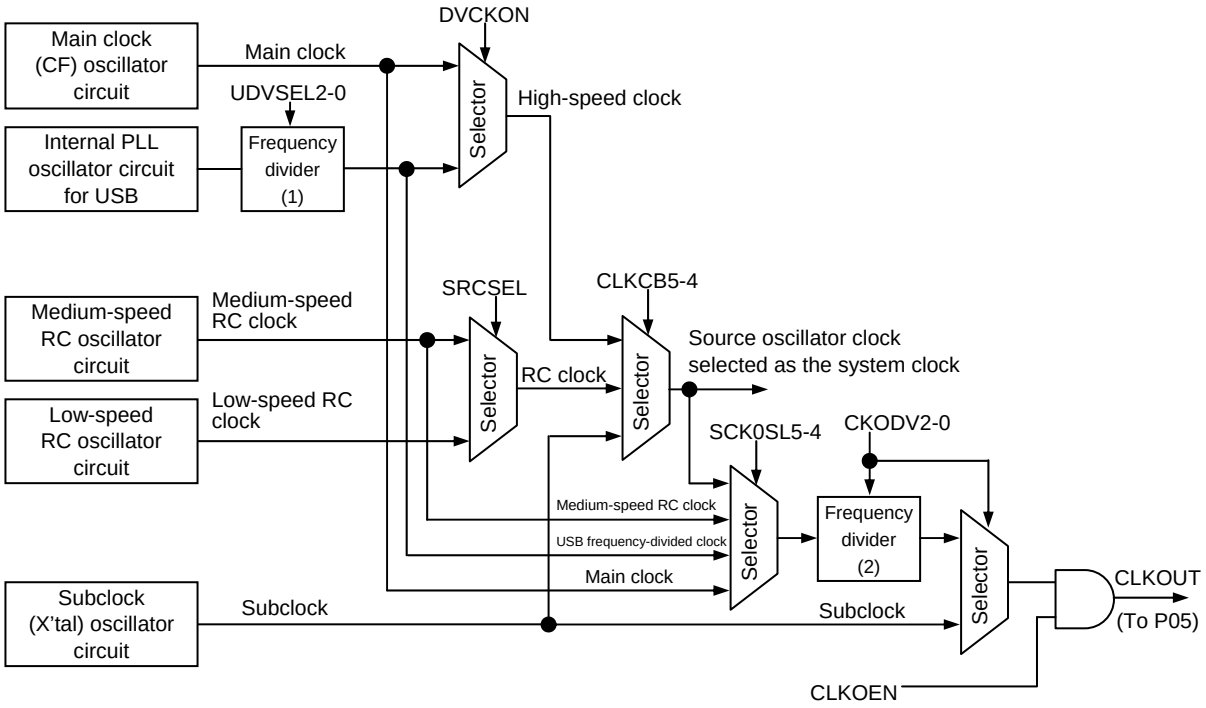


Figure 3.1.1 P05 Output Clock Selection

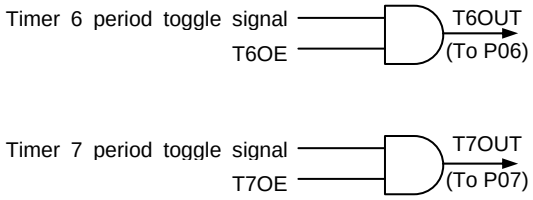


Figure 3.1.2 Timer 6, Timer 7 Toggle Output

3.1.4 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)

3.1.5 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 equipped 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. The I/O direction is determined by 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 the 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 port (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 also serves as the timer 1 PWMH/base timer buzzer 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 that controls the 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 a NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced instead of the data at the 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 multiplexed pin outputs of port 1.

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 data of timer 1 PWMH and base timer BUZ
	1	1	NAND data 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 and base timer BUZ output control)

This bit controls the output data at pin P17.

When P17 is placed in output mode (P17DDR=1) and P17FCR is set to 1, the AND data 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 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 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 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 SIO1 is operating, 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 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 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 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 SIO0 is operating, 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 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 a 6-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. The I/O direction is determined by 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, or 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 (6 bits: P20 to P25)
   - The port output data is controlled by the port 2 data latch (P2:FE48) and the I/O direction is controlled by the port 2 data direction register (P2DDR:FE49).
   - Each port is equipped with a programmable pull-up resistor.

2) Interrupt input pin function
   - A port (INT4) selected from P20 to P23 and a port (INT5) selected from P24, P25,UHBD–, and UHBD+ are provided with a pin interrupt function that detects a low edge, high edge, or both edges and sets the interrupt flag. These selected two ports can also be used as the timer 1 count clock input and timer 0 capture signal input.
   - P20 (INT6) and P24 (INT7) are provided with a pin interrupt function that detects a low edge, high edge, or both edges and sets the interrupt flag. They can also be used as the timer 0 capture 1 signal input.

3) Hold mode release function
   - When both the interrupt flag and the interrupt enable flag are set for INT4 or INT5, a HOLD mode release signal is generated, releasing HOLD mode. The CPU then enters HALT mode (system clock by medium-speed RC or low-speed RC). 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 the INT4 or INT5 data that is established when HOLD mode is entered is in the high state, or by a falling edge occurring when the INT4 or INT5 data that 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 both-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	I45CR	INT5HEG	INT5LEG	INT5IF	INT5IE	INT4HEG	INT4LEG	INT4IF	INT4IE
FE4B	0000 0000	R/W	145SL	I5SL3	I5SL2	I5SL1	I5SL0	I4SL3	I4SL2	I4SL1	I4SL0
FE4E	0000 0000	R/W	I67CR	INT7HEG	INT7LEG	INT7IF	INT7IE	INT6HEG	INT6LEG	INT6IF	INT6IE

3.3.3 Related Registers

3.3.3.1 Port 2 data latch (P2)

1) This latch is an 8-bit register that controls the port 2 output data and pull-up resistors and the output data from the USB downstream B port (UHBD– and UHBD+).
2) When this register is read with an instruction, data at pins P20 to P25, UHBD–, and UHBD+ is read in. However, if P2 (FE48) is manipulated using a NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced instead of the data at the pins.
3) Port 2 data and USB downstream B port 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 the port 2 data and the general-purpose port output mode of the USB downstream B port in 1-bit units. Port P2n are placed in output mode when bit P2nDDR is set to 1 and in input mode when bit P2nDDR is set to 0 (n＝0 to 5). UHBD– is placed in general-purpose port output mode when bit P26DDR is set to 1 and UHBD+ is placed in general-purpose port output mode when P27DDR is set to 1.
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 (n＝0 to 5).

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 (n=0 to 5)		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

Register Data		UHBD–/UHBD+ Port State (n=6 and 7)		Internal Pull-up Resistor
P2n	P2nDDR	Input	Output	
0	0	Enabled	UBS communication mode	None
1	0	Enabled		
0	1	Enabled	CMOS low output	
1	1	Enabled	CMOS high output	

3.3.3.3 External interrupt 4/5 control register (I45CR)

1) This register is an 8-bit register that controls 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 the INT5 data that is established when HOLD mode is entered is in the high state, or by a falling edge occurring when the INT5 data that 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 both-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 the INT4 data that is established when HOLD mode is entered is in the high state, or by a falling edge occurring when the INT4 data that 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 both-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 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	I5SL3	I5SL2	I5SL1	I5SL0	I4SL3	I4SL2	I4SL1	I4SL0

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	UHBD–
1	1	UHBD+

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 together 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.*

3.3.3.5 External interrupt 6/7 control register (I67CR)

1) This register is an 8-bit register that controls external interrupts 6 and 7.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE4E	0000 0000	R/W	I67CR	INT7HEG	INT7LEG	INT7IF	INT7IE	INT6HEG	INT6LEG	INT6IF	INT6IE

INT7HEG (bit 7): INT7 rising edge detection control

INT7LEG (bit 6): INT7 falling edge detection control

Timer 0H capture 1 signal is generated when the data change specified by bits 7 and 6 is given to pin P24.

INT7HEG	INT7LEG	INT7 Interrupt Conditions (P24 Pin Data)
0	0	No edge detected
0	1	Falling edge detected
1	0	Rising edge detected
1	1	Both edges detected

INT7IF (bit 5): INT7 interrupt source flag

This bit is set when the conditions specified by INT7HEG and INT7LEG are satisfied.

When this bit and the INT7 interrupt request enable bit (INT7IE) are set to 1, an interrupt request to vector address 002BH is generated.

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

INT7IE (bit 4): INT7 interrupt request enable

When this bit and INT7IF are set to 1, an interrupt request to vector address 002BH is generated.

INT6HEG (bit 3): INT6 rising edge detection control

INT6LEG (bit 2): INT6 falling edge detection control

Timer 0L capture 1signal is generated when the data change specified by bits 3 and 2 is given to pin P20.

INT6HEG	INT6LEG	INT6 Interrupt Conditions (P20 Pin Data)
0	0	No edge detected
0	1	Falling edge detected
1	0	Rising edge detected
1	1	Both edges detected

INT6IF (bit 1): INT6 interrupt source flag

This bit is set when the conditions specified by INT6HEG and INT6LEG are satisfied.

When this bit and the INT6 interrupt request enable bit (INT6IE) are set to 1, an interrupt request to vector address 0023H is generated.

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

INT6IE (bit 0): INT6 interrupt request enable

When this bit and INT6IF are set to 1, an interrupt request to vector address 0023H is generated.

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 equipped with programmable pull-up resistors. It is made up of a data latch, a data direction register, and a control circuit. The I/O direction is determined by 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.

3.4.2 Functions

1) Input/output port (5 bits: P30 to P34)
   - The port output data is controlled by the port 3 data latch (P3: FE4C), and the I/O direction is controlled by the port 3 data direction register (P3DDR: FE4D).
   - Each port is equipped with a programmable pull-up resistor.

2) Multiplexed functions

   P34 also serves as the PLL filter connection pin for USB interface and P30 to P31 as the UART1 I/O.

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 that controls the port 3 output data and pull-up resistors.
2) When this register is read with an instruction, the data at pins P30 to P34 is read in. However, if P3 (FE4C) is manipulated using a 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 that controls 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) When bit P3nDDR is set to 0 and bit P3n of the port 3 data latch is set to 1, port P3n becomes an input with a pull-up resistor.

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 equipped with programmable pull-up resistors. It is made up of a data control latch and a control circuit. The I/O direction is determined in 1-bit units.

Port 7 can also serve 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 low-order 4 bits of the port 7 control register (P7: FE5C) are used to control the port output data, and the high-order 4 bits are used to control the I/O direction of port data.
   - P70 is the N-channel open drain output type and P71 to P73 are the 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 are used to detect a low or high level, a low or high edge of the signal and to set the interrupt flag.
   - P72 and P73 are assigned to INT2 and INT3, respectively, and are used to detect a low or high edge, or both edges and to 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 a 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 a 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 intervals for the duration of the input 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 a 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 intervals for the duration of the input signal.

6) HOLD mode release function
   - When the interrupt flag and interrupt enable flag are set for INT0, INT1, or INT2, a HOLD mode release signal is generated, releasing HOLD mode. The CPU then enters HALT mode (system clock by medium-speed RC or low-speed RC). When the interrupt is accepted, the CPU switches from HALT mode to normal operating mode.

- When a signal level 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 the P72 data that is established when HOLD mode is entered is in the high state, or by a falling edge occurring when the P72 data that 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 both-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.

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 that controls the I/O of port 7 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 a 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 the 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.

PORT 7

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 the internal pull-up resistor on or off 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 the internal pull-up resistor on or off 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 the internal pull-up resistor on or off 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 pin is the N-channel open drain output type, however, it is placed in a high-impedance output state when P70 is set to 1.

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

3.5.3.2 External interrupt 0/1 control register (I01CR)

1) This register is an 8-bit register that controls 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 (only when level detection is set) 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 (only when level detection is set) 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 (only when level detection is set) 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 (only when level detection is set) 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 that controls 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 the P72 data that is established when HOLD mode is entered is in the high state, or by a falling edge occurring when the P72 data that 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 both-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 that controls the timer 0 input, noise filter time constant, buzzer output/timer 1PWMH output select, 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 this bit is 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 this bit is 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	Low-speed RC oscillator clock
1	1	Timer/counter 0 prescaler output

BUZON (bit 3): Buzzer output/timer 1PWMH output select

When P17FCR (P1FCR, bit 1) is set to 1, this bit selects the data (buzzer output/timer 1 PWMH output) to be sent to port P17.

When this bit is set to 1, timer 1 PWMH output is fixed at a high level and a signal that is obtained by dividing the base timer clock by 16 (fBST/16) is sent to port P17 as buzzer output.

When this bit is set to 0, buzzer output is fixed at a high level and the timer 1 PWMH output data is sent to port P17.

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 this bit is 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 of 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 two 8-bit capture registers) × 2 channels
2) Mode 1: 8-bit programmabe timer with a programmable prescaler (with two 8-bit capture registers) + 8-bit programmable counter (with two 8-bit capture registers)
3) Mode 2: 16-bit programmable timer with a programmable prescaler (with two 16-bit capture registers)
4) Mode 3: 16-bit programmable counter (with two 16-bit capture registers)

3.6.2 Functions

1) Mode 0: 8-bit programmable timer with a programmable prescaler (with two 8-bit capture registers) × 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, P20 to P25, UHBD–, and UHBD+ timer 0L capture input pins.
   - The contents of T0L are captured into the capture register T0CA1L on external input detection signals from the P20/INT4/T1IN/INT6/T0LCP1 pin.
   - The contents of T0H are captured into the capture register T0CAH on external input detection signals from the P71/INT1/T0HCP, P73/INT3/T0IN, P20 to P25, UHBD–, and UHBD+ timer 0H capture input pins.
   - The contents of T0H are captured into the capture register T0CA1H on external input detection signals from the P24/INT5/T1IN/INT7/T0HCP1 pin.

     T0L period = (T0LR + 1) × (T0PRR + 1) × Tcyc

     T0H period = (T0HR + 1) × (T0PRR + 1) × Tcyc

     Tcyc = Period of cycle clock

2) Mode 1: 8-bit programmable timer with a programmable prescaler (with two 8-bit capture registers) + 8-bit programmable counter (with two 8-bit capture registers)
   - 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 P70/INT0/T0LCP, P72/INT2/T0IN, P20 to P25, UHBD–, and UHBD+ timer 0L capture input pins.
   - The contents of T0L are captured into the capture register T0CA1L on external input detection signals from P20/INT4/T1IN/INT6/T0LCP1 pin.
   - The contents of T0H are captured into the capture register T0CAH on external input detection signals from P71/INT1/T0HCP, P73/INT3/T0IN, P20 to P25, UHBD–, and UHBD+ timer 0H capture input pins.
   - The contents of T0H are captured into the capture register T0CA1H on external input detection signals from P24/INT5/T1IN/INT7/T0HCP1 pin.

     T0L period = (T0LR + 1)

     T0H period = (T0HR + 1) × (T0PRR +1) × Tcyc

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

- 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, P20 to P25, UHBD–, and UHBD+ timer 0H capture input pins.
- The contents of T0L and T0H are captured into the capture registers T0CA1L and T0CA1H at the same time on external input detection signals from the P24/INT5/T1IN/INT7/T0HCP1 pin.

$$\text{T0 period} = ([\text{T0HR, T0LR}] + 1) \times (\text{T0PRR} + 1) \times \text{Tcyc}$$

16 bits

4) Mode 3: 16-bit programmable counter (with two 16-bit capture registers)

- 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, P20 to P25, UHBD–, and UHBD+ timer 0H capture input pins.
- The contents of T0L and T0H are captured into the capture registers T0CA1L and T0CA1H at the same time on external input detection signals from the P24/INT5/T1IN/INT7/T0HCP1 pin.

$$\text{T0 period} = [\text{T0HR, T0LR}] + 1$$

16 bits

5) Interrupt generation

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

6) It is necessary to manipulate the following special function registers to control timer/counter 0 (T0).

- T0CNT, T0PRR, T0L, T0H, T0LR, T0HR
- P7, ISL, I01CR, I23CR
- P2, P2DDR, I45CR, I45SL, I67CR

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)

This register controls the operation and interrupts of T0L and T0H.

3.6.3.2 Programmable prescaler match register (T0PRR) (8-bit register)

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 the 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: Stop/start is controlled by the 0/1 value of T0LRUN (timer/counter 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/counter 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: Stop/start is controlled by the 0/1 value of T0HRUN (timer/counter 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/counter 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 must 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 low-order byte of timer/counter 0 (16 bits of data must match in the 16-bit mode).
2) The match buffer register is updated as follows:
   - When it is inactive (T0LRUN=0), the match buffer 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 high-order byte of timer/counter 0 (16 bits of data must match in the 16-bit mode).
2) The match buffer register is updated as follows:
   - When it is inactive (T0HRUN=0), the match buffer 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, P20 to P25, UHBD–, and UHBD+ timer 0L capture input pins when T0LONG (timer/counter 0 control register, bit 5) is set to 0.
   External input detection signals from the P71/INT1/T0HCP, P73/INT3/T0IN, P20 to P25, UHBD–, and UHBD+ timer 0L capture input pins when T0LONG (timer/counter 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, P20 to P25, UHBD–, and UHBD+ timer 0H capture input pins.
2) Capture data: Contents of timer/counter 0 high byte (T0H)

3.6.3.10 Timer/counter 0 capture register 1 low byte (T0CA1L) (8-bit register)

1) Capture clock: External input detection signals from the P20/INT4/T1IN/INT6/T0LCP1 pin when T0LONG (timer/counter 0 control register, bit 5) is set to 0.
   External input detection signals from the P24/INT5/T1IN/INT7/T0HCP1 pin when T0LONG (timer/counter 0 control register, bit 5) is set to 1.
2) Capture data: Contents of timer/counter 0 low byte (T0L).

3.6.3.11 Timer/counter 0 capture register 1 high byte (T0CA1H) (8-bit register)

1) Capture clock: External input detection signals from the P24/INT5/T1IN/INT7/T0HCP1 pin.
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

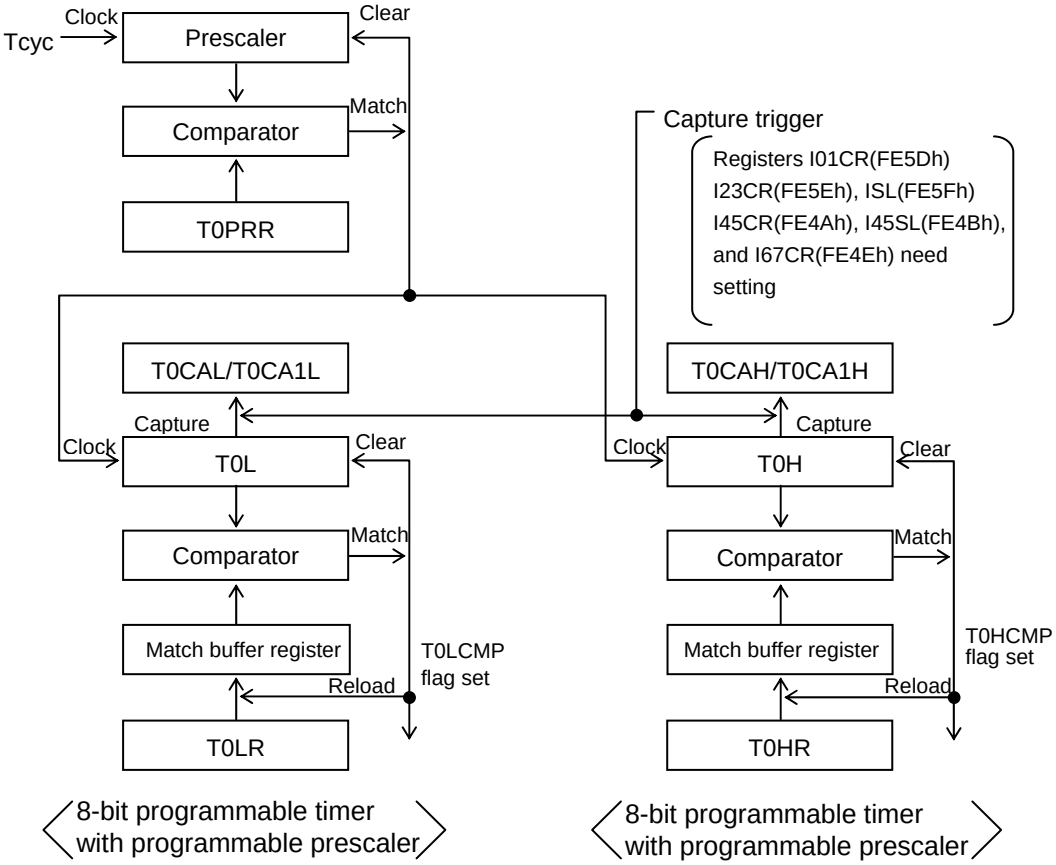


Figure 3.6.1 Mode 0 Block Diagram (T0LONG =0 , T0LEXT = 0)

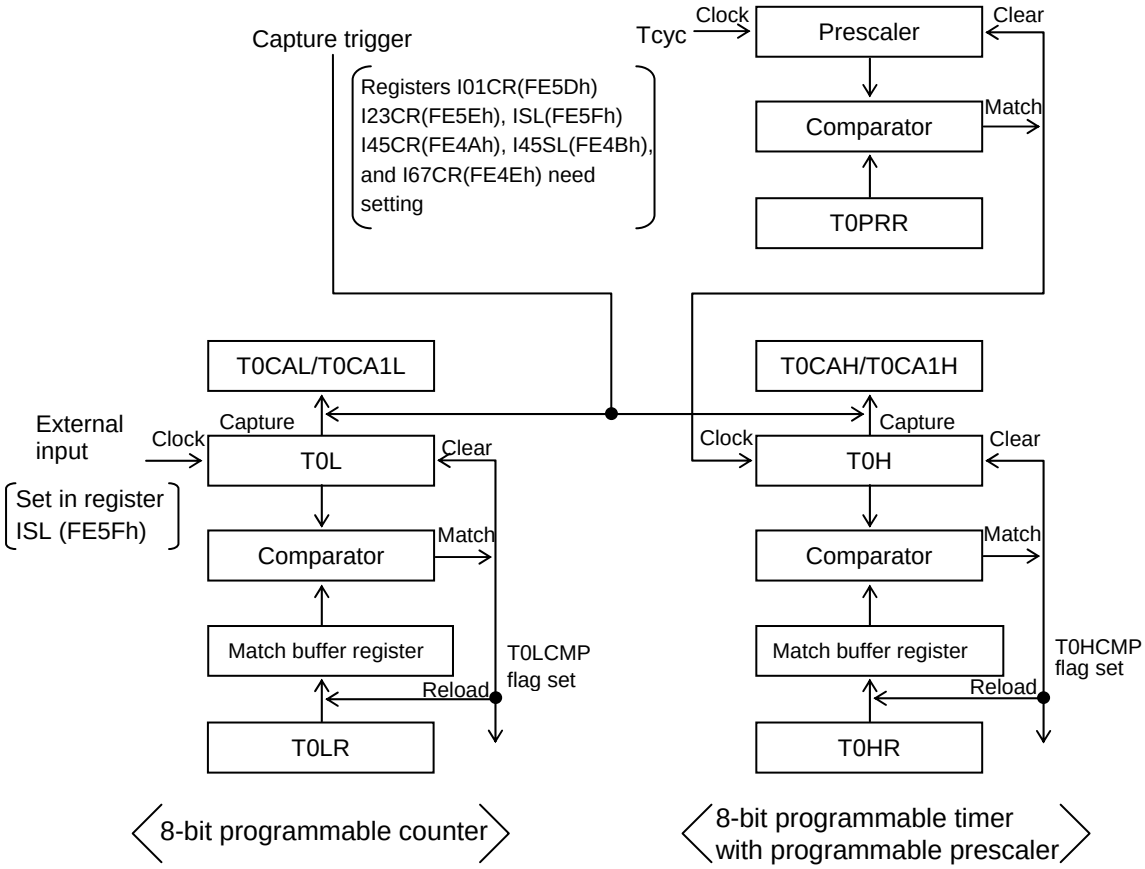


Figure 3.6.2 Mode 1 Block Diagram (T0LONG = 0, T0LEXT = 1)

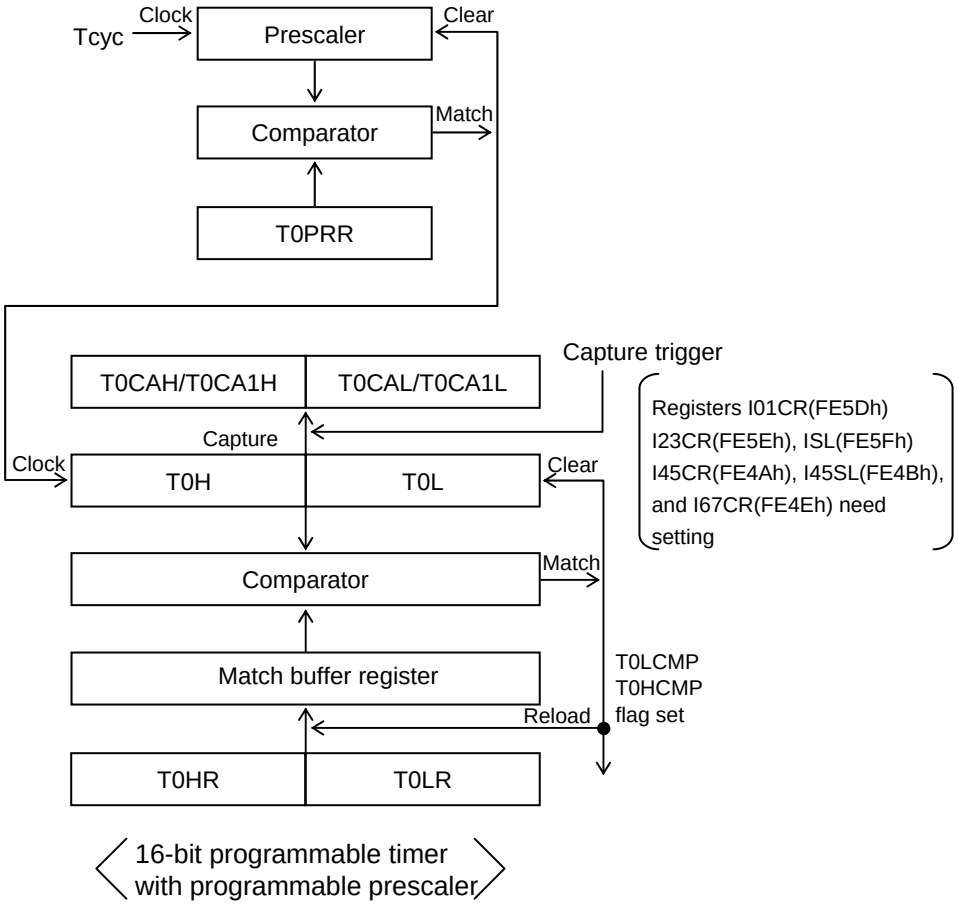


Figure 3.6.3 Mode 2 Block Diagram (T0LONG = 1, T0LEXT = 0)

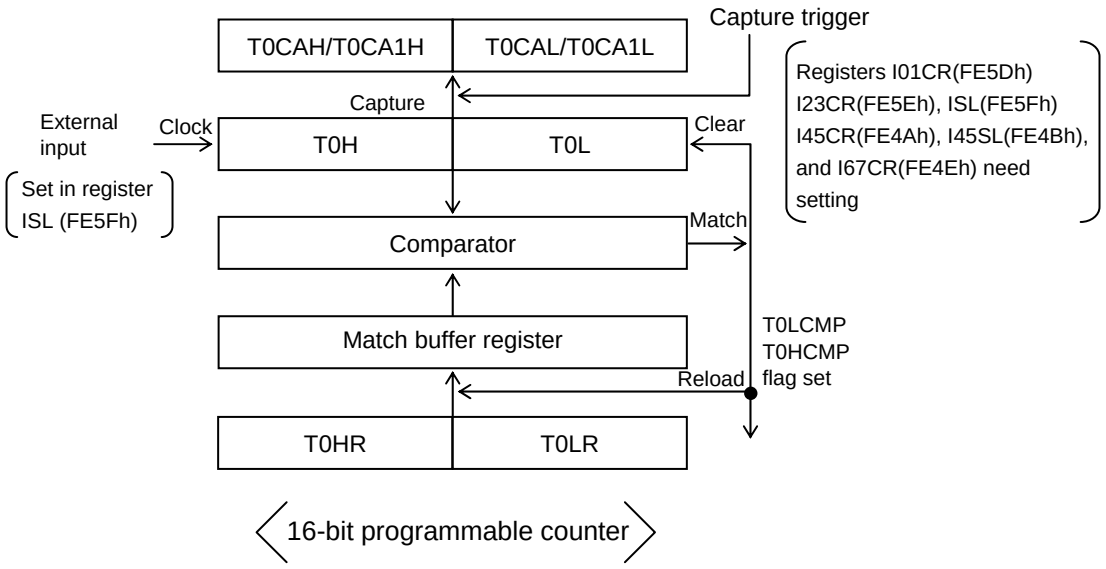


Figure 3.6.4 Mode 3 Block Diagram (T0LONG = 1, T0LEXT = 1)

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 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 high-order and low-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 consisting of T0H and T0L matches the contents of the match buffer registers 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 16-bit mode (T0LONG=1), a match must 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 16-bit mode (T0LONG=1), a match must 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) This register is an 8-bit register that is used to determine the clock period (Tpr) of timer/counter 0.
2) The count value of the prescaler starts at 0 when data is loaded into T0PRR.
3) Tpr = (T0PRR+1) × 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 low-order byte of timer/counter 0 (16 bits of data must match in the 16-bit mode).
2) The match buffer register is updated as follows:
   - When it is inactive (T0LRUN=0), the match buffer 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 high-order byte of timer/counter 0 (16 bits of data must match in the 16-bit mode).
2) The match buffer register is updated as follows:
   - When it is inactive (T0HRUN=0), the match buffer 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

3.6.4.9 Timer/counter 0 capture register 1 low byte (T0CA1L)

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
FE1E	XXXX XXXX	R	T0CA1L	T0CA1L7	T0CA1L6	T0CA1L5	T0CA1L4	T0CA1L3	T0CA1L2	T0CA1L1	T0CA1L0

3.6.4.10 Timer/counter 0 capture register 1 high byte (T0CA1H)

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
FE1F	XXXX XXXX	R	T0CA1H	T0CA1H7	T0CA1H6	T0CA1H5	T0CA1H4	T0CA1H3	T0CA1H2	T0CA1H1	T0CA1H0

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 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
   - Coupling the timer/counter 0 low byte (T0L) and timer/counter 0 high byte (T0H), the clock counter 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) It is necessary to manipulate the following special function registers to control the high-speed clock counter.
   - 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) This register controls the high-speed clock counter. It contains the start bit, count value setting bit, and counter value capture bit.
2) Start/stop: Controlled by the start/stop operation of timer/counter 0 low byte (T0L) when NKEN=1.
3) Count clock: External input signals from the P72/INT2/T0IN/NKIN pin.
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 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 0.
2) The real-time output function is disabled when DSNKOT (P1TST register, bit 2) is set to 1.

   In this case, the real-time output pin functions as an ordinary port pin.

3.7.3.3 Timer/counter 0 operation

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

When NKEN=1 and T0LONG (T0CNT register, 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 register

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 high-order counter. Counting is started by setting this bit to 1 and starting timer 0 in the external clock mode.

NKCMP2 to NKCMP0 (bits 6 to 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," 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 to NKCAP0 (bits 3 to 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.

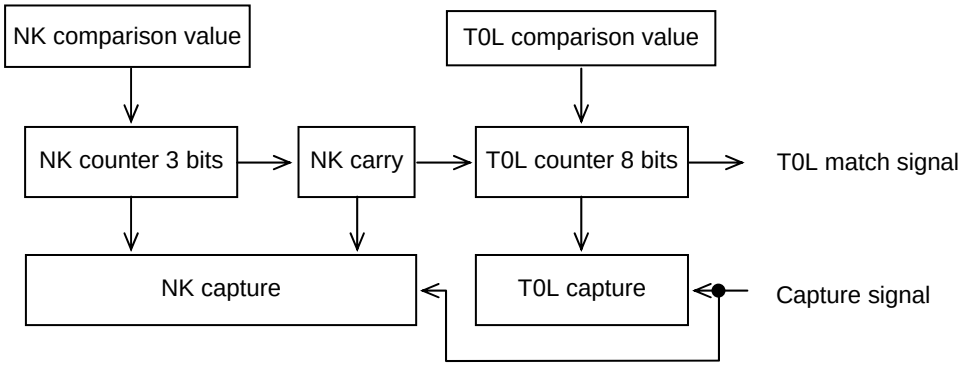


Figure 3.7.1 T0LONG = 0 Block Diagram (Timer 0: 8-bit mode)

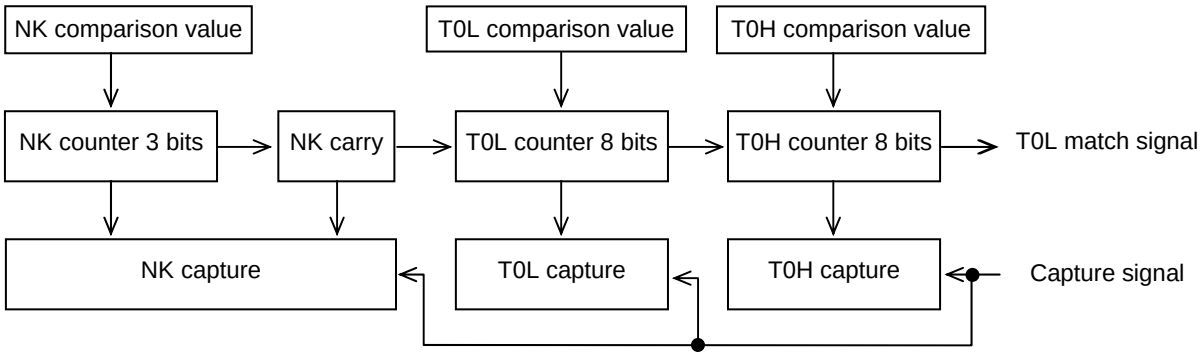


Figure 3.7.2 T0LONG = 1 Block Diagram (Timer 0: 16-bit mode)

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 low-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 low-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 period = (T1LR + 1) × (T1LPRC count) × 2Tcyc or
     (T1LR + 1) × (T1LPRC count) events detected

     T1PWML period = T1L period × 2

     T1H period = (T1HR + 1) × (T1HPRC count) × 2Tcyc

     T1PWMH period = T1H period × 2

     Tcyc = 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 period = 256 × (T1LPRC count) × Tcyc

     T1PWML low period = (T1LR + 1) × (T1LPRC count) × Tcyc

     T1PWMH period = 256 × (T1HPRC count) × Tcyc

     T1PWMH low period = (T1HR + 1) × (T1HPRC count) × Tcyc

3) Mode 2: 16-bit programmable timer/counter with an 8-bit prescaler (with toggle output)
   (The low-order 8 bits may be used as a timer/counter with toggle output.)
   - T1 functions a16-bit programmable timer/counter that counts the number of signals obtained by dividing the cycle clock by 2 or external events. Since interrupts can occur from the low-order 8-bit timer (T1L) at the interval of T1L period, the low-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)

     T1L period = (T1LR + 1) × (T1LPRC count) × 2Tcyc or
     (T1LR + 1) × (T1LPRC count) events detected

     T1PWML period = T1L period × 2

     T1 period = (T1HR + 1) × (T1HPRC count) × T1L period

     T1PWMH period = T1 period × 2

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

- A 16-bit programmable timer runs on the cycle clock.
- The low-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)

T1PWML period = 256 × (T1LPRC count) × Tcyc
T1PWML low period = (T1LR + 1) × (T1LPRC count) × Tcyc
T1 period = (T1HR + 1) × (T1HPRC count) × T1PWML period
T1PWMH period = T1 period × 2

5) Interrupt generation

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

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

- 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 T1PWML is fixed at a high level if T1L is stopped. If T1L is running, the output of T1PWML is fixed at a low level when T1LR = FFH. The output of T1PWMH is fixed at a high level if T1H is stopped. If T1H is running, the output of T1PWMH is fixed at a 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 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: Stop/start is controlled by the 0/1 value of T1LRUN (timer 1 control register, bit 6).
2) Count clock: Depends on the operating 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: Stop/start is controlled by the 0/1 value of T1HRUN (timer 1 control register, bit 7).
2) Count clock: Depends on the operating 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 × (T1LPRC count) × 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 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: Stop/start 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 in mode 0, or 2.

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

1) Start/stop: Stop/start 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 in mode 0, 2, or 3.

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 matches the value of timer 1 low byte (T1L).
2) The match buffer register is updated as follows:
   - When it is inactive (T1LRUN=0), the match buffer 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 matches the value of timer 1 high byte (T1H).
2) The match buffer register is updated as follows:
   - When it is inactive (T1HRUN=0), the match buffer 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 a high level when T1L is inactive. When T1L is active, the T1PWML output is fixed at a low level when T1LR = FFH.
2) When T1PWM (timer 1 control register, bit 4) is set to 0, timer 1 low byte output is a toggle output whose state changes on a T1L match signal.
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 a high level when T1H is inactive. When T1H is active, the T1PWMH output is fixed at a low level when T1HR = FFH.
2) When T1PWM = 0 or T1LONG = 1, the timer 1 high byte output is a toggle output whose state changes on a T1H match signal.
3) When T1PWM = 1 and T1LONG = 0, this PWM output is cleared on a T1H overflow and set on a T1H match signal.

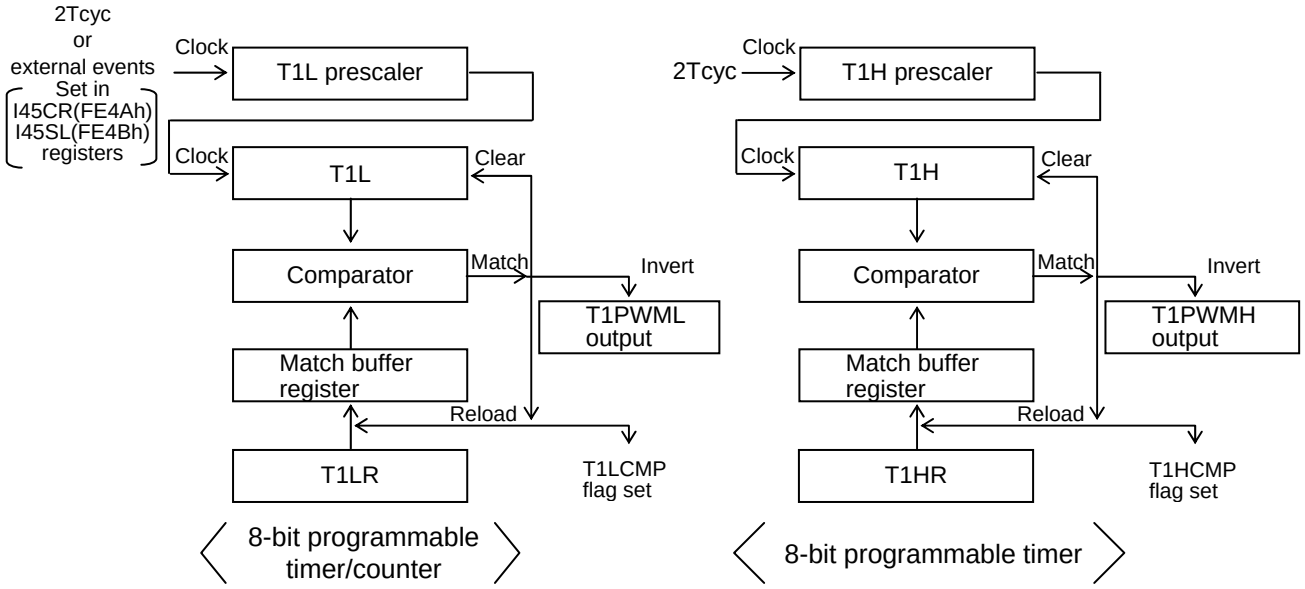


Figure 3.8.1 Mode 0 Block Diagram (T1LONG = 0, T1PWM = 0)

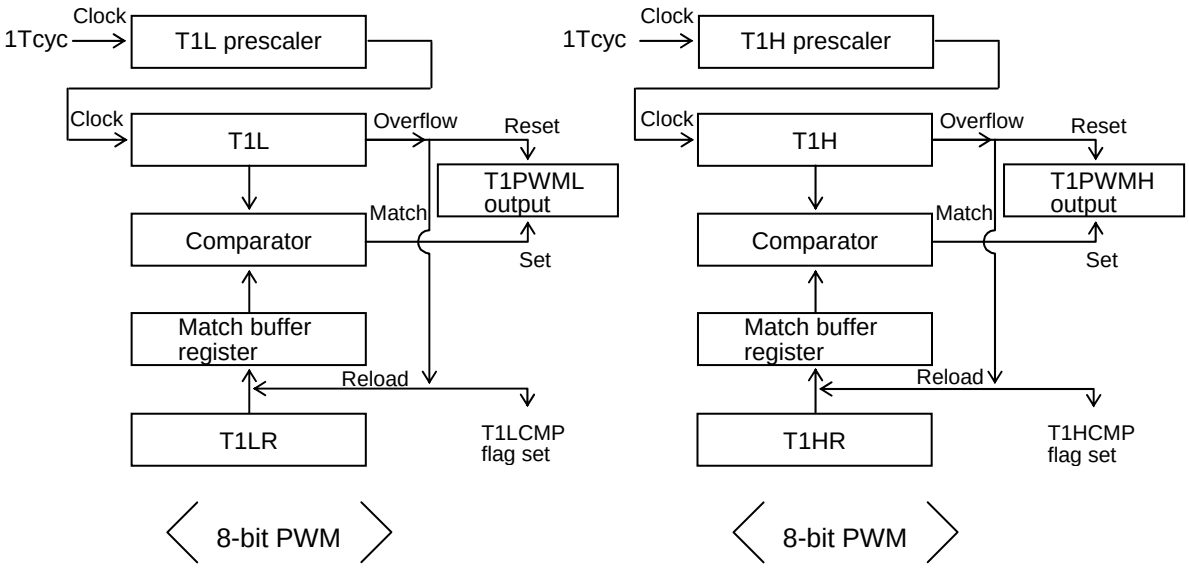


Figure 3.8.2 Mode 1 Block Diagram (T1LONG = 0, T1PWM = 1)

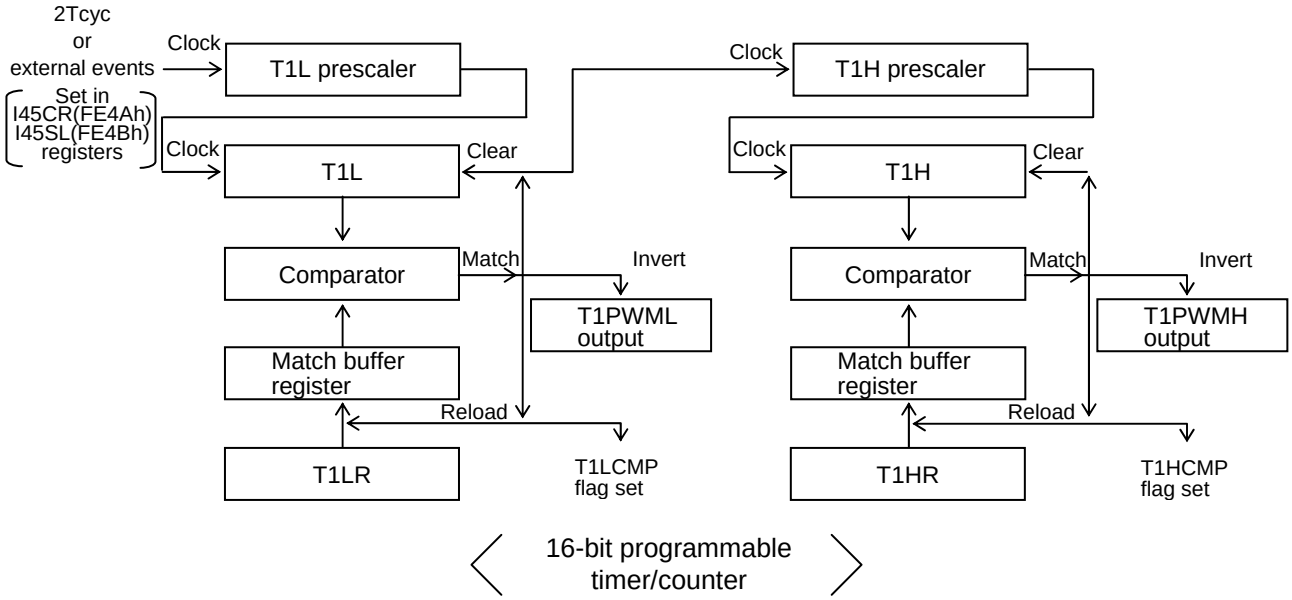


Figure 3.8.3 Mode 2 Block Diagram (T1LONG = 1, T1PWM = 0)

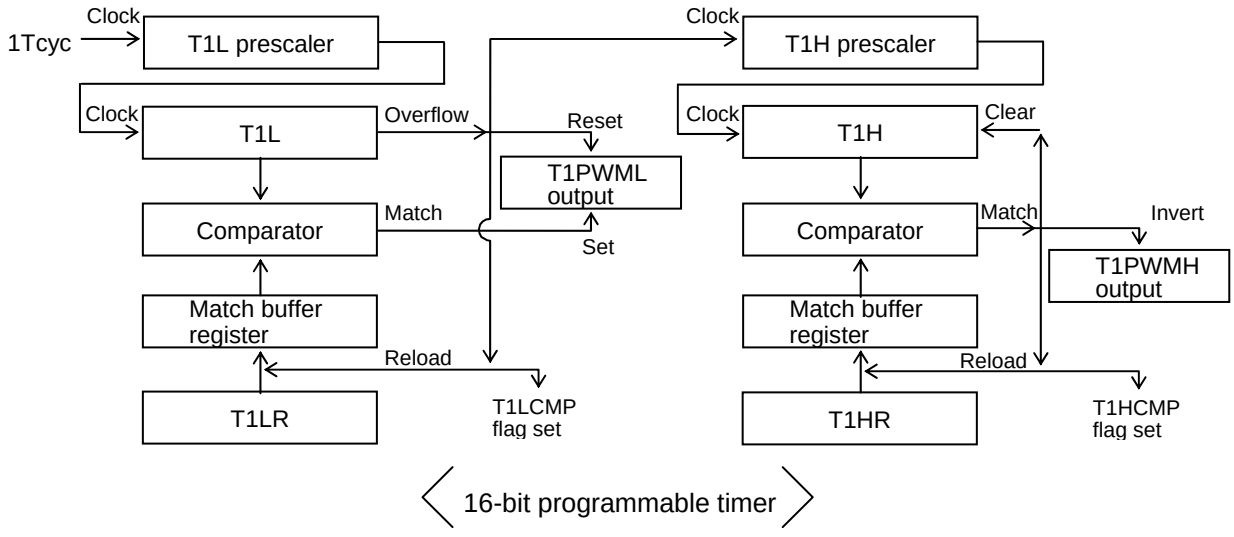


Figure 3.8.4 Mode 3 Block Diagram (T1LONG = 1, T1PWM = 1)

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 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 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 high and low 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) × (T1HPRC count) × 2Tcyc} × 2	Toggle output	Period: {(T1LR+1) × (T1LPRC count) × 2Tcyc} × 2
					or	Period: {(T1LR+1) × (T1LPRC count) × events} × 2
1	0	1	PWM output	Period: 256 × (T1HPRC count) × Tcyc	PWM output	Period: 256 × (T1LPRC count) × Tcyc
2	1	0	Toggle output	Period: {(T1HR+1) × (T1HPRC count) × (T1LR+1) × (T1LPRC count) × 2 Tcyc} ×2	Toggle output	Period: {(T1LR+1) × (T1LPRC count) × 2Tcyc} × 2
			or	Period: {(T1HR+1) × (T1HPRC count) × (T1LR+1) × (T1LPRC count) × events} × 2	or	Period: {(T1LR+1) × (T1LPRC count) × events} × 2
3	1	1	Toggle output	Period: {(T1HR+1) × (T1HPRC count) × 256 × (T1LPRC count) × Tcyc} × 2	PWM output	Period: 256 × (T1LPRC count) × 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): Timer 1 prescaler high byte control

T1HPRC2 (bit 6): Timer 1 prescaler high byte control

T1HPRC1 (bit 5): Timer 1 prescaler high byte control

T1HPRC0 (bit 4): Timer 1 prescaler high byte control

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): Timer 1 prescaler low byte control

T1LPRC2 (bit 2): Timer 1 prescaler low byte control

T1LPRC1 (bit 1): Timer 1 prescaler low byte control

T1LPRC0 (bit 0): Timer 1 prescaler low byte control

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 the 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 the 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 matches the value of timer 1 low byte.
2) The match buffer register is updated as follows:
   - When it is inactive (T1LRUN=0), the match buffer 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

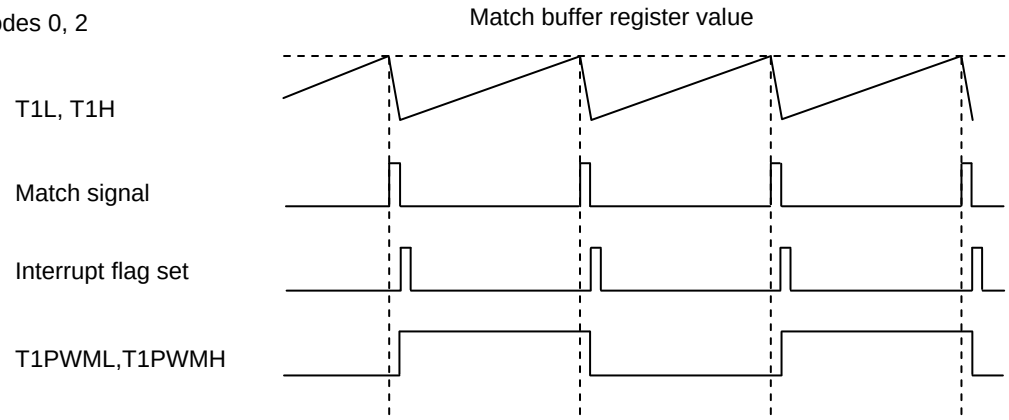
3.8.4.6 Timer 1 match data register high byte (T1HR)

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 matches the value of timer 1 high byte.
2) The match buffer register is updated as follows:
   - When it is inactive (T1HRUN=0), the match buffer 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.

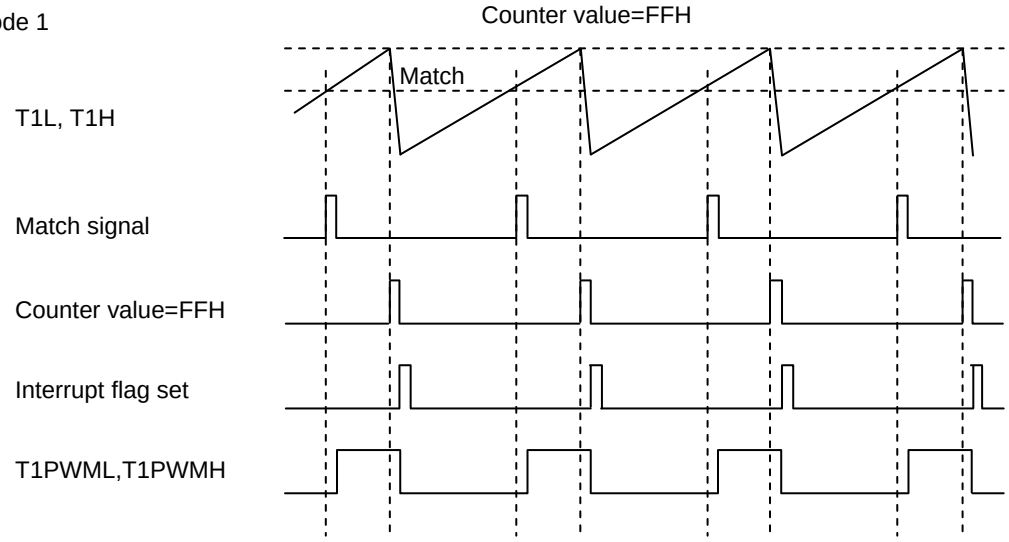
Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE1D	0000 0000	R/W	T1HR	T1HR7	T1HR6	T1HR5	T1HR4	T1HR3	T1HR2	T1HR1	T1HR0

T1

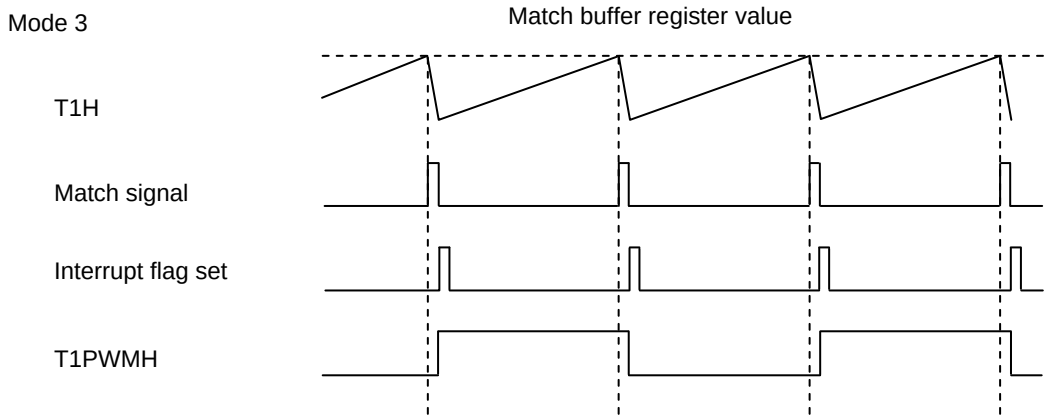
Modes 0, 2

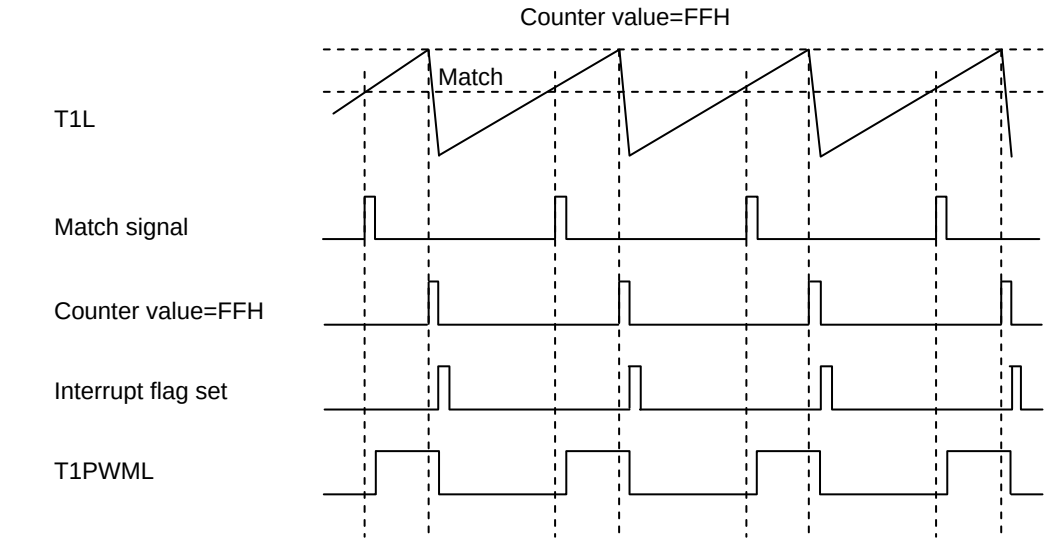


Mode 1



Mode 3


Match buffer register value
T1H
Match signal
Interrupt flag set
T1PWMH


Counter value=FFH
Match
T1L
Match signal
Counter value=FFH
Interrupt flag set
T1PWML

3.9 Timers 4 and 5 (T4, T5)

3.9.1 Overview

Timer 4 (T4) and timer 5 (T5) incorporated in this series of microcontrollers are 8-bit timers with two independently controlled 6-bit prescalers.

3.9.2 Functions

1) Timer 4 (T4)

Timer 4 is an 8-bit timer that runs on either 4Tcyc, 16Tcyc, or 64Tcyc clock.

$$\text{T4 period} = (\text{T4R}+1) \times 4^{n}\text{Tcyc} \quad (n=1, 2, 3)$$

Tcyc = Period of cycle clock

2) Timer 5 (T5)

Timer 5 is an 8-bit timer that runs on either 4Tcyc, 16Tcyc, 64Tcyc clock.

$$\text{T5 period} = (\text{T5R}+1) \times 4^{n}\text{Tcyc} \quad (n=1, 2, 3)$$

Tcyc = Period of cycle clock

3) Interrupt generation

Interrupt request to vector address 004BH is generated when the overflow flag is set at the interval of timer 4 or timer 5 period and the corresponding interrupt request enable bit is set.

4) It is necessary to manipulate the following special function registers to control timer 4 (T4) and timer 5 (T5).

- T45CNT, T4R, T5R

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE3C	0000 0000	R/W	T45CNT	T5C1	T5C0	T4C1	T4C0	T5OV	T5IE	T4OV	T4IE
FE3E	0000 0000	R/W	T4R	T4R7	T4R6	T4R5	T4R4	T4R3	T4R2	T4R1	T4R0
FE3F	0000 0000	R/W	T5R	T5R7	T5R6	T5R5	T5R4	T5R3	T5R2	T5R1	T5R0

3.9.3 Circuit Configuration

3.9.3.1 Timer 4/5 control register (T45CNT) (8-bit register)

1) This register controls the operation and interrupts of T4 and T5.

3.9.3.2 Timer 4 counter (T4CTR) (8-bit counter)

1) The timer 4 counter counts the number of clocks from the timer 4 prescaler (T4PR). The value of the timer 4 counter (T4CTR) is reset to 0 on the next clock that reaches value specified in the timer 4 period setting register (T4R), when the interrupt flag (T4OV) is set.
2) When T4C0 and T4C1 (T45CNT: FE3C, bits 4 and 5) are set to 0, the timer 4 counter stops at a count value of 0. In other cases, the timer 4 counter continues operation.
3) When data is written into T4R while timer 4 is running, both the timer 4 prescaler and counter are cleared and start counting again.

3.9.3.3 Timer 4 prescaler (T4PR) (6-bit counter)

1) This prescaler is used to define the clock period for timer 4 with T4C0 and T4C1 (T45CNT: FE3C, bits 4 and 5).

Table 3.9.1 Timer 4 Count Clocks

T4C1	T4C0	T4 Count Clock
0	0	Timer 4 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 4 period setting register (T4R) (8-bit register)

1) This register defines the period of timer 4.
2) When data is written into T4R while timer 4 is running, both the timer 4 prescaler and counter are cleared and start counting again.

3.9.3.5 Timer 5 counter (T5CTR) (8-bit counter)

1) The timer 5 counter counts the number of clocks from the timer 5 prescaler (T5PR). The value of the timer 5 counter is reset to 0 on the next clock that reaches the value specified in the timer 5 period setting register (T5R), when the interrupt flag (T5OV) is set.
2) When T5C0 and T5C1 (T45CNT: FE3C, bits 6 and 7) are set to 0, the timer 5 counter stops at a count value of 0. In other cases, the timer 5 counter continues operation.
3) When data is written into T5R while timer 5 is running, both the timer 5 prescaler and counter are cleared and start counting again..

3.9.3.6 Timer 5 prescaler (T5PR) (6-bit counter)

1) This prescaler is used to define the clock period for timer 5 with T5C0 and T5C1 (T45CNT: FE3C, bits 6 and 7).

Table 3.9.2 Timer 5 Count Clocks

T5C1	T5C0	T5 Count Clock
0	0	Timer 5 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 5 period setting register (T5R) (8-bit register)

1) This register defines the period of timer 5.
2) When data is written into T5R while timer 5 is running, both the timer 5 prescaler and counter are cleared and start counting again..

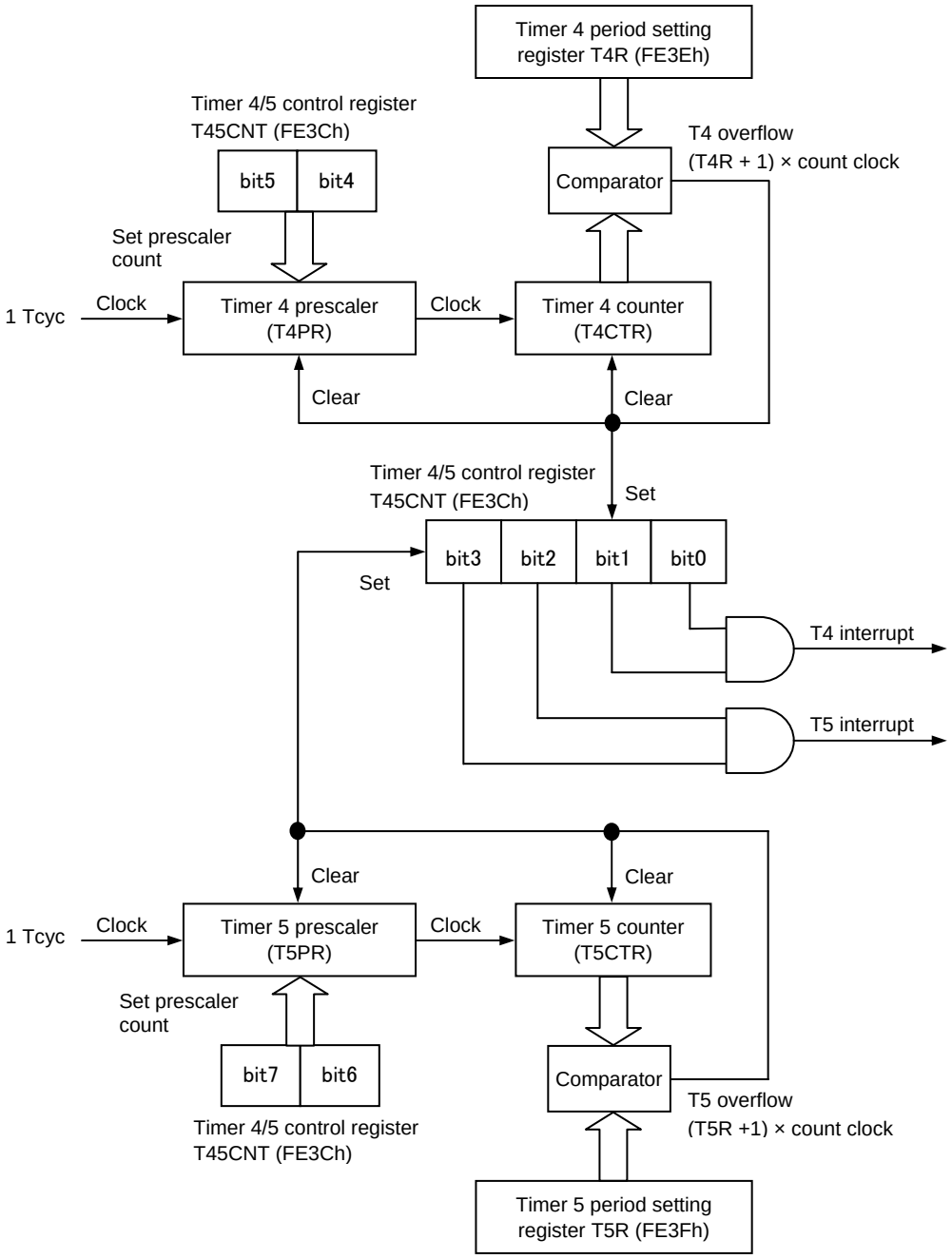


Figure 3.9.1 Timer 4/5 Operation Block Diagram

3.9.4 Related Registers

3.9.4.1 Timer 4/5 control register (T45CNT)

1) This register is an 8-bit register that controls the operation and interrupts of T4 and T5.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE3C	0000 0000	R/W	T45CNT	T5C1	T5C0	T4C1	T4C0	T5OV	T5IE	T4OV	T4IE

T5C1 (bit 7): T5 count clock control

T5C0 (bit 6): T5 count clock control

T5C1	T5C0	T5 Count Clock
0	0	Timer 5 prescaler and timer/counter are stopped in the reset state.
0	1	4 Tcyc
1	0	16 Tcyc
1	1	64 Tcyc

T4C1 (bit 5): T4 count clock control

T4C0 (bit 4): T4 count clock control

T4C1	T4C0	T4 Count Clock
0	0	Timer 4 prescaler and timer/counter are stopped in the reset state.
0	1	4 Tcyc
1	0	16 Tcyc
1	1	64 Tcyc

T5OV (bit 3): T5 overflow flag

This flag is set at the interval of the timer 5 period when timer 5 is running.

This flag must be cleared with an instruction.

T5IE (bit 2): T5 interrupt request enable control

An interrupt request to vector address 004BH is generated when this bit and T5OV are set to 1.

T4OV (bit 1): T4 overflow flag.

This flag is set at the interval of the timer 4 period when timer 4 is running.

This flag must be cleared with an instruction.

T4IE (bit 0): T4 interrupt request enable control

An interrupt request to vector address 004BH is generated when this bit and T4OV are set to 1.

3.9.4.2 Timer 4 period setting register (T4R)

1) This register is an 8-bit register for defining the period of timer 4.

Timer 4 period = (T4R value+1) × Timer 4 prescaler value

(4, 16 or 64 Tcyc)

2) When data is written into T4R while timer 4 is running, both the timer 4 prescaler and counter are cleared and start counting again.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE3E	0000 0000	R/W	T4R	T4R7	T4R6	T4R5	T4R4	T4R3	T4R2	T4R1	T4R0

3.9.4.3 Timer 5 period setting register (T5R)

1) This register is an 8-bit register for defining the period of timer 5.

   Timer 5 period = (T5R value+1) × Timer 5 prescaler value

   (4, 16 or 64 Tcyc)

2) When data is written into T5R while timer 5 is running, both the timer 5 prescaler and counter are cleared and start counting again.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE3F	0000 0000	R/W	T5R	T5R7	T5R6	T5R5	T5R4	T5R3	T5R2	T5R1	T5R0

3.10 Timers 6 and 7 (T6, T7)

3.10.1 Overview

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

3.10.2 Functions

1) Timer 6 (T6)

Timer 6 is an 8-bit timer that runs on either 4Tcyc, 16Tcyc, or 64Tcyc clock. It can generate toggle waveforms with the period of timer 6 at pin P06.

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

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 toggle waveforms with the period of timer 7 at pin P07.

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

Tcyc = Period of cycle clock

3) Interrupt generation

An interrupt request 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) It is necessary to manipulate the following special function registers to control timer 6 (T6) and timer 7 (T7).

- 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

3.10.3 Circuit Configuration

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

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

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

1) The timer 6 counter counts the number of clocks from the timer 6 prescaler (T6PR). The value of the timer 6 counter (T6CTR) is reset to 0 on the next clock that reaches 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 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.

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

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

Table 3.10.1 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

3.10.3.4 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.

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

1) The timer 7 counter counts the number of clocks from the timer 7 prescaler (T7PR). The value of the timer 7 counter (T7CTR) is reset to 0 on the next clock that reaches 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 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.

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

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

Table 3.10.2 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

3.10.3.7 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.

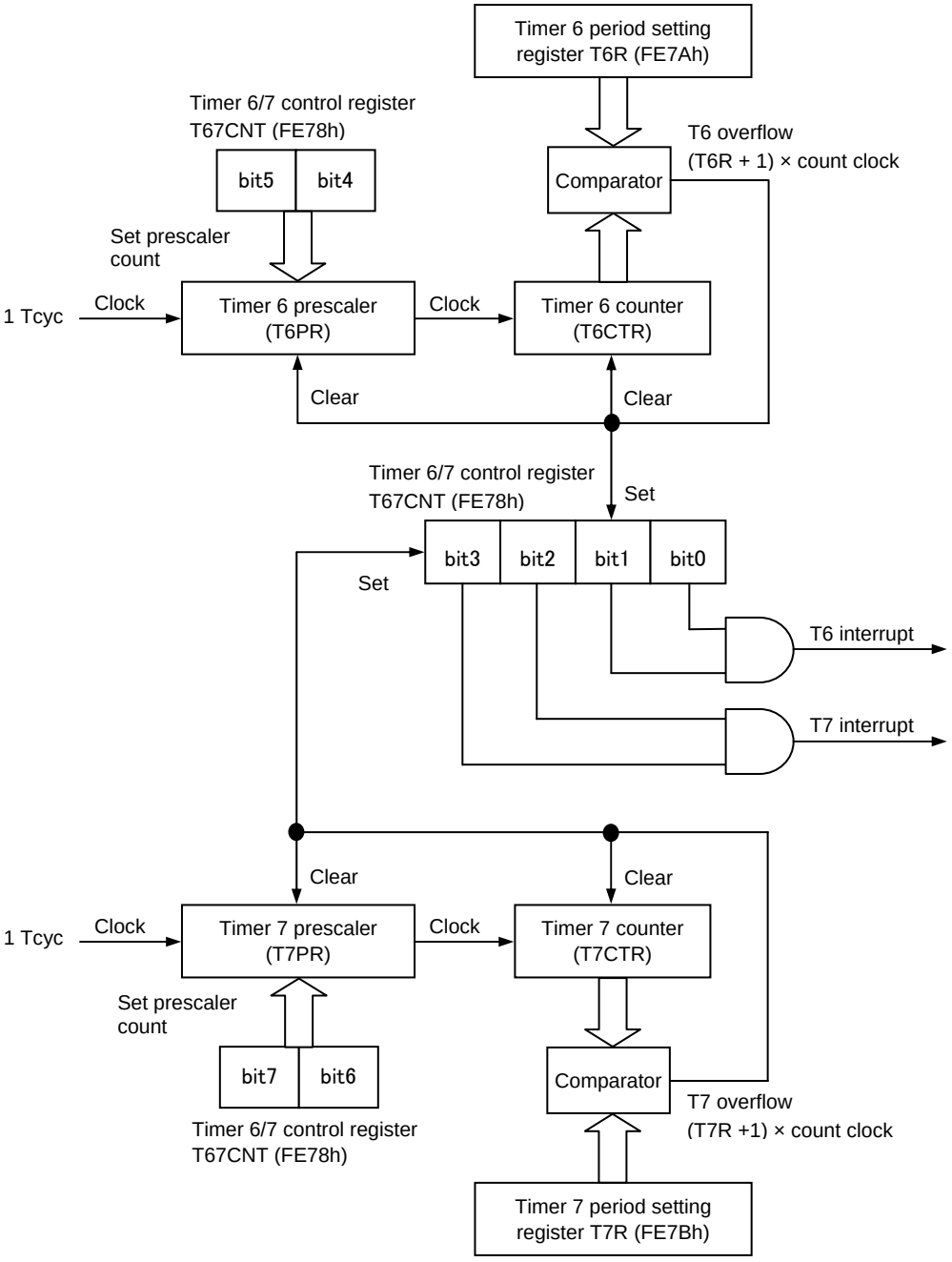


Figure 3.10.1 Timer 6/7 Operation Block Diagram

3.10.4 Related Registers

3.10.4.1 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 the 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 the 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.

3.10.4.2 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

3.10.4.3 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

3.10.4.4 Port 0 function control register (P0FCRU)

1) P0FCRU is an 8-bit register used to control the multiplexed output of port 0 pins. It controls the toggle outputs of timers 6 and 7.

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

This flag is disabled when pin P07 is set to input mode (P07DDR=0).

When pin P07 is set to output mode (P07DDR=1):

0: Outputs the value of port data latch.

1: Outputs the OR of the value of the port data latch and the waveform that toggles at the interval of the timer 7 period.

T6OE (bit 6):

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

This flag is disabled when pin P06 is set to input mode (P06DDR=0).

When pin P06 is set to output mode (P06DDR=1):

0: outputs the value of the port data latch.

1: Outputs the OR of the value of the port data latch and the waveform that toggles at the interval of the timer 6 period.

SCKOSL5 (bit 5):

SCKOSL4 (bit 4):

CLKOEN (bit 3):

CKODV2 (bit 2):

CKODV1 (bit 1):

CKODV0 (bit 0):

These 6 bits have nothing to do with the control functions of timers 6 and 7. See the description of port 0 for details on these bits.

3.11 Base Timer (BT)

3.11.1 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 (with a programmable prescaler)
3) High-speed mode (when used as a 6-bit base timer)
4) Buzzer output
5) X'tal Hold mode release

3.11.2 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.

   One of the four clocks, i.e., cycle clock, timer/counter 0 prescaler output, subclock, and low-speed RC oscillator clock, is selected with the input signal select register (ISL) as the base timer count clock.

2) 14-bit binary up-counter (with a programmable prescaler)

   A 14-bit binary up-counter can be constructed using an 8-bit binary up-counter which runs on the clock output from the 8-bit programmable prescaler 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 is specified using the base timer control register (BTCR).

4) Buzzer output function

   The signal generated by frequency-dividing the output clock of the programmable prescaler by 16 can be used as the buzzer signal. The buzzer output signals are controlled using the input signal select register (ISL). The buzzer output is ORed with timer 1 PWMH 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) X'tal HOLD mode operation and X'tal HOLD mode release function

   The base timer is enabled for operation in X'tal HOLD mode by selecting the subclock or low-speed RC oscillator clock as the base timer count clock source and setting bit 2 of the power control register (PCON). X'tal HOLD mode can be released by an interrupt from the base timer. This function allows the microcontroller to perform low-current intermittent operations.

7) It is necessary to manipulate the following special function registers to control the base timer.
   - BTCR, BTPRR, ISL, OCR, OCR3, T0PRR, 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
FE3D	0000 0000	R/W	BTPRR	BTPRR7	BTPRR6	BTPRR5	BTPRR4	BTPRR3	BTPRR2	BTPRR1	BTPRR0
FE5F	0000 0000	R/W	ISL	ST0HCP	ST0LCP	BTIMC1	BTIMC0	BUZON	NFSL1	NFSL0	ST0IN

3.11.3 Circuit Configuration

3.11.3.1 8-bit programmable prescaler

1) This prescaler is an 8-bit programmable prescaler that uses the signal selected by the input signal select register (ISL) as its clock source. A match signal is generated when a match occurs between its output value and the value of the programmable prescaler match register (BTPRR). This match signal serves as the clock input to the binary up-counter in the following stage.
2) This counter is reset under the conditions: BTON (BTCR, bit 6) is set to 0 (stop base timer operation), a match signal is generated, data is loaded into the BTPRR, and the microcontroller is in HOLD mode.

3.11.3.2 8-bit binary up-counter

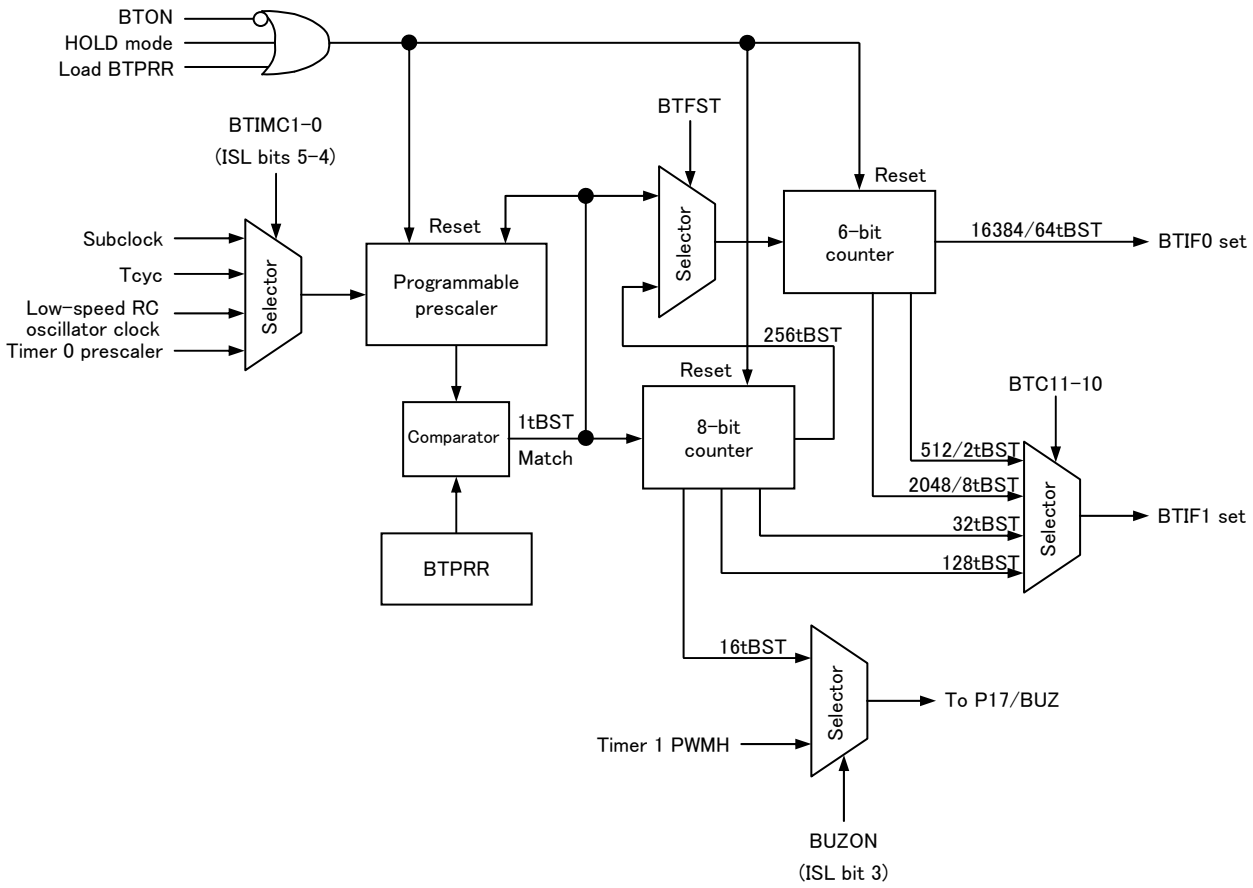
1) This counter is an 8-bit binary up-counter that uses, as its clock source, the match signal from the programmable prescaler. Its output is used as the buzzer output signal or used to set the base timer interrupt 1 flag. The overflow of this counter serves as the clock input to the 6-bit binary up-counter.
2) This counter is reset under the conditions: BTON (BTCR, bit 6) is set to 0 (stop base timer operation), data is loaded into the BTPRR, and the microcontroller is in HOLD mode.

3.11.3.3 6-bit binary up-counter

1) This counter is a 6-bit binary up-counter that uses, as its clock source, the match signal from the programmable prescaler or overflow output from the 8-bit binary up-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).
2) This counter is reset under the conditions: BTON (BTCR, bit 6) is set to 0 (stop base timer operation), data is loaded into the BTPRR, and the microcontroller is in HOLD mode.

3.11.3.4 Base timer input clock source

1) The clock input to the base timer can be selected from among four clock sources, i.e., cycle clock, timer/counter 0 prescaler output, subclock, and low-speed RC oscillator clock via the input signal select register (ISL).



*** tBST: Base timer input clock period selected by (BTPRR value + 1) × BTIMC1-0 (ISL bits 5 to 4)**

Figure 3.11.1 Base Timer Block Diagram

3.11.4 Related Registers

3.11.4.1 Base timer control register (BTCR)

1) This 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.

When this bit 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.

When 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 high-speed mode is to be used.

tBST: Period of the input clock to the base timer that is selected by (BTPRR value + 1) × BTIMC1 to 0 (ISL, bits 5 to 4)

BTON (bit 6): Base timer operation control

When this bit is set to 0, the base timer stops operation at the count value of 0.

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
0	0	1	16384tBST	128tBST
0	1	0	16384tBST	512tBST
0	1	1	16384tBST	2048tBST
1	0	0	64tBST	32tBST
1	0	1	64tBST	128tBST
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 of 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 an 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 of the base timer interrupt 0 period that is defined by BTFST.

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 an 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 period so that the period of the cycle clock (Tcyc) and the base timer interrupt period satisfy the following relationship:*

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

  Since program processing (e.g., interrupt processing routine) is involved in practice, the time that is required to execute such processing should be taken into consideration when setting up the optimum interrupt period.
- *Note that there are cases BTIF1 is set to 1 if BTC11 or BTC10 is rewritten when the base timer is active.*
- *If the crystal oscillator (subclock) is selected as the base timer clock source, erroneous counting can occur in the base timer because oscillation stabilization time cannot be secured when HOLD mode is exited. It is therefore recommended that measures be taken to stop the base timer before placing the CPU in HOLD mode.*

  (See Section 4.2, "System Clock Generator Function," for the state of the oscillator circuits in HOLD mode.)
- *Counting errors can occur in the base timer if the base timer clock source is changed (change ISL, bits 5 and 4) while the base timer is running. Be sure to stop the base timer in advance when changing the base timer clock source.*

3.11.4.2 Base timer programmable prescaler match register (BTPRR)

1) This register is an 8-bit register that sets the clock period (tBST) of 8-bit/6-bit binary up-counter.
2) When data is loaded into BTPRR, the prescaler and the binary up-counter are reset to the count value 0.
3) tBST = (BTPRR value + 1) × base timer input clock period

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE3D	0000 0000	R/W	BTPRR	BTPRR7	BTPRR6	BTPRR5	BTPRR4	BTPRR3	BTPRR2	BTPRR1	BTPRR0

3.11.4.3 Input signal select register (ISL)

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

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	NFSL1	NFSL0	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 of 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 (Note 1)
0	1	Cycle clock
1	0	Low-speed RC oscillator clock (Note 2)
1	1	Timer/counter 0 prescaler output

Note 1: Set Bit 6 (EXTOSC) of the OCR register to 1.

Note 2: Set bit 0 (SRCSTART) of the OCR3 control register to 1.

BUZON (bit 3): Buzzer output / timer 1 PWMH output select

This bit selects data (buzzer output/timer 1 PWMH) to be transferred to port P17 when P17FCR (P1FCR, bit7) is set to 1.

When this bit is set to 1, timer 1 PWMH output is fixed at a high level, and a signal that is obtained by dividing the base timer clock is sent to port P17 as the buzzer output.

When this bit is set to 0, the buzzer output is fixed at a high level, and timer 1 PWMH output data is sent to port P17.

NFSL1 (bit 2): Noise filter sampling clock select

NFSL0 (bit 1): Noise filter sampling clock 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.

3.12 Serial Interface 0 (SIO0)

3.12.1 Overview

The serial interface 0 (SIO0) incorporated in this series of microcontrollers has the following two functions:

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

3.12.2 Functions

1) Synchronous 8-bit serial I/O
   - Performs 2- or 3-wire synchronous serial communication. Either an internal or external clock may be used.
   - 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 clock synchronization mode. Either an internal or external clock may 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 transmit 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) It is necessary to manipulate the following special function registers to control serial interface 0 (SIO0).
   - 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

3.12.3 Circuit Configuration

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

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

3.12.3.2 SIO0 data shift register (SBUF0) (8-bit register)

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

3.12.3.3 SIO0 baudrate generator (SBR0) (8-bit reload counter)

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).

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

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

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

1) This register controls the suspension and resumption of serial transfer in byte units in continuous data transmission/reception mode.
2) It allows the application program to read the number of bytes transferred in continuous data transmission/reception mode.

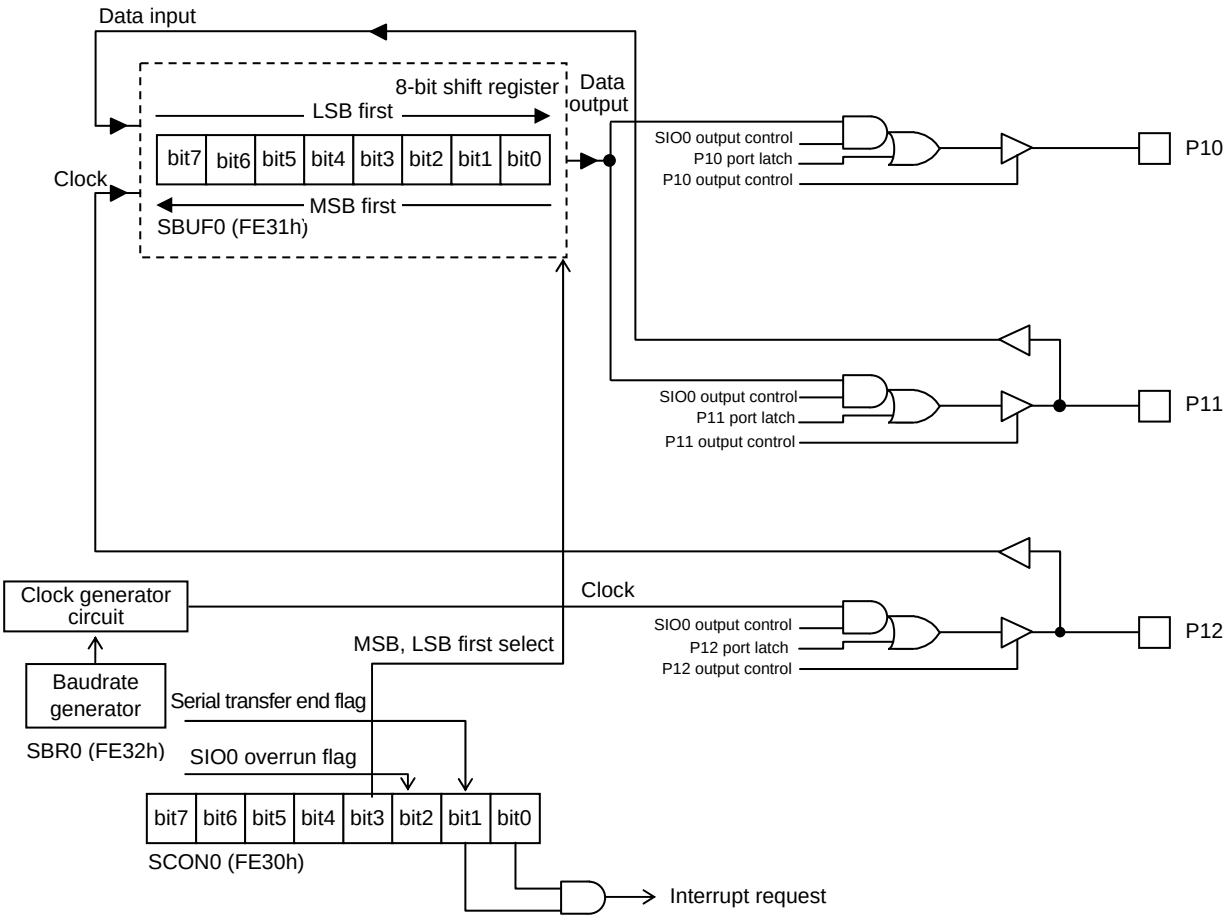


Figure 3.12.1 SIO0 Synchronous 8-bit Serial I/O Block Diagram (SI0CTR=0)

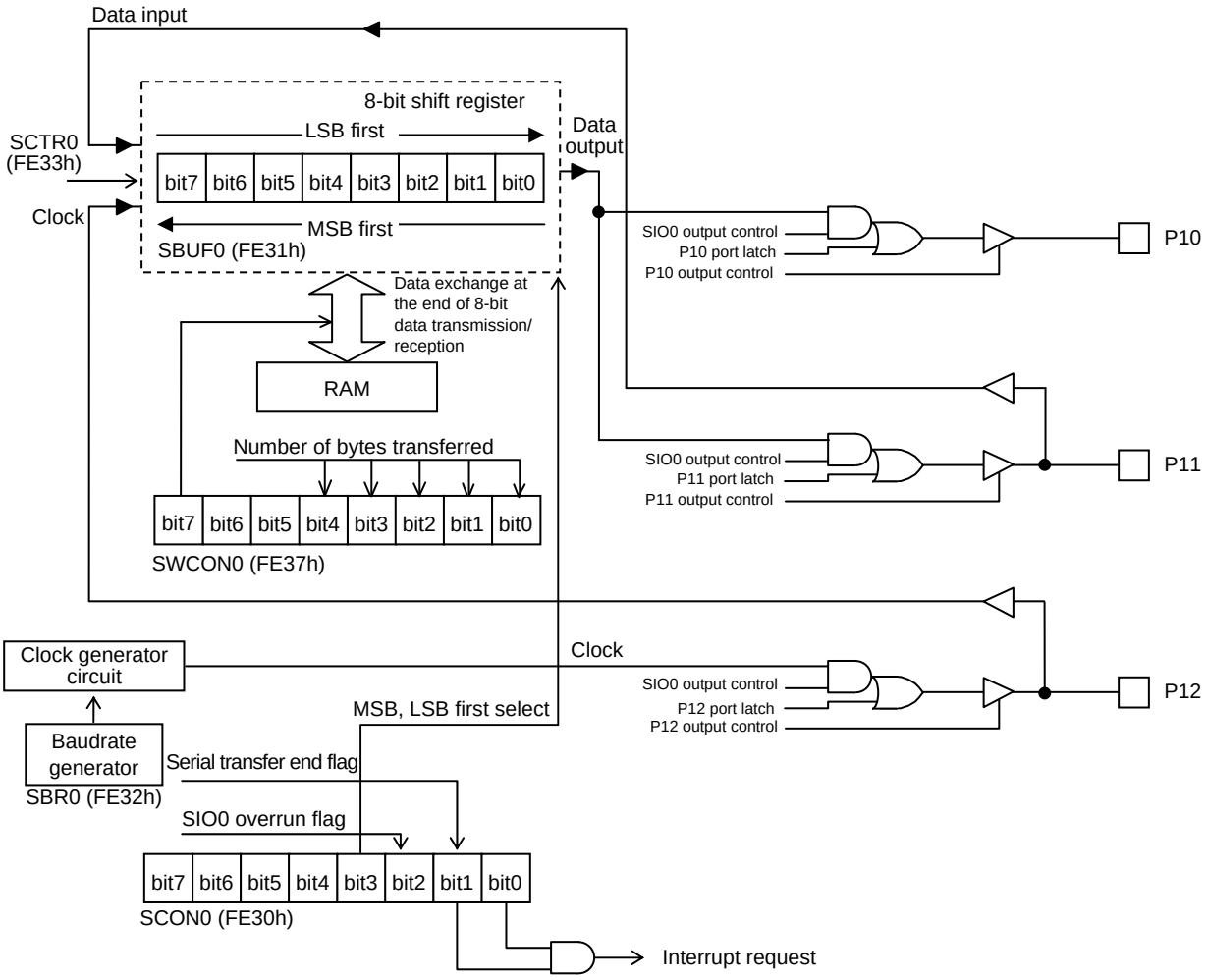


Figure 3.12.2 SIO0 Continuous Data Transmission/Reception Mode Block Diagram (SI0CTR=1)

3.12.4 Related Registers

3.12.4.1 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> When this bit is set to 1, SIO0 operates in continuous data transmission/reception mode.

<2> When this bit is set to 0, SIO0 operates in 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 selects MSB first.

<2> A 0 in this bit selects LSB first.

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 after each 8-bit transfer in continuous data transmission/reception mode.

<3> Read this bit and determine if the communication is performed normally at the end of the communication.

<4> This bit must be cleared with an instruction.

SI0END (bit 1): Serial transfer end 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.

3.12.4.2 SIO0 data shift register (SBUF0)

1) This register is an 8-bit shift register for SIO0 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

3.12.4.3 Baudrate generator register (SBR0)

1) This register is an 8-bit register that defines the transfer rate of an SIO0 serial transfer.
2) The transfer rate is computed as follows:

   TSBR0 = (SBR0 value + 1) $\times \frac{2}{3}$ 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 inhibited.*

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

3.12.4.4 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 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 are 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

3.12.4.5 Continuous data transfer control register (SWCON0)

1) This register is used to suspend or resume the operation of SIO0 in byte units in 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 of data in 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 to S0XBYT0 (bits 4 to 0):

These bits can be read to determine the number of bytes transferred in continuous data transfer mode.

3.12.4.6 RAM used in continuous data transmission/reception mode

SIO0 can transmit and receive 1 to 256 bits of serial data in 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 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 is 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.

3.12.5 SIO0 Communication Examples

3.12.5.1 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 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 transmission mode).
   - Clear SI0END.
   - Return to step 4) when continuing the communication.

3.12.5.2 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
   - Write the output data of the specified bit length to data RAM at the specified address in data transmission or data transmission/reception mode.
   - Write 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 continuing the communication.

3.12.6 SIO0 HALT Mode Operation

3.12.6.1 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.

3.12.6.2 Continuous data transmission/reception mode

1) SIO0 operation is suspended immediately before the contents of RAM and SBUF0 are exchanged when HALT mode is entered in continuous data transmission/reception mode. After HALT mode is entered, SIO0 operation continues until immediately before the contents of the 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.

3.13 Serial Interface 1 (SIO1)

3.13.1 Overview

The serial interface 1 (SIO1) incorporated in this series of microcontrollers has the following four functions:

1) Mode 0: Synchronous 8-bit serial I/O (2- or 3-wire system, 2 to 512 Tcyc transfer clock)
2) Mode 1: Asynchronous serial (half-duplex, 8 data bits, 1 stop bit, 8 to 2048 Tcyc baudrate)
3) Mode 2: Bus-master (start bit, 8 data bits, 2 to 512 Tcyc transfer clock)
4) Mode 3: Bus-slave (start detection, 8 data bits, stop detection)

3.13.2 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 period 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 output of acknowledge require program intervention.
   - SIO1 can generate an interrupt by forcing the clock line to a low level on the falling edge of the 8th clock for recognition by a program.

5) Interrupt generation

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

6) It is necessary to manipulate the following special function registers to control the serial interface 1 (SIO1).

   - 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

3.13.3 Circuit Configuration

3.13.3.1 SIO1 control register (SCON1) (8-bit register)

1) This register controls the operation and interrupts of SIO1.

3.13.3.2 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 directly accessed with an instruction. It is accessed via SBUF1.

3.13.3.3 SIO1 data register (SBUF1) (9-bit register)

1) The low-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 low-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, etc.

3.13.3.4 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 3.13.1 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 1's)	8 (Shift data)	8 (All 1's)	8 (Shift data)	8 (All 1's)	8 (Shift data)	8 (All 1's)
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 value = H and data port value = L are detected on the rising edges of the 1st to 8th clocks, the microcontroller recognizes a bus contention loss and clears SI1RUN (and also stops sending the clock at the same time).

(Continued on next page)

Table 3.13.1 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 data 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	←

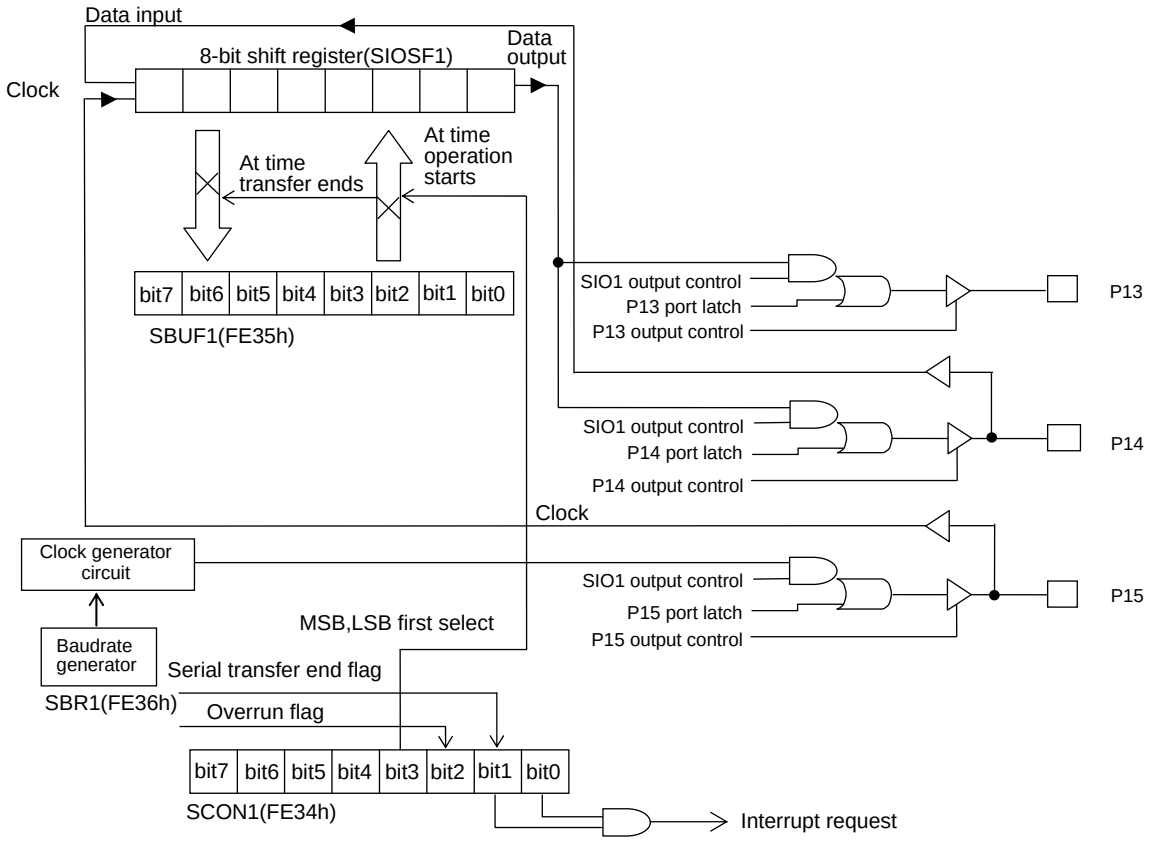


Figure 3.13.1 SIO1 Mode 0: Synchronous 8-bit Serial I/O Block Diagram (SI1M1=0, SI1M0=0)

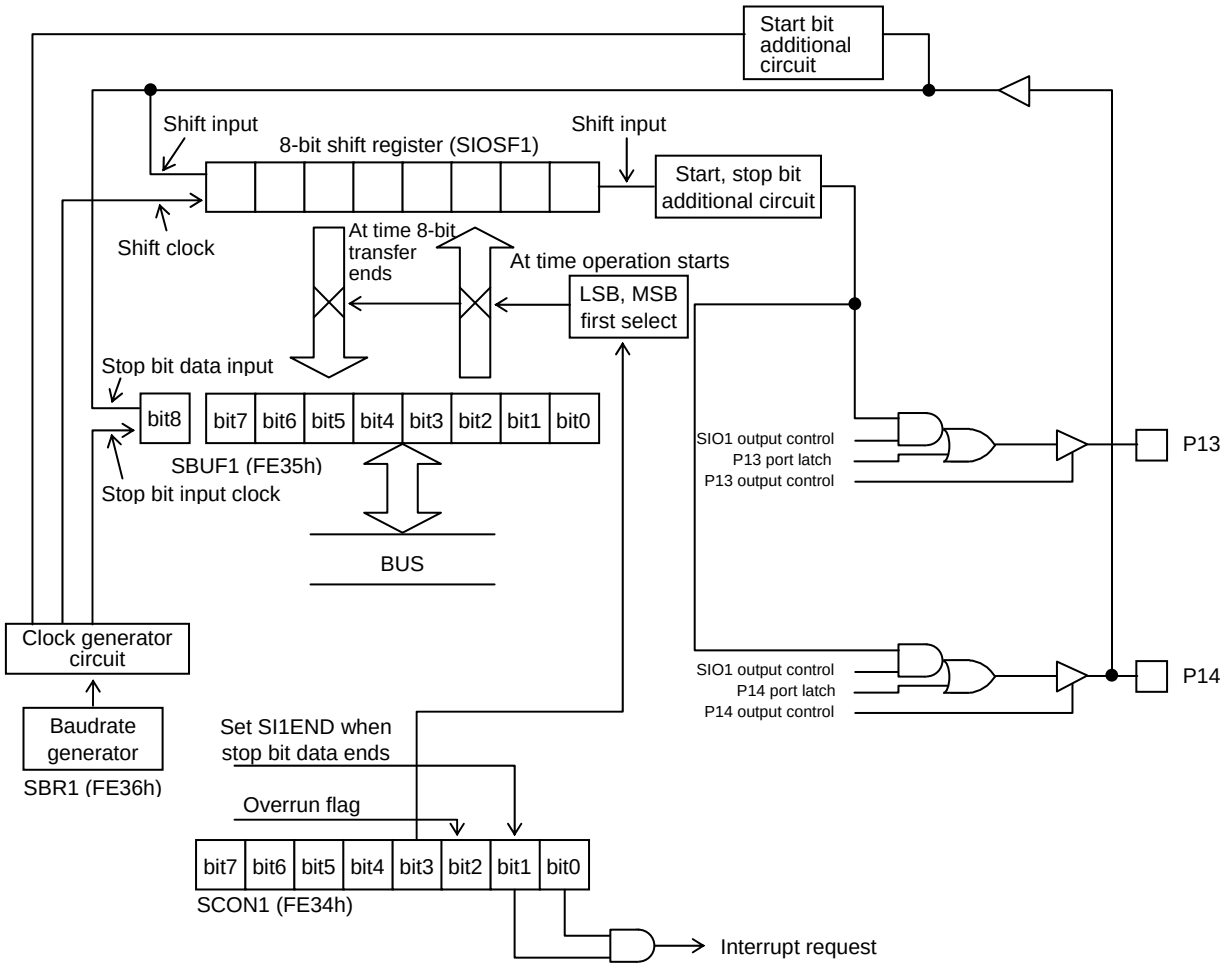


Figure 3.13.2 SIO1 Mode 1: Asynchronous Serial [UART] Block Diagram (SI1M1=0, SI1M0=1)

3.13.4 SIO1 Communication Examples

3.13.4.1 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 transmission mode).
   - Clear SI1END and exit interrupt processing.
   - Return to step 4) when repeating processing.

3.13.4.2 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 transmit operation
   - 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 detected 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 transmission is 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 from the data I/O port even in transmission mode. When SBUF1 is read in, the data on the position of the stop bit is read into bit 1 of the PSW.)
   - Clear SI1END and exit interrupt processing.
   - Return to step 4) to repeat processing.

Note: Make sure that the following conditions are met when performing continuous receive operation 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.*

3.13.4.3 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
   - Configure the clock port (P15) and data port (P14) as N-channel open drain output ports by setting the option.
   - Set P14 (P1, bit 4) and P15 (P1, bit 5) to 0.
   - Set P14FCR (P1FCR, bit 4) and P15FCR (P1FCR, bit 5) to 1.
   - Set P14DDR (P1DDR, bit 4) and P15DDR (P1DDR, bit 5) to 1.

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 transmission mode. When SBUF1 is read in, the data on 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 3.13.1), 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, for example, when another device in master mode is in the system, perform timeout processing using a timer module, etc. and detect the condition.

6) Sending data
   - Load SBUF1 with output data.
   - Clear SI1END and exit interrupt processing (transfer SBUF1 (8 bits) + stop bit (H)).

7) Checking transmission data (after an interrupt)
   - Read SBUF1. (SBUF1 has been loaded with serial data from the data I/O port even in transmission mode. When SBUF1 is read in, the data on 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 3.13.1), 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, for example, when another device in master mode is in the system, perform timeout processing using a timer module, etc. and detect the condition.
   - Return to step 6) to continue 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) + output SBUF1, bit 8 (acknowledge)).

9) Reading received data (after an interrupt)
   - Read SBUF1.
   - Return to step 8) to continue data reception.
   - Go to * in step 10) to terminate processing. At this moment, SBUF1,bit 8 data has already been output 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 to the original state (P15FCR=1, P15DDR=1, P15=0) and release the clock output.

* - Wait for all slaves to release the clock and for 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 to 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 continue processing.

3.13.4.4 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
   - Configure the clock port (P15) and data port (P14) as N-channel open drain output ports by setting the option.
   - Set P14 (P1, bit 4) and P15 (P1, bit 5) to 0.
   - Set P14FCR (P1FCR, bit 4) and P15FCR (P1FCR, bit 5) to 1.
   - Set P14DDR (P1DDR, bit 4) and P15DDR (P1DDR, bit 5) to 1.

4) Starting communication (waiting for an address)

*1 • Set SI1REC.

*2 • SI1RUN is automatically set on detection of a start bit.

• 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.

5) Checking address data (after an interrupt)

• When a start condition is detected, SI1OVR is set. Check SI1RUN=1 and SI1OVR=1 to determine if the address has been received.

(SI1OVR is not automatically cleared. Clear it by software.)

• 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 * in 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 + 1/3) × 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. However, the clock counter is cleared if a start condition is detected in the middle of receive processing, in which 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 the 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 receive operation and release the clock port after the lapse of (SBR1 value + 1/3) × 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 the 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 + 1/3) × Tcyc.)

• Return to *1 in step 7) if an acknowledge from the master is present (L).

• If there is no acknowledge from the master (H), SIO1, recognizing the end of data transmission, automatically clears SI1RUN and releases the data port.

* However, if the restart condition occurs 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 the slave transmission (when SI1REC is not set to 1 by software).

*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 automatically terminate communication.
   - To force communication to terminate, 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).

3.13.5 Related Registers

3.13.5.1 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 3.13.2 SIO1 Operating Modes

Mode	SI1M1	SI1M0	Operating 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 3.13.1 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 receive mode.

<2> Setting this bit to 0 places SIO1 into transmit mode.

SI1DIR (bit 3): MSB/LSB first select

<1> A 1 in this bit selects MSB first.

<2> A 0 in this bit selects LSB first

SI1OVR (bit 2): SIO1 overrun flag

<1> This bit is set when the falling edge of the input clock is detected with SI1RUN =0 in modes 0, 1, and 3.

<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): Serial transfer end flag

<1> This bit is set when serial transfer terminates (see Table 3.13.1).

<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.

3.13.5.2 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 low-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 low-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 on 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

3.13.5.3 Baudrate generator register (SBR1)

1) This register is an 8-bit register that defines the baudrate of the SIO1. (Modes 0, 1, 2)
2) Loading this register with data causes the baudrate generating counter to be initialized immediately.
3) The baudrate varies from mode to mode.

   Modes 0 and 2: TSBR1 = (SBR1 value + 1) × 2 Tcyc
   (Value range = 2 to 512 Tcyc)

   Mode 1: TSBR1 = (SBR1 value + 1) × 8 Tcyc
   (Value range = 8 to 2048 Tcyc)
4) When in mode 3, it sets the acknowledge data setup time (See 3.13.4.4 6), 7)). When setting to mode 3, time that clock port is released after SI1END is cleared is

   (SBR1 value + 1/3) × Tcyc (SBR1=0 is inhibited)

   Set this value to meet the opponent device's data setup time.

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

3.14 Serial Interface 4 (SIO4)

3.14.1 Overview

The serial interface 4 (SIO4) incorporated in this series of microcontrollers is a synchronous serial interface that has the following functions:

1) Continuous synchronous data transfer
   - Data transfer of any number of bytes between 1 and 8192 bytes
   - Transfer clock period (master operation): 4/3 to 1020/3 Tcyc
2) 16-bit CRC code calculation

3.14.2 Functions

1) Continuous synchronous data transfer
   - Performs 2- or 3-wire synchronous serial communication. The clock may be an internal clock (master operation) or external clock (slave operation).
   - The period of the internal clock (master operation) is programmable within the range of 4n/3 Tcyc (n= 1 to 255; Note: n=0 is inhibited).
   - Transmits and receives 1 to 8192 of arbitrary byte data 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, suspend/resume 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

     Ports P22 to P24 are used for serial communication.

Port	I/O	Pin	Function
P22	I/O	SO4	Serial data I/O pin
P23	I/O	SI4	Serial data I/O pin
P24	I/O	SCK4	Synchronous clock I/O pin

2) Interrupt generation

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

3) It is necessary to manipulate the following special function registers to control serial interface 4 (SIO4).
   - S4ADRL, S4BYTH, CRCL, CRCH, CRCCNT, SI4CN0, SI4CN1, SI4BUF, S4BAUD, S4ADDR, S4BYTE
   - 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

3.14.3 Circuit Configuration

3.14.3.1 SIO4 transfer RAM address register low byte (S4ADRL) (8-bit register)

1) This register defines the starting address of the RAM area to be used for data transfer.

3.14.3.2 SIO4 transfer data byte register high byte (S4BYTH) (8-bit register)

1) This register defines the number of data bytes to be transferred via the SIO4 in continuous data transfer mode.

3.14.3.3 CRC (Cyclic Redundancy Check) registers (CRCL, CRCH) (8-bit register)

1) These registers are used to define the generator polynomial for cyclic redundancy check (CRC) encoding.

3.14.3.4 CRC calculation result register (CRC16) (16-bit register)

1) This register stores the calculation results of CRC encoding.

3.14.3.5 CRC control register (CRCCNT) (8-bit register)

1) This register controls the CRC operation.
2) This register controls the suspension of the ports in continuous data transfer mode.

3.14.3.6 SIO4 control register 0 (SI4CN0) (8-bit register)

1) This register controls the operation and interrupts of SIO4.

3.14.3.7 SIO4 control register 1 (SI4CN1) (8-bit register)

1) This register controls the SIO4 interface port.

3.14.3.8 SIO4 shift register (SI4BUF) (8-bit shift register)

1) This register is an 8-bit shift register used for SIO4 serial transfer.

3.14.3.9 SIO4 baudrate register (S4BAUD) (8-bit reload register)

1) This register is a reload counter for generating internal clocks.
2) It can generate a clock with period of 4n/3 Tcyc (n=1 to 255; Note: n=0 is inhibited).

3.14.3.10 SIO4 transfer RAM address register high byte (S4ADDR) (8-bit register)

1) This register defines the starting address of the RAM area to be used for data transfer.

3.14.3.11 SIO4 transfer data byte register low byte (S4BYTE) (8-bit register)

1) This register defines the number of data bytes to be transferred in continuous data transfer mode.

3.14.4 Related Registers

3.14.4.1 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 consists 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 generator polynomial $G(x) = X^{16} + X^{12} + X^5 + 1$ is shown below.

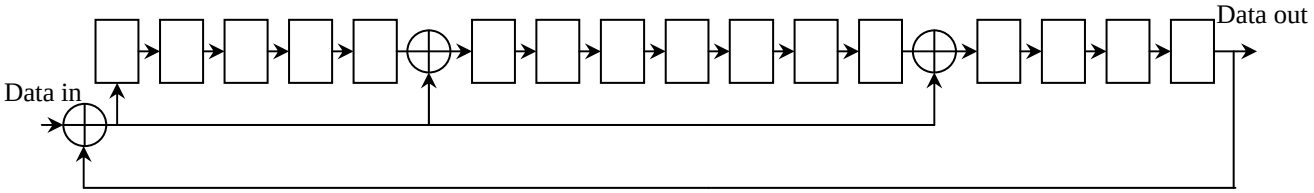


Figure 3.14.1 CRC Encoding/Decoding Circuit

In this example, the 16-bit CRC register (CRCH and CRCL) is set as follows:

CRCH = 10[H], CRCL = 21[H]

3) The results of CRC calculation can be read from the CRC register (CRCH and CRCL) by setting bit 5 (CRCRD) of the CRC control register (CRCCNT) to 1.

3.14.4.2 CRC control register (CRCCNT)

1) This register controls cyclic redundancy check (CRC) operation
2) This register controls suspension of ports in 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 result register are retained.

0: The CRC result register is initialized.

CRCRD (bit 5): CRC results read control flag

1: The CRC results are read from the CRC register.

0: The generator polynomial is read from the CRC register.

1/0SEL (bit 4): CRC result register initialization control flag

1: All CRC result register bits are initialized to 1.

0: All CRC result register bits are initialized to 0.

S4STPCEN (bit 3): Suspension port control enable flag

1: Enables suspension port control (in 1- or 2-byte units) according to the level (H or L level) of the port (P70 to P73) in continuous transfer mode.

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.

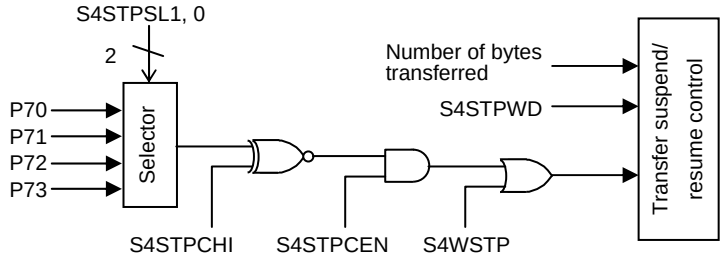
0: Transfer is suspended when the port (P70 to P73) is set to the low level and resumed when the port is set to the high level.

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



3.14.4.3 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.
- When SI4RUN is set to 1 in internal clock operating mode (master operation), the transmission of the clock from the SCK4 pin and loading of the serial input data into the shift register are started regardless of the setting of SBITON.

0: Stops transfer.

SBITON (bit 6): Automatic transfer (after start bit detection) control flag

1: Sets the serial data transfer control flag (SI4RUN) automatically on detection of the falling edge of the serial input data.

- Even when SBITON is set to 1 with SI4RUN set to 0 in internal clock operating mode (master operation), no clock is transmitted from the SCK4 pin until a falling edge of input serial data is detected and SI4RUN is automatically set.

0: Does nothing for the automatic transfer setting.

MSBSEL (bit 5): MSB/LSB transfer direction control flag

1: MSB transfer

0: LSB transfer

S4RAM (bit 4): Selects the starting RAM address

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 input, the 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 select flag

1: Data is output on the rising edge of the clock and input on the falling edge of the clock.

0: Data is output on the falling edge of the clock and input on the rising edge of the clock.

SI4WRT (bit 2): SIO4 transmission/reception mode setting flag

1: Transmission and reception (the contents of RAM and shift register are automatically exchanged in the continuous data transfer mode.)

0: Transmission only (the contents of RAM are automatically transferred to the shift register in the continuous data transfer mode but the contents of RAM remain unchanged.)

SI4END (bit 1): SIO4 transfer end flag

This bit is automatically set at the end of SIO4 transfer.

This bit 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.

3.14.4.4 SIO4 control register 1 (SI4CN1)

1) This register is an 8-bit register that sets up the 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 (SO4 pin).

0: Serial data to SIO4 is received via P23 (SI4 pin).

P24OUT (bit 4): P24 (sync clock) I/O control flag

1: The SIO4 sync clock is output from P24 (SCK4 pin) (master operation).

0: No SIO4 sync clock is output from P24 (SCK4 pin) (slave operation).

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 (SI4 pin).

0: No SIO4 serial data is output from P23 (SI4 pin).

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 (SO4 pin).

0: No SIO4 serial data is output from P22 (SO4 pin).

Mode			SI4CN1 register							
Transmission	Reception	Clock Internal/ External	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			PARA	P1/ P0	P22/P23	P24OUT	P23MOS	P23OUT	P22MOS	P22OUT
P22 data transmission	None	Internal / External	0	*1 / —	1 (*2)	1 / 0	—	0	*1	1
P23 data transmission	None	Internal / External	0	*1 / —	0 (*2)	1 / 0	*1	1	—	0
P23&P22 data transmission	None	Internal / External	0	*1 / —	0/1 (*2)	1 / 0	*1	1	*1	1
None	P22 data reception	Internal / External	0	*1 / —	1	1 / 0	—	0	—	0
None	P23 data reception	Internal / External	0	*1 / —	0	1 / 0	—	0	—	0
P22 data transmission	P23 data reception	Internal / External	0	*1 / —	0	1 / 0	—	0	*1	1
P23 data transmission	P22 data reception	Internal / External	0	*1 / —	1	1 / 0	*1	1	—	0

**1: Set according to the output type (CMOS/N-channel open drain) selected.*

**2: Since CRC encoding is performed on the input data, select the port (P22/P23) that is set for output when performing calculation on the output data.*

3.14.4.5 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

3.14.4.6 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.

   *Tcyc=3/fSCLK

   Tcyc: Minimum instruction cycle time

   fSCLK: System clock frequency

   (Example) When fSCLK=12 MHz, Tcyc=250 ns.

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

3.14.4.7 SIO4 transfer RAM address register low byte (S4ADRL)

1) This register defines the low-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

3.14.4.8 SIO4 transfer RAM address register high byte (S4ADDR)

1) This register is used to control the suspension of continuous data transfer operation.
2) The register is used to select the ports for serial communication.
3) This register defines the high-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 operation 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 operation is suspended after the transfer of an even byte data is finished.

Suspension cannot be controlled when the SIO4 is running on an external clock.

0: Suspension is released.

S4PTSEL (bit 6): Reserved bit

This bit must always be set to 0.

S4ADR5 to S4ADR0 (bits 5 to 0): Transfer RAM start address high-order 6 bits

The low-order 6 bits of S4ADDR and the 8 bits of S4ADRL are used to define the start address of the RAM area to be used for data transfer.

S4ADDR Low-order 6 Bits [H]	S4ADRL [H]	RAM Start Address [H]
00	00	0000
00	01	0001
⋮	⋮	⋮
1F	FF	1FFF

3.14.4.9 SIO4 transfer data byte register high byte (S4BYTH)

1) This register is used to specify continuous data transfer operation to be suspended in 1- or 2-byte units.
2) This register controls reading the number of transferred data bytes.
3) This register defines the high-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-byte units/2-byte units suspension select flag

1: Continuous data transfer mode is suspended in 2-byte units.

0: Continuous data transfer mode 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 low-order 4 bits of S4BYTH and S4BYTE. If continuous data transfer is suspended with S4RAM (SI4CN0, bit 4) set to 0, however, the byte count that is read is the “number of data bytes that are transferred minus 1.” A 0 is read as the transferred byte count after the continuous data transfer operation is finished.

0: The readout of transferred byte count is disabled.

S4BYTH5 (bit 5): Reserved bit

This bit must always be set to 0.

S4BYTH4 to S4BYTH0 (bits 4 to 0): High-order 5 bits of transfer data byte count

See the next Subsection.

3.14.4.10 SIO4 transfer data byte register low byte (S4BYTE)

1) This register is used to define the low-order 8 bits of the number of data bytes to be transferred in the continuous data transfer mode.

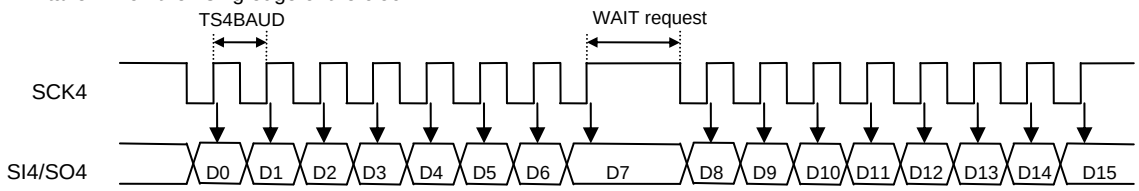
Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEE0	0000 0000	R/W	S4BYTE	S4BYT7	S4BYT6	S4BYT5	S4BYT4	S4BYT3	S4BYT2	S4BYT1	S4BYT0

The low-order 5 bits of S4BYTH and 8 bits of S4BYTE are used to define the number of data bytes to be transferred.

S4BYTH Low-order 5 bits [H]	S4BYTE [H]	Transfer Data Byte Count
0	00	1
0	01	2
⋮	⋮	⋮
1F	FF	8192

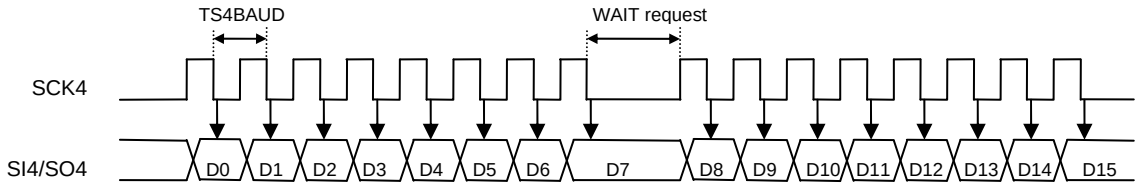
・When S4CKPL=0

SCK4 is held high when the clock is stopped, the data state is changed on the falling edge of the clock and data is taken in on the rising edge of the clock.



・When S4CKPL=1

SCK4 is held low when the clock is stopped, the data state is changed on the rising edge of the clock and data is taken in on the falling edge of the clock.



Notes:

- *In continuous data transfer mode, a wait request occurs every 8-bit data transfer and the CPU performs 1 cycle of wait operation (data transfer between RAM and shift register).*
- *If a wait request has been issued by another module (SIO0 or USB), the wait operation for the SIO4 is made pending; the wait operation for the SIO4 is carried out after the wait operation for the other module is finished (see "SIO4 Serial Input/Output Characteristics" in the Data Sheet).*
- *The CPU suspends instruction execution for 1 cycle while it is performing a wait operation.*
- *For details on the wait operation, see Section 2.12, "Wait Operation," in the User's Manual.*

Figure 3.14.2 Continuous Data Transfer Timing Chart

3.14.5 SIO4 Communication Examples

3.14.5.1 Synchronous serial interface

Example 1: Continuous data transmission (on internal clock)

1) Setting up the ports
   - [P2DDR] P24DDR=0, P23DDR=0, P22DDR=0
   - [P2] P24=0, P23=0, P22=0

2) Setting 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 from 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 address offset register [S4ADDR, S4ADRL]
   - Set the starting address of the RAM data area to be used for continuous data transmission.

8) Setting up output data
   - Transfer 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 operation
   - 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: Continuous data reception (on external clock)

1) Setting up the ports
   - [P2DDR] P24DDR=0, P23DDR=0, P22DDR=0
   - [P2] P24=0, P23=0, P22=0

2) Setting the mode
   - [SI4CN0] SI4RUN=0, SBITON=1, MSBSEL=1/0, S4RAM=0, SI4WRT=1, SI4IE=1
   - [IE] IE7=1

   ** SBITON=1: 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 P22>
   - P22/P23=1, P24OUT=0, P22OUT=0

   <For data reception from P23>
   - P22/P23=0, P24OUT=0, P23OUT=0

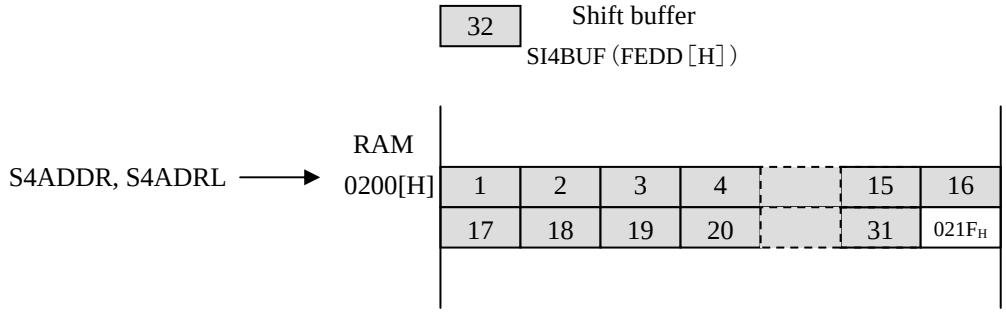
5) Setting the byte count [S4BYTH, S4BYTE]
   - Specify the number of bytes to be received continuously.

6) Setting up the SIO4 data transfer address offset register [S4ADDR, S4ADRL]
   - Set the starting address of the RAM data area to be used for continuous data reception.

7) Starting data transfer
   - When the falling edge at the serial data input port is detected, SI4RUN (SI4CN0, bit 7) is automatically set and SIO4 transfer is started.

8) End of transfer operation
   - 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 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 register and not transferred to RAM.
   - For example, when data is received with S4ADDR set to 02[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 (0200[H] to 021E[H]) and the shift buffer [SI4BUF].



Example 3: CRC (cyclic redundancy checking) calculation

1) Setting up CRC-related registers

Set the 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.*

2) to 9) Set 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

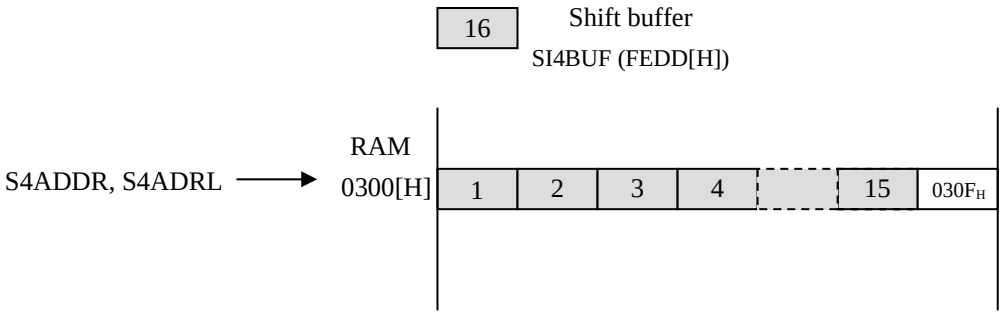
00_11_22_33_44_55_66_77_88_99_aa_bb_cc_dd_ee_ff[H]

is received with S4ADDR set to 03[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 (0300[H] to 030E[H]) and the shift buffer [SI4BUF].

11) Reading CRC results

- The results of CRC calculated on the received data are as follows:

CRCH = 12[H], CRCL = 48[H]



3.14.6 SIO4 HALT Mode Operation

1) The SIO4 suspends operation 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 operation until immediately before the contents of the first RAM and SI4BUF are exchanged. The SIO4 resumes and continues operation after the microcontroller exits HALT mode.
2) Since the SIO4 suspends operation on entry into HALT mode, HALT mode cannot be released using the interrupt to SIO4.

3.15 Parallel Interface

3.15.1 Overview

This series of microcontrollers can generate a read or write signal to external memory when an instruction for accessing a port (P0 or P1) is executed. The generation of the address needs to be set up under program control.

3.15.2 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.

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.

Port Assignment	I/O	Description
P22	Output	Read signal output pin
P23	Output	Write signal output pin

3) It is necessary to manipulate the following special function registers to control the parallel interface.

- SI4CN0, SI4CN1, SI4BUF
- 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

3.15.3 Related Registers

See Subsection 3.14.4 for a description of the special function registers (SI4CN0, SI4CN1, and SI4BUF) for controlling the parallel interface.

3.15.4 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) Initialization

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 setting 1 into the output data latch) once.

- Set SI4CN1= 00[H].
- Set SI4BUF= FF[H] (load FF[H] into the shift register).
- Set SI4CN0=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=00[H].

2) Setting the mode

- [SI4CN1]PARA=1

3) Setting up the port [SI4CN1]

- P22OUT=1, P22MOS=1 (generate the read signal)
- P23OUT=1, P23MOS=1 (generate the write signal)

4) Selecting the parallel interface data I/O port [SI4CN1]

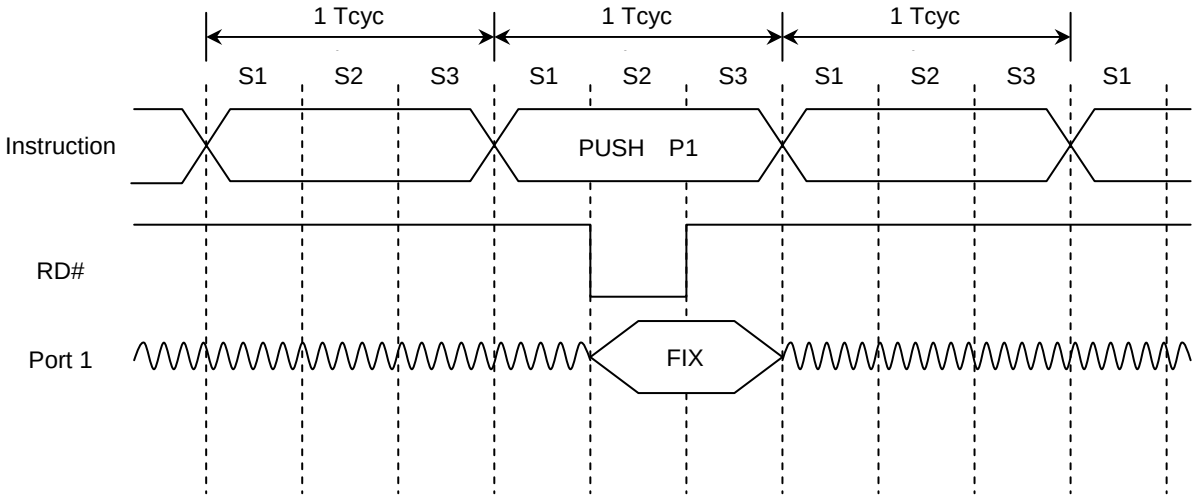
- P1/P0=1 (port 1)

5) Accessing the port

<1> External memory read mode

- Execute an instruction (e.g., PUSH) for reading data from P1 and generate a read signal at P22 at the timing of S2.

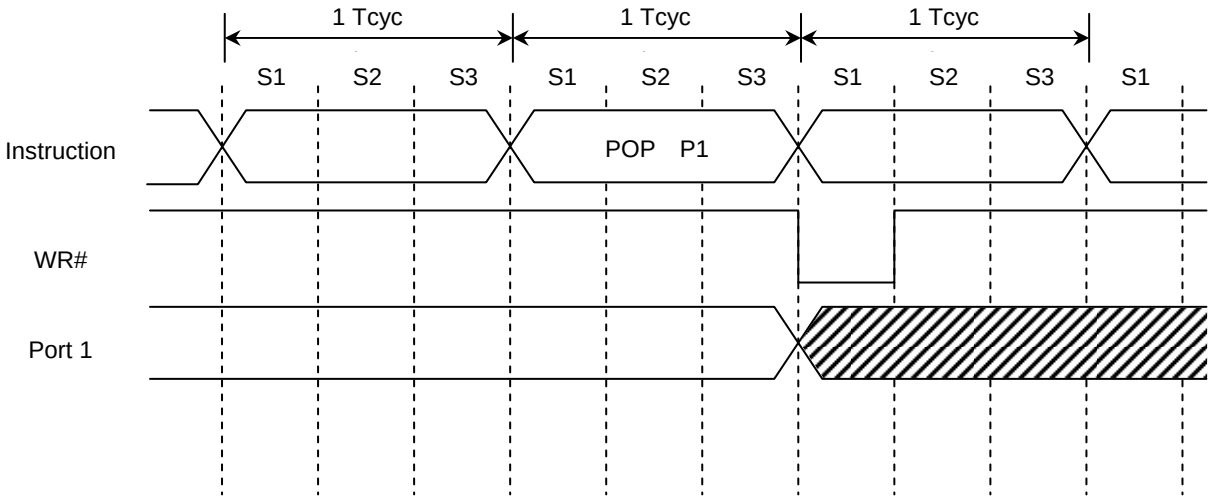
<Read mode timing chart>



<2> External memory write mode

- Execute an instruction (e.g., POP) for writing data into P1 to generate a write signal at P23 at the timing of S1.

<Write mode timing chart>



3.16 SMIIC0 (Single Master I²C)

3.16.1 Overview

The I²C bus module incorporated in this series of microcontrollers has the following two functions:

1) I²C communication in the single-master master mode (Note)
2) Synchronous 8-bit serial I/O (2- or 3-wire system, data MSB first)

Note: This module does not have an address comparator function. Consequently, it is necessary to perform address comparison and other processing under program control when using this module in the single-master slave mode or performing I²C communication in the multi-master mode.

3.16.2 Circuit Configuration

3.16.2.1 I²C control register 0 (SMIC0CNT) (8-bit register)

1) This register controls the I²C bus mode.
2) This register controls interrupts.

3.16.2.2 I²C status register 0 (SMIC0STA) (8-bit register)

1) This register is used to provide I²C bus event detection flags.
2) This register controls the ACK data.

3.16.2.3 I²C baudrate control register 0 (SMIC0BRG) (8-bit register)

1) This register is used to control the clock frequency of the noise filter in the SDA and SCL import blocks.
2) This register controls the frequency of the SCL clock.

3.16.2.4 I²C data buffer 0 (SMIC0BUF) (8-bit register)

1) The data is transmitted and received through this register.

3.16.2.5 I²C port control register 0 (SMIC0PCT) (4-bit register)

1) This register controls the I²C ports

3.16.2.6 I²C port select register 0 (SMIC0PSL) (8-bit register)

1) This register is used to select and control the I²C ports.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA0	0000 0000	R/W	SMIC0CNT	RUN	MST	TRX	SCL8	MKC	BB	END	IE
FEA1	0000 0000	R/W	SMIC0STA	SMD	RQL9	STD	SPD	AL	OVR	TAK	RAK
FEA2	0000 0000	R/W	SMIC0BRG	BRP		BRDQ	BRD				
FEA3	0000 0000	R/W	SMIC0BUF	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA4	HHHH 0000	R/W	SMIC0PCT	-	-	-	-	SHDS	PHV	PCLV	PSLW
FEA5	0000 0000	R/W	SMIC0PSL	BIT7	OPSDO0	OPSDA0	OPSCL0	SMI0OSL	SDO0EN	SDA0EN	SCL0EN

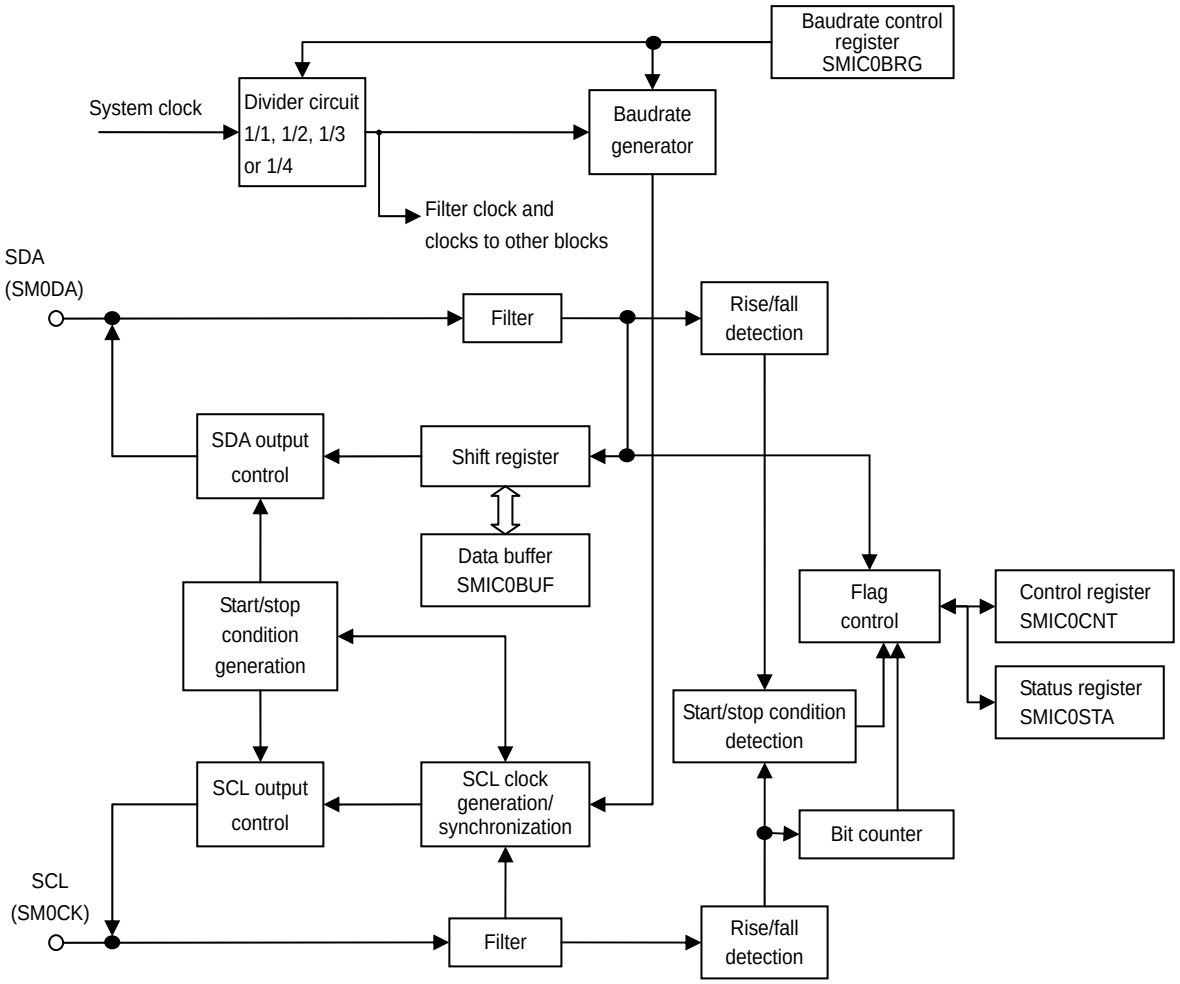


Figure 3.16.1 SMIIC0 Block Diagram

3.16.3 Related Registers

3.16.3.1 I²C control register 0 (SMIC0CNT)

1) This register is an 8-bit register that controls the operation of the SMIIC module.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA0	0000 0000	R/W	SMIC0CNT	RUN	MST	TRX	SCL8	MKC	BB	END	IE

RUN (bit 7): SMIIC0 operation enable

Setting this bit to 1 activates the SMIIC0 module.
Setting this bit to 0 stops the SMIIC0 module.

MST (bit 6): Master-slave control

- I²C mode (SMD = 0)

  When this bit is set to 1, the SMIIC0 module runs in master mode.

  (The module generates start and stop conditions and sends transfer clocks.)

  When this bit is set to 0, the SMIIC0 module runs in slave mode.

  (The module generates no clocks. It performs data transmission and reception in synchronization with a clock from the master.)

  Conditions under which MST is reset:

  <1> A stop condition is detected.

  <2> An arbitration lost is detected.

  After an arbitration lost is detected, this bit remains uncleared and the transmission of the clock is continued until the end of the transfer of one byte.

  After an arbitration lost, the MST flag is cleared when the interrupt request flag (END) is set.

- Synchronous 8-bit serial mode (SMD = 1)

  Setting this bit to 1 starts 8-bit communication.

  Conditions under which MST is reset:

  <1> MST is reset on the rising edge of the 8th clock.

TRX (bit 5): Transmitter/receiver control

- I²C mode (SMD = 0)

  When this bit is set to 1, the SMIIC0 module serves as a transmitter.

  When this bit is set to 0, the SMIIC0 module serves as a receiver.

  Conditions under which TRX is reset:

  <1> A stop condition is detected.

  <2> An arbitration lost is detected.

  <3> A start condition is detected in slave mode.

- Synchronous 8-bit serial mode (SMD = 1)

  Setting this bit to 1 places the module into data transfer mode.

  Setting this bit to 0 places the module into data reception mode.

SCL8 (bit 4): Interrupt control on the falling edge of the 8th clock

- I^2C mode (SMD = 0)

  When this bit is set to 1, an interrupt request is generated on the falling edge of the 8th clock.

  When this bit is set to 0, no interrupt request is generated on the falling edge of the 8th block.

  Conditions under which SCL8 is set:

  <1> A start condition is detected.

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

- Synchronous 8-bit serial mode (SMD = 1)

  This bit must always be set to 0.

MKC (bit 3): Start/stop condition generation control

- I^2C mode (SMD = 0)

  This bit is a write-only bit and is set to 1 to generate a start or stop condition. (This bit is always read as 0.)

- Synchronous 8-bit serial mode (SMD = 1)

  This bit must always be set to 0.

BB (bit 2): Bus busy flag (read-only)

- I^2C mode (SMD = 0)

  Bit 2 consists of a read-only BB and write-only BBW.

  The read-only BB indicates the busy status of the bus. It is set when a start condition is detected and reset when a stop condition is detected.

  A 1 in this bit indicates that the I^2C bus is busy.

  When generating a start condition, make sure that this bit is set to 0 and that both SDA and SCL are set to high (except when generating a restart condition).

  This bit is a read-only bit. It cannot be rewritten directly with an instruction.

Conditions under which BB is set:

<1> A start condition is detected.

Conditions under which BB is reset:

<1> A stop condition is detected.

<2> RUN is set to 0.

BBW (bit 2): Start/stop condition generation control

Bit 2 consists of a read-only BB and write-only BBW.

The write-only BBW is used to control the generation of start/stop conditions by writing its value together with bits 6, 5, and 3 of this register (SMIC0CNT: 0FEA0h) with a MOV instruction.

- If the interrupt request enable control bit IE is set to 1:

  Loading SMIC0CNT with EDh generates a start condition.

  Loading SMIC0CNT with E9h generates a stop condition.

- If the interrupt request enable control bit IE is set to 0:

  Loading SMIC0CNT with ECh generates a start condition.

  Loading SMIC0CNT with E8h generates a stop condition.

* See Subsection 3.16.5 “Start Condition and Stop Condition,” for details on the generation of start/stop conditions.

- Synchronous 8-bit serial mode (SMD = 1)

  This bit is a read-only bit and gives the same value as MST (bit 6) when read.

END (bit 1): Interrupt flag

- I^2C mode (SMD = 0)

  This bit is set at the end of data transfer or on a stop condition.

  If this bit is set to 1 and SCL is set to low, this module continuously sends low signals to SCL until this flag is cleared, whether it is in master or slave mode.

  Conditions under which END is set:

  <1> The falling edge of the 8th clock if SCL8 is set to 1

  <2> The falling edge of the ACK clock

  <3> A stop condition is detected

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

  When this bit is cleared, the module stops the continuous transmission of low signals to SCL and continues transfer operation. Data loading into or reading from the buffer SMIC0BUF must be completed before this bit is cleared.

- Synchronous 8-bit serial mode (SMD = 1)
  This bit is set at the end of data transfer.

  Conditions under which END is set:
  <1> The rising edge of the 8th clock

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

IE (bit 0): Interrupt request enable control

When this bit and END are set to 1, an interrupt request to vector address 002BH is generated.

3.16.3.2 I^2C status register 0 (SMIC0STA)

1) This register is an 8-bit register used to control the I^2C bus and detect each event.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA1	0000 0000	R/W	SMIC0STA	SMD	RQL9	STD	SPD	AL	OVR	TAK	RAK

SMD (bit 7): I^2C /synchronous 8-bit serial mode select

Setting this bit to 1 runs this module in the synchronous 8-bit serial mode.
When this bit is set to 1, the noise filter function for the clock input pin and data input pin is disabled.
Setting this bit to 0 runs this module in the I^2C communication mode.
When this bit is set to 0, the noise filter function for the clock input pin and data input pin is enabled.

RQL9 (bit 6): ACK clock timing detection flag (read-only)

This flag is set and held at 1 from the falling edge of the 9th clock until the falling edge of the next clock.
This bit is a read-only bit. It cannot be rewritten directly with an instruction.

* This bit is not used in synchronous 8-bit serial mode (SMD = 1). This bit is always read as 0.

STD (bit 5): Start condition detection flag

This flag bit is set when a start condition is detected.

Conditions under which STD is set:
<1> A start condition is detected.

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

* This bit is not automatically set in the synchronous 8-bit serial mode (SMD = 1). This bit must always be set to 0.

SPD (bit 4): Stop condition detection flag

This flag is set when a stop condition is detected.

Conditions under which SPD is set:

<1> A stop condition is detected.

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

* This bit is not automatically set in the synchronous 8-bit serial mode (SMD = 1). This bit must always be set to 0.

AL (bit 3): Arbitration lost detection flag

This flag is set when an arbitration lost is detected in master mode.

Conditions under which AL is set:

<1> On the rising edges of the 1st to 8th clocks in master transmitter mode and on the rising edge of the 9th clock in master receiver mode, when the state of the internal SDA is high and the level at the SDA pin is low.

<2> Generation of start conditions is disabled by the duplicate start condition prevention function.

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

* This bit is not automatically set in the synchronous 8-bit serial mode (SMD = 1). This bit must always be set to 0.

OVR (bit 2): Overrun detection flag

- I^2C mode (SMD = 0)

  This flag is set if the falling edge of the clock signal on the SCL line is detected when BB (0FEA0h, bit 2) is set to 0.

  Conditions under which OVR is set:

  <1> A falling edge of SCL is detected when BB is set to 0.

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

- Synchronous 8-bit serial mode (SMD = 1)

  This flag is set if the falling edge of the clock signal on the SCL line is detected when MST (0FEA0h, bit 6) is set to 0.

  Conditions under which OVR is set:

  <1> A falling edge of SCL is detected when MST is set to 0.

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

TAK (bit 1): ACK clock timing SDA control bit

The value of this bit is placed in SDA at the ACK clock timing in master receiver/slave receiver mode.

In master transmitter/slave transmitter mode, SDA is set to the high level at the ACK clock timing regardless of the value of this bit.

Conditions under which TAK is set:

<1> A stop condition is detected.

<2> An arbitration lost is detected.

<3> A start condition is detected in slave mode.

* This bit must always be set to 0 in the synchronous 8-bit serial mode (SMD = 1).

RAK (bit 0): Receive acknowledge data storage bit (read-only)

This bit stores the ACK receive data.

This bit is loaded with the SDA data that is established at the rise timing of the ACK clock in both transmitter and receiver modes.

Conditions under which RAK is set:

<1> SDA is set to the high level on the rising edge of an ACK clock.

Conditions under which RAK is reset:

<1> SDA is set to the low level on the rising edge of an ACK clock.

This bit is a read-only bit. It cannot be rewritten directly with an instruction.

* This bit is not used in synchronous 8-bit serial mode (SMD = 1). This bit is always read as 0.

3.16.3.3 I²C baudrate control register 0 (SMIC0BRG)

1) This register is an 8-bit register that controls the frequency of the SDA and SCL filter clocks and the frequency of the SCL clock.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA2	0000 0000	R/W	SMIC0BRG	BRP		BRDQ	BRD				

BRP (bits 7, 6): Filter clock control

BRP	Filter Clock Period (Tfilt)
00	1/3 × Tcyc × 1
01	1/3 × Tcyc × 2
10	1/3 × Tcyc × 3
11	1/3 × Tcyc × 4

* Tcyc denotes the instruction cycle time.

BRP must be set so that the filter clock period Tfilt falls within the following value range:

250nsec ≧ Tfilt ＞ 140nsec

System Clock Frequencies and BRP Values

System Clock	Instruction Cycle Time	BRP	Tfilt
6 MHz	500ns	00	500ns × 1/3 = 166ns
8 MHz	375ns	01	375ns × 2/3 = 250ns
12 MHz	250ns	01	250ns × 2/3 = 166ns

BRDQ (bit 5): SCL clock frequency control

This bit must be set to 1 in Standard-mode and to 0 in Fast-mode.

BRD (bits 4 to 0): SCL clock frequency control

Assuming that the 5 bits of BRD are set to n, the SCL clock period Tfsck is calculated as follows:

When BRDQ = 0 (Fast-mode)

$Tfsck = Tfilt \times (n + 1) \times 2$

When BRDQ = 1 (Standard-mode)

$Tfsck = Tfilt \times (n + 1) \times 8$

The SCL clock frequency fsck is calculated as follows:

When BRDQ = 0 (Fast-mode)

$fsck = 1/ (Tfilt \times (n + 1) \times 2)$

When BRDQ = 1 (Standard-mode)

$fsck = 1/ (Tfilt \times (n + 1) \times 8)$

* Tfilt denotes the filter clock period that is determined by the system clock frequency and filter clock control bits BRP (SMIC0BRG, bits 7 and 6).

* When used in I^2C communication mode (SMD=0), the BRD 5 bits value n must be set to 4 or greater (setting it to a value of 0 to 3 is inhibited).

* When used in synchronous 8-bit serial mode (SMD=1), this register must be set as follows:

  BRP (SMIC0BRG, bits 7 and 6) = 00

  BRDQ = 0 or 1

  The BRD 5 bits value n must be 1 or greater (setting it to a value of 0 is inhibited).

  In this case, the frequency of the output clock fsck can be determined by the following formula:

  When BRDQ = 0: $fsck = 3 / (Tcyc \times (n + 1) \times 2)$

  When BRDQ = 1: $fsck = 3 / (Tcyc \times (n + 1) \times 8)$

SMIIC0

Standard-mode: BRDQ = 1

SCL Frequency (kHz)

BRD Value n	Tfilt Period	
	250ns (4MHz)	166ns (6MHz)
00h	Inhibited	Inhibited
01h	Inhibited	Inhibited
02h	Inhibited	Inhibited
03h	Inhibited	Inhibited
04h	100	*
05h	83.3	*
06h	71.4	*
07h	62.5	94.1
08h	55.6	83.7
09h	50	75.3
0Ah	45.5	68.5
0Bh	41.7	57.9
0Ch	38.5	53.8
0Dh	35.7	50.2
0Eh	33.3	47.1
0Fh	31.3	44.3
10h	29.4	41.8
11h	27.8	39.6
:	:	:
1Ch	17.2	25.9
1Dh	16.7	25.1
1Eh	16.1	24.3
1Fh	15.6	23.5

Fast-mode: BRDQ = 0

SCL Frequency (kHz)

BRD Value n	Tfilt Period	
	250ns (4MHz)	166ns (6MHz)
00h	Inhibited	Inhibited
01h	Inhibited	Inhibited
02h	Inhibited	Inhibited
03h	Inhibited	Inhibited
04h	400	*
05h	333.3	*
06h	328.7	*
07h	250	376.5
08h	222.2	334.7
09h	200	301.2
0Ah	181.8	273.8
0Bh	166.7	251
0Ch	153.8	231.7
0Dh	142.9	215.1
0Eh	133.3	200.8
0Fh	125	188.3
10h	117.6	177.2
11h	111.1	167.3
:	:	:
1Ch	69	103.9
1Dh	66.7	100.4
1Eh	64.5	97.23
1Fh	62.5	94.1

* Out of I^2C bus specifications

3.16.3.4 I²C data buffer 0 (SMIC0BUF)

1) This buffer is an 8-bit register used to store the receive data or write the transmit data.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA3	0000 0000	R/W	SMIC0BUF	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

- Data reception
  - I²C mode (SMD = 0)

    The data from the receive shift register is transferred to the SMIC0BUF register on the falling edge of the 8th SCL clock in both transmitter and receiver modes.

  - Synchronous 8-bit serial mode (SMD = 1)

    The data from the receive shift register is transferred to the SMIC0BUF register on the rising edge of the 8th SCL clock in both transmitter and receiver modes.

- Data transmission
  - I²C mode (SMD = 0)

    In the transmitter mode, the contents of the SMIC0BUF register are transferred to the transmit shift register at one of the following timings:

    <1> A start condition is detected

    <2> Data is written into SMIC0BUF when END is set to 1.

  - Synchronous 8-bit serial mode (SMD = 1)

    In the data transmission mode, the contents of the SMIC0BUF register are transferred to the transmit shift register at the following timing:

    <1> Data is written into SMIC0BUF when MST is set to 0.

3.16.3.5 I²C port control register 0 (SMIC0PCT)

1) This register is a 4-bit register used to control the I²C ports.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA4	HHHH 0000	R/W	SMIC0PCT	-	-	-	-	SHDS	PHV	PCLV	PSLW

SHDS (bit 3): SDA internal HOLD time adjustment

This bit must be set to 0.

PHV (bit 2):

This bit must be set to 0.

PCLV (bit 1):

This bit must be set to 0.

PSLW (bit 0):

This bit must be set to 0.

3.16.3.6 I²C port select register 0 (SMIC0PSL)

1) This register is an 8-bit register that is used to select and control the I²C ports.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA5	0000 0000	R/W	SMIC0PSL	BIT7	OPSDO0	OPSDA0	OPSCL0	SMI0OSL	SDO0EN	SDA0EN	SCL0EN

BIT7 (bit 7):

This bit must be set to 0.

OPSDO0 (bit 6): P15 (serial data) output type select

A 1 in this bit sets the output type of this pin to N-channel open drain.

A 0 in this bit sets the output type of this pin to CMOS.

OPSDA0 (bit 5): P16 or P14 (SDA/serial data) output type select

A 1 in this bit sets the output type of this pin to N-channel open drain.

A 0 in this bit sets the output type of this pin to CMOS.

OPSCL0 (bit 4): P17 or P13 (SCL/serial clock) output type select

A 1 in this bit sets the output type of this pin to N-channel open drain.

A 0 in this bit sets the output type of this pin to CMOS.

SMI0OSL (bit 3): I²C/synchronous 8-bit serial port select

This pin selects the ports to be used for I²C/synchronous 8-bit serial communication.

Port Function		SMI0OSL	
I²C Mode	Synchronous 8-bit Serial Mode	0	1
SCL input/output	Serial clock input/output	P17	P13
SDA input/output	Serial data input/output	P16	P14
–	Serial data output	P15	P15

SDO0EN (bit 2): P15 (serial data) output select

When P15 is set up for output (P15DDR=1) and SDO0EN is set to 1, an OR of the serial data and port data latch data is output at pin P15.

SDA0EN (bit 1): P16 or P14 (SDA/serial data) otuput select

When P16 or P14 is set up for output (P16DDR=1 or P14DDR=1) and SDA0EN is set to 1, an OR of the SDA/serial data and port data latch data is output at pin P16 or P14.

SCL0EN (bit 0): P17 or P13 (SCL/serial clock) otuput select

When P17 or P13 is set up for output (P17DDR=1 or P13DDR=1) and SCL0EN is set to 1, an OR of the SCL/serial clock and port data latch data is output at pin P17 or P13.

3.16.3.7 SMIIC port settings

1) SCL/serial clock I/O port setting (P17 or P13)

Set P17FCR or P13FCR to 0 when using P17 or P13 as an SMIIC port.

Register Data					Port P17 State
P17DDR	P17	SMI0OSL	SCL0EN	OPSCL0	Output
0	0	–	–	–	Open
0	1	–	–	–	Internal pull-up resistor on
1	0	0	0	–	Low output
1	0	0	1	0	Serial clock output (CMOS)
1	0	0	1	1	SCL/serial clock output (N-channel open drain)
1	0	1	–	–	Low output
1	1	–	–	–	High output (open if N-channel open drain output is selected as a user option)

Register Data					Port P13 State
P13DDR	P13	SMI0OSL	SCL0EN	OPSCL0	Output
0	0	–	–	–	Open
0	1	–	–	–	Internal pull-up resistor on
1	0	1	0	–	Low output
1	0	1	1	0	Serial clock output (CMOS)
1	0	1	1	1	SCL/serial clock output (N-channel open drain)
1	0	0	–	–	Low output
1	1	–	–	–	High output (open if N-channel open drain output is selected as a user option)

2) SDA/serial data I/O port (P16 or P14) setting

Set P16FCR or P14FCR to 0 when using P16 or P14 as an SMIIC port.

Register Data					Port P16 State
P16DDR	P16	SMI0OSL	SDA0EN	OPSDA0	Output
0	0	–	–	–	Open
0	1	–	–	–	Internal pull-up resistor on
1	0	0	0	–	Low output
1	0	0	1	0	Serial data output (CMOS)
1	0	0	1	1	SDA/serial data output (N-channel open drain)
1	0	1	–	–	Low output
1	1	–	–	–	High output (open if N-channel open drain output is selected as a user option)

Register Data					Port P14 State
P14DDR	P14	SMI0OSL	SDA0EN	OPSDA0	Output
0	0	–	–	–	Open
0	1	–	–	–	Internal pull-up resistor on
1	0	1	0	–	Low output
1	0	1	1	0	Serial data output (CMOS)
1	0	1	1	1	SDA/serial data output (N-channel open drain)
1	0	0	–	–	Low output
1	1	–	–	–	High output (open if N-channel open drain output is selected as a user option)

3) Serial data output port (P15) setting (used in 3-wire synchronous 8-bit serial mode)

Set P15FCR to 0 when using P15 as an SMIIC port.

Register Data					Port P15 State
P15DDR	P15	SMI0OSL	SDO0EN	OPSDO0	Output
0	0	–	–	–	Open
0	1	–	–	–	Internal pull-up resistor on
1	0	–	0	–	Low output
1	0	–	1	0	Serial data output (CMOS)
1	0	–	1	1	Serial data output (N-channel open drain)
1	1	–	–	–	High output (open if N-channel open drain output is selected as a user option)

* When using this module in I^2C mode, set SCL0EN, SDA0EN, OPSCL0, and OPSDA0 of the I^2C port select register 0 (SMIC0PSL) to 1.
* Set the output type of the clock I/O port to open when using an external clock in the synchronous 8-bit serial mode. When receiving data in the synchronous 8-bit serial mode, set the output type of the data I/O port to open.

3.16.4 Waveform of Generated Clocks and SCL Rise Times

3.16.4.1 Waveform of generated clocks

The SCL clock output waveform has a duty cycle of 50% of the clock period Tfsck that is defined by the I^2C baudrate control register 0 (SMIC0BRG).

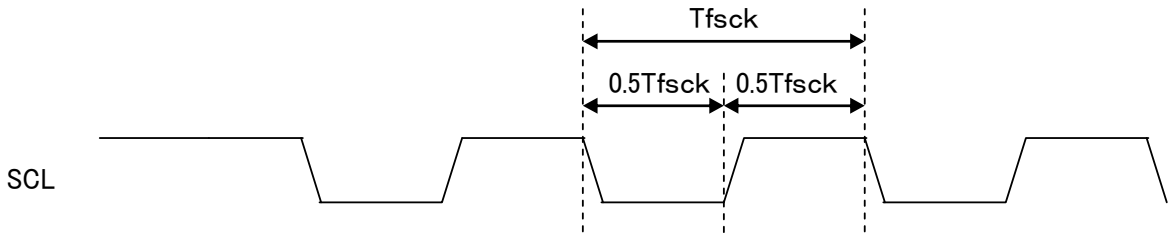


Figure 3.16.2 SCL Clock Waveform

If the clock frequency is set to 400 kHz for processing in Fast-mode, the low period of the SCL is 1.25 μs (provided that the rise and fall times of the signal are ignored), which does not meet the I^2C bus interface specification (1.3 μs minimum).

To cope with this issue, consider the following countermeasures:

1) Reduce the transfer rate so as to meet the specification.
2) Adjust the rise and fall times by adjusting the external components such as the resistance of the pull-up resistor.

3.16.4.2 SCL rise time

This module always monitors the rise timing of the SCL clock line and attempts to establish synchronization to guarantee the predetermined high-level width of the clock output even if the SCL line is set to low by another master or slave in I^2C mode.

The SCL rise time is defined by the I^2C bus interface specifications as being within 300 ns in Fast-mode and within 1000 ns in Standard-mode.

No problem occurs in Fast-mode because the maximum SCL rise time is 300 ns. If the rise time is longer than (Tfilt × 2.5) in Standard-mode, however, the module's synchronization function is activated, making the transfer rate lower than the preset clock frequency.

System Clock	BRP	Tfilt	Tfilt x 2.5
6 MHz	00	166 ns	415 ns
8 MHz	01	250 ns	625 ns
12 MHz	01	166 ns	415 ns

To run the module at the preset transfer rate, set the resistance of the pull-up resistor and the load capacitance so that the rise time of the SCL line is shorter than the Tfilt × 2.5 value that is shown above.

3.16.5 Start Condition and Stop Condition

3.16.5.1 Definition of start and stop conditions

SDA must be in a stable state while SCL is high. That is, it is only when SCL is low that the state of SDA can switch between high and low. By making use of this fact, the I^2C protocol defines special conditions for signals indicating start and stop of data transfer as follows:

- Start condition (S)

  Data transfer start condition. The state of SDA changes from high to low when SCL is set to high.
- Stop condition (P)

  Data transfer stop condition. The state of SDA changes from low to high when SCL is set to high.

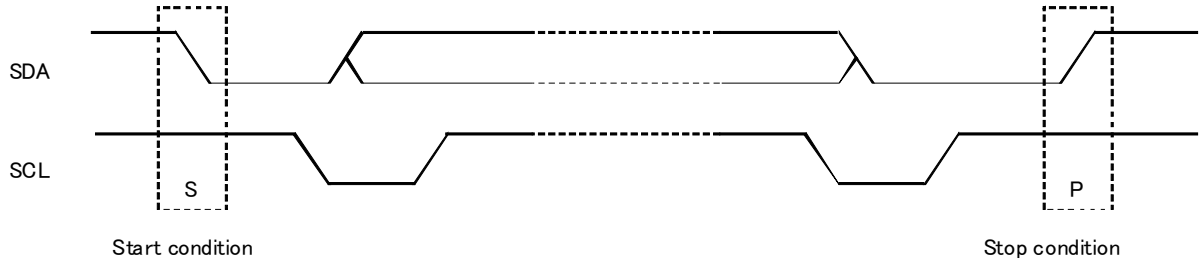


Figure 3.16.3 Start and Stop Conditions

3.16.5.2 Generating a start condition

The process of generating a start condition is initiated by loading the I^2C control register SMIC0CNT with the value given below when SMIIC0 operation enable bit RUN (SMIC0CNT, bit 7) is preset to 1.

Since bit 0 of the SMIC0CNT register is an interrupt request enable control bit, data loaded into the register varies depending on whether interrupts are to be enabled (IE = 1) or disabled (IE = 0).

Methods of generating a start condition:

Loading SMIC0CNT with EDh (when enabling interrupts)

Loading SMIC0CNT with ECh (when disabling interrupts)

3.16.5.3 Start condition generation timing

Before generating a start condition, make sure that the BB flag (SMIC0CNT, bit 2) is set to 0.

Follow the procedure below when starting this module after a reset.

<1> Set up the ports (see Subsection 3.16.3.7, "SMIIC port settings").

<2> Set the filter clock and baudrate clock using SMIC0BRG.

<3> Set RUN (SMIC0CNT, bit 7) to 1.

<4> Insert wait equivalent to several baudrate clock cycles and make sure that both BB (SMIC0CNT, bit 2) and OVR (SMIC0STA, bit 2) are set to 0.

<5> To determine whether SDA and SCL lines are fixed by another master or slave device, read the SDA and SCL ports and make sure that they are set to high.

<6> If the result of the check in steps <4> and <5> is OK, it indicates that the start condition instructions can be safely executed.

<7> If the result of the check in steps <4> and <5> is NG, determine that the use of the bus is started by another master before this module starts operation, and wait until a stop condition is received. (It is necessary to perform wait time timeout processing using a timer in a situation in which the bus is locked under an abnormal condition)

<8> In a single master configuration or if the wait processing for a stop condition performed in step <7> times out, it is necessary to generate a stop condition by manipulating relevant ports under program control, considering that the bus is locked by another slave device.

Step 1. Set SCL0EN to 0 and set SCL low. In this case, if SDA is low, keep supplying clocks to SCL by setting SCL0EN to 0 and 1 alternately until SDA becomes high while SCL is low.

Step 2. Change the state of the SDA and SCL lines as follows:

1— SDA = H SCL = L (SDA0EN=1, SCL0EN=0)

2— SDA = L SCL = L (SDA0EN=0, SCL0EN=0)

3— SDA = L SCL = H (SDA0EN=0, SCL0EN=1)

4— SDA = H SCL = H (SDA0EN=1, SCL0EN=1)

(When the ports are manipulated as indicated above, it is necessary to take the setup/hold times for the other devices into consideration.)

The figure below shows a timing example for generating a start condition.

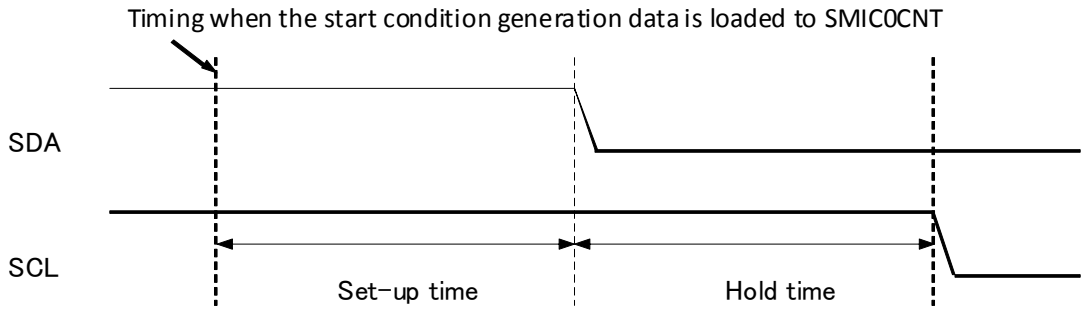


Figure 3.16.4 Start Condition Generation Timing Diagram

3.16.5.4 Restart condition generation timing

Follow the procedure below to generate a restart condition which is required to switch the transmission/reception mode or the destination slave device without generating a stop condition after transmitting a start condition and transmitting/receiving data in master communication mode.

<1> If the module is in the master receiver mode, send ACK data = 1 (NACK) to force the slave to release the SDA line.

<2> After the falling edge of the ACK data clock, make sure that END (SMIC0CNT, bit 1) is set to 1 and RQL9 (SMIC0STA, bit 6) is set to 1. While END = 1, the low level is kept output to the SCL line.

<3> Load SMIC0BUF with 7 bits of slave address data and the $R/\overline{W}$ bit.

<4> Load SMIC0CNT with the data for generating a start condition.

<5> Loading SMIC0CNT with the data for generating a start condition causes END (SMIC0CNT, bit 1) to be cleared and, after the elapse of the pre-set-up time for the restart condition, causes the SCL line to be released. Since the END flag is cleared by the start condition instruction, if interrupt processing is being executed as controlled by IE (SMIC0CNT, bit 0) set to 1, it is necessary to execute this start condition instruction immediately before exiting that interrupt processing.

The figure below shows a timing example for generating a restart condition.

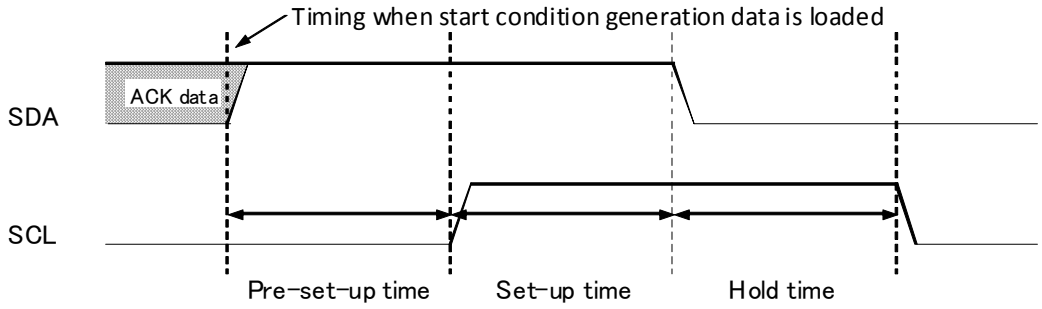


Figure 3.16.5 Restart Condition Generation Timing

3.16.5.5 Generating a stop condition

The process for generating a stop condition begins when END (SMIC0CNT, bit 1) is set to 1 on the falling edge of the ACK clock and the I^2C control register SMIC0CNT is loaded with the data given below while SCL is held low.

Since the bit 0 of SMIC0CNT is an interrupt request enable control bit, the data to be loaded into the SMIC0CNT register varies depending on whether interrupts are to be enabled (IE = 1) or disabled (IE = 0).

Methods of generating a stop condition:

Loading SMIC0CNT with E9h (when enabling interrupts)

Loading SMIC0CNT with E8h (when disabling interrupts)

3.16.5.6 Stop condition generation timing

Follow the procedure below when generating a stop condition in master communication mode.

<1> When the module is in master receiver mode, send ACK data = 1 (NACK) to force the slave to release the SDA line.

<2> After the falling edge of the ACK data clock, make sure that END (SMIC0CNT, bit 1) is set to 1 and RQL9 (SMIC0STA, bit 6) is set to 1. While END = 1, the low level is kept output to the SCL line.

<3> Load SMIC0BUF with 0FFh.

<4> Load SMIC0CNT with the data for generating a stop condition.

<5> Loading SMIC0CNT with the data for generating a stop condition causes END (SMIC0CNT, bit 1) to be cleared and, after the elapse of the pre-set-up time for a stop condition, causes the SCL line to be released. Since the END flag is cleared by the stop condition instruction, if interrupt processing is being executed as controlled by IE (SMIC0CNT, bit 0) set to 1, it is necessary to execute this stop condition instruction immediately before exiting that interrupt processing.

The figure below shows a timing example for generating a stop condition.

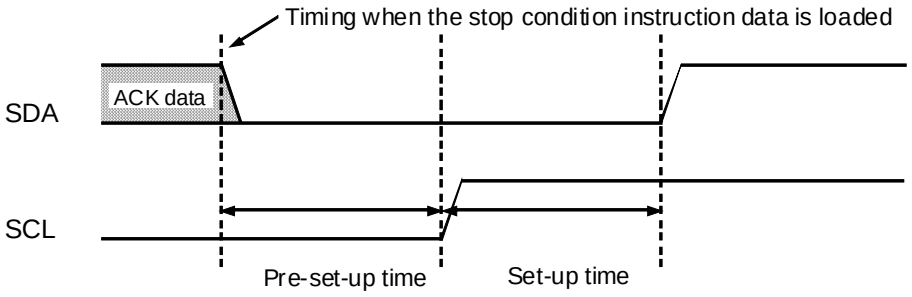


Figure 3.16.6 Stop Condition Generation Timing

3.16.6 Arbitration Lost

3.16.6.1 Arbitration

Arbitration refers to the process of enabling communication or the procedure for enabling a single master to control a bus. Arbitration is implemented by ANDing the SDA lines to the devices (the SDA line being set to low under the influence of a device that generates a low output). In this case, a master whose output does not match the SDA value is disabled for communication. Such a master needs to keep its output high so that it does not affect the SDA line. This state of a master that becomes disabled for communication is called an arbitration lost. Arbitration lost is detected when generating a start condition and when sending data in the master mode.

3.16.6.2 Arbitration lost during data transfer

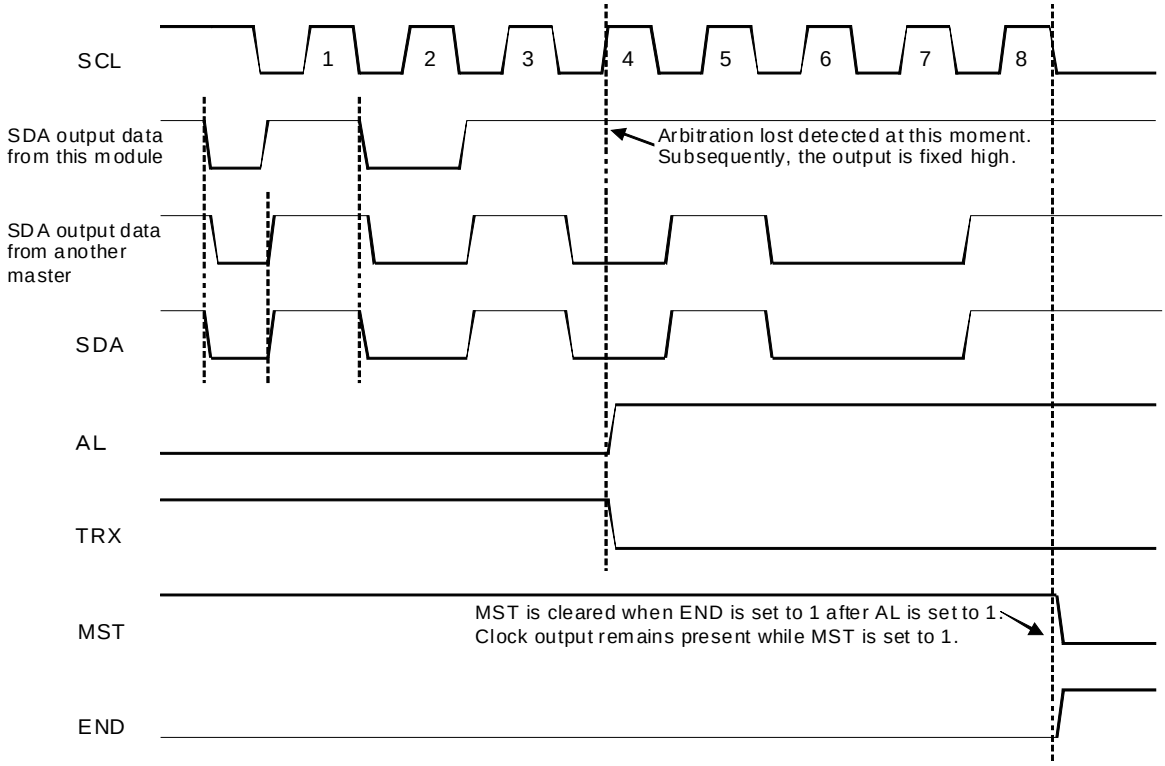


Figure 3.16.7 Arbitration Lost During Data Transfer

An arbitration lost during data transfer is identified by the SDA value that is established on the rising edge of SCL.

In Figure 3.16.7, since the output value of the internal SDA is high and the SDA value is low on the rising edge of the 4th clock, an arbitration lost is detected at this point and AL is set to 1.

Following the detection of an arbitration lost, AL is set, TRX is reset, and the SDA output is fixed at high. MST is not reset at this point and the transmission of SCL clocks is continued.

MST is cleared at the timing when END is set. When SCL8 (SMIC0CNT, bit 4) is set to 1, MST is cleared on the falling edge of the 8th clock, and on the falling edge of the 9th clock if SCL8 is set to 0, after which the transmission of clocks is stopped.

The detection of an arbitration lost is attempted in the data block (1st to 8th clocks) in the master transmitter mode and in the ACK block (9th clock) in the master receiver mode.

A master that has detected an arbitration lost needs to continue its operation as a slave until a stop condition is detected.

3.16.6.3 Arbitration lost while a start condition is being transmitted

It is under one of the following two cases in which an arbitration lost is detected during the period from the execution of a start condition instruction until a start condition is generated:

<1> The overrun detection flag OVR (SMIC0STA, bit 2) or the start condition detection flag STD (SMIC0STA, bit 5) is set to 1 when the start condition instruction is being executed.

<2> A change in the state of SDA from high to low is detected earlier than expected during the generation of the start condition due to the influences exerted by another master.

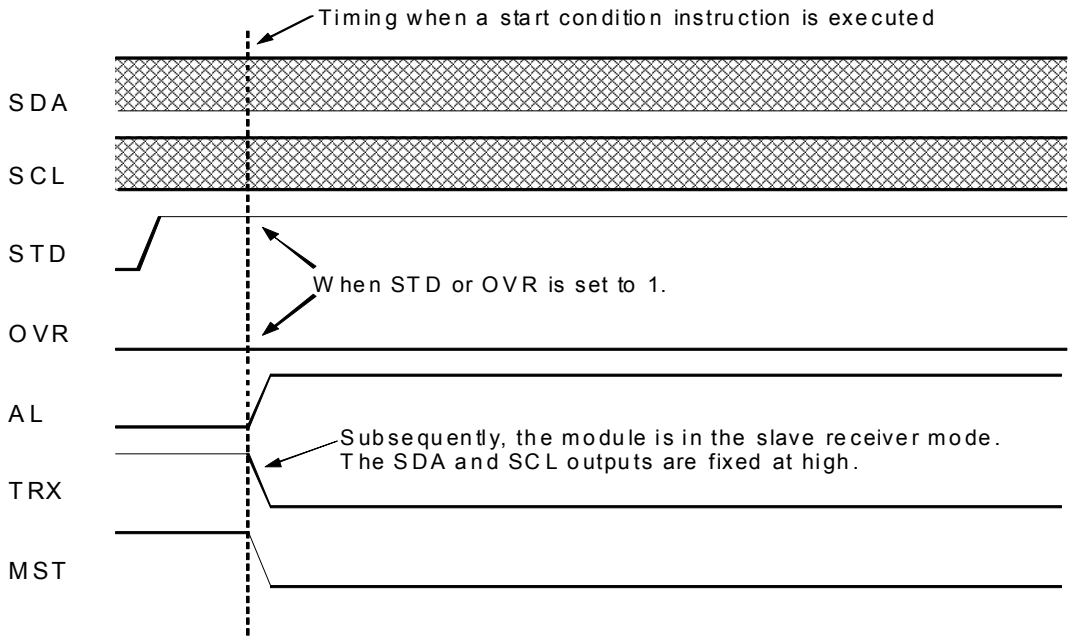


Figure 3.16.8 Arbitration Lost During Start Condition Generation <1>

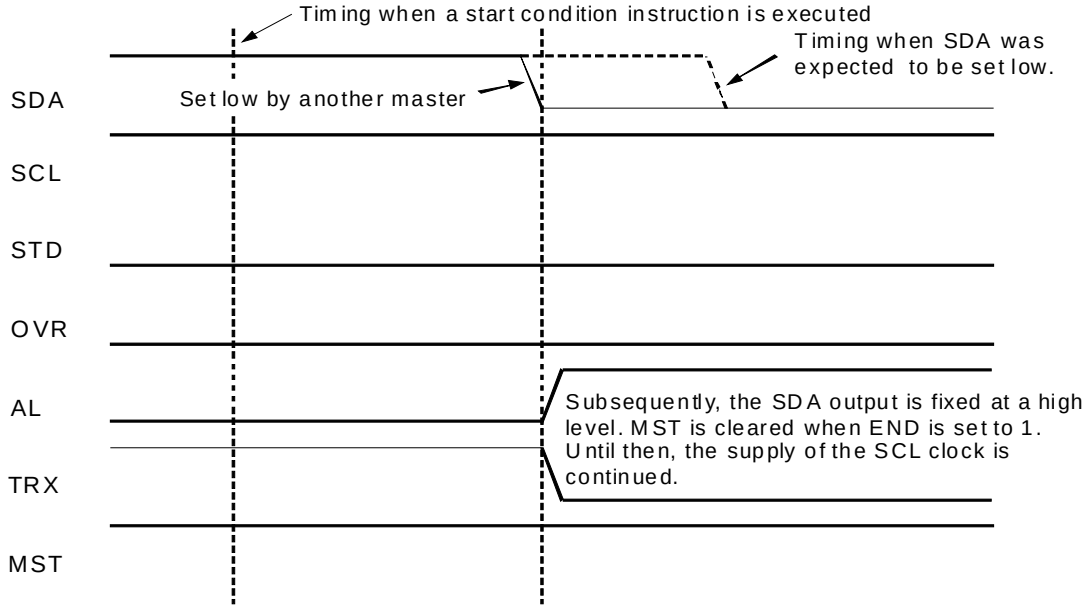


Figure 3.16.9 Arbitration Lost During Start Condition Generation <2>

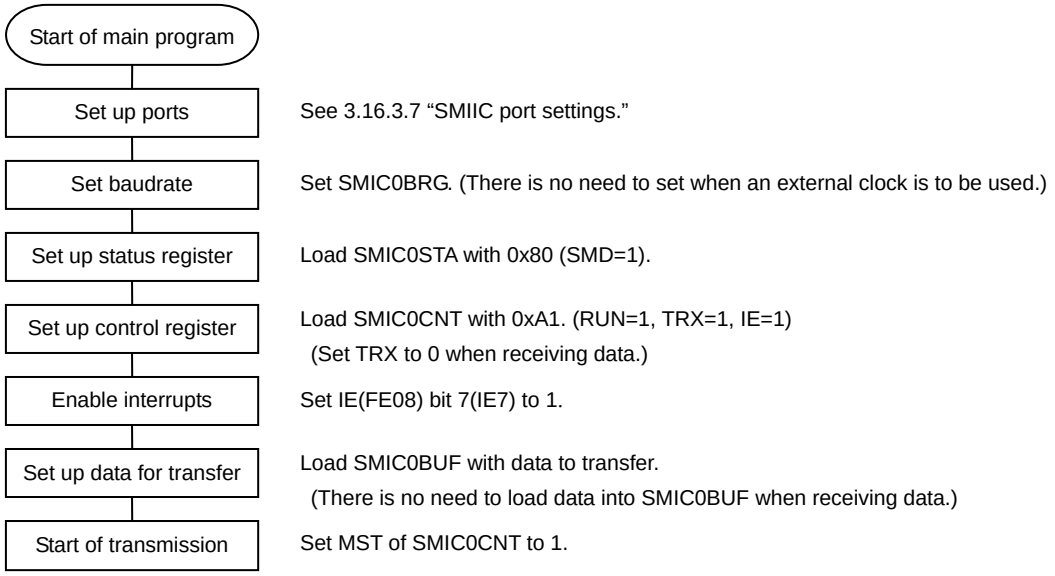
If an arbitration lost is detected under the condition <1> above, both MST and TRX are cleared at the timing when AL is set to 1, which causes the module to enter the slave receiver mode and to receive the incoming address.

If an arbitration lost is detected under the condition <2> above, TRX is cleared at the timing when AL is set to 1 but MST is not cleared. As in the case of arbitration lost during data transfer discussed in 3.16.6.2, the transmission of clocks is continued and MST is cleared at the timing when END is set. At this moment, the module enters the slave receiver mode and processes the received address under program control.

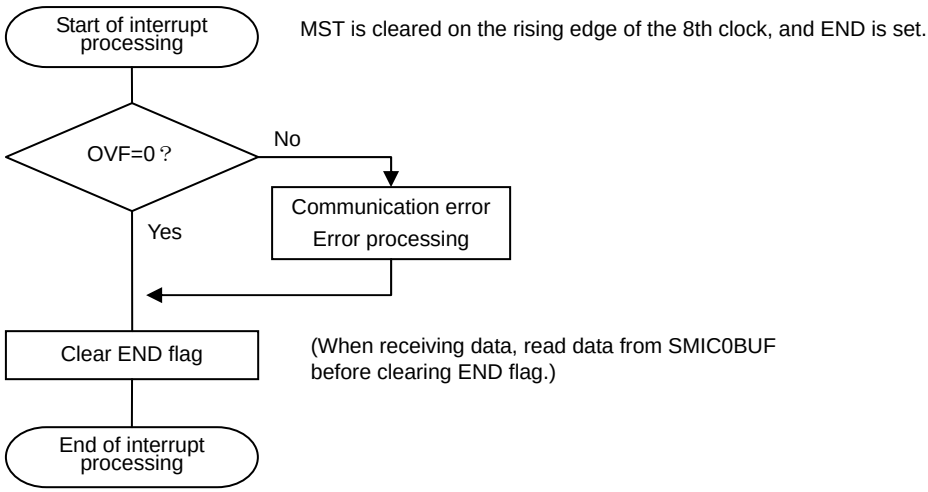
3.16.7 Examples of Simple SIO Mode Communication

3.16.7.1 Example of transmitting and receiving 1 byte in simple SIO mode

1. Main Program



2. Interrupt processing



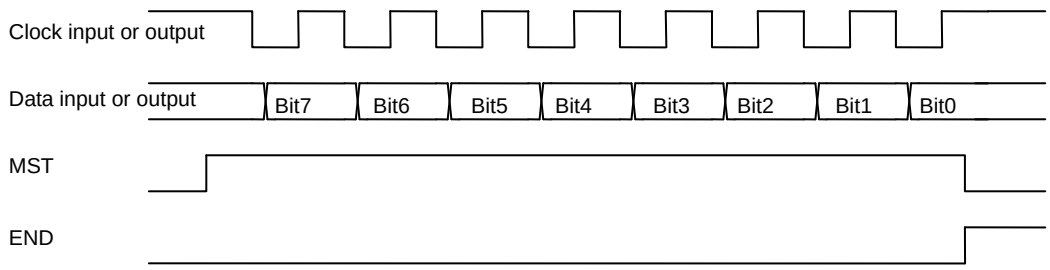


Figure 3.16.10 Waveforms of Simple SIO Mode 1-byte Transmission/Reception

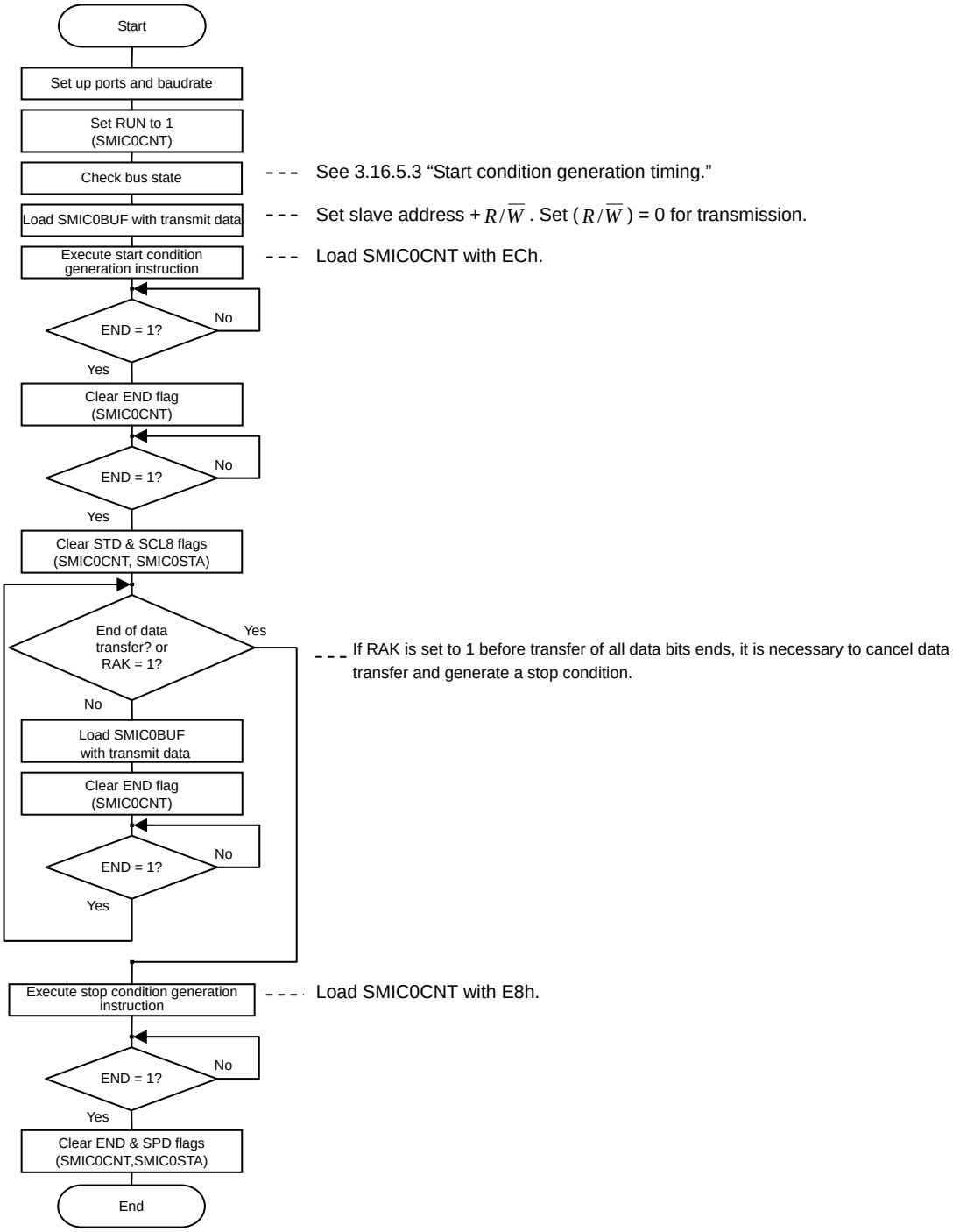
3.16.8 Examples of Single Master I²C Communication

The I²C communication flowcharts of each mode are given below.

* If abnormal conditions are expected to occur due to noise interferences or malfunctioning of the devices connected to the bus, it is necessary to provide measures to avoid lock conditions by implementing timeout processing using a timer, etc.

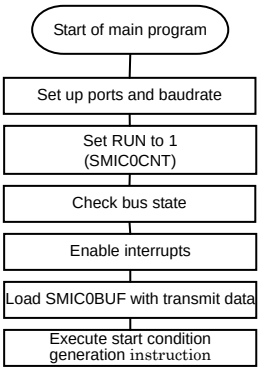
3.16.8.1 Example of transmitting data in single master mode (using no interrupt)

Below is the flowchart for sending data without using an interrupt.



3.16.8.2 Example of transmitting data in single master mode (using interrupts)

Below is the flowchart for sending data using interrupts.

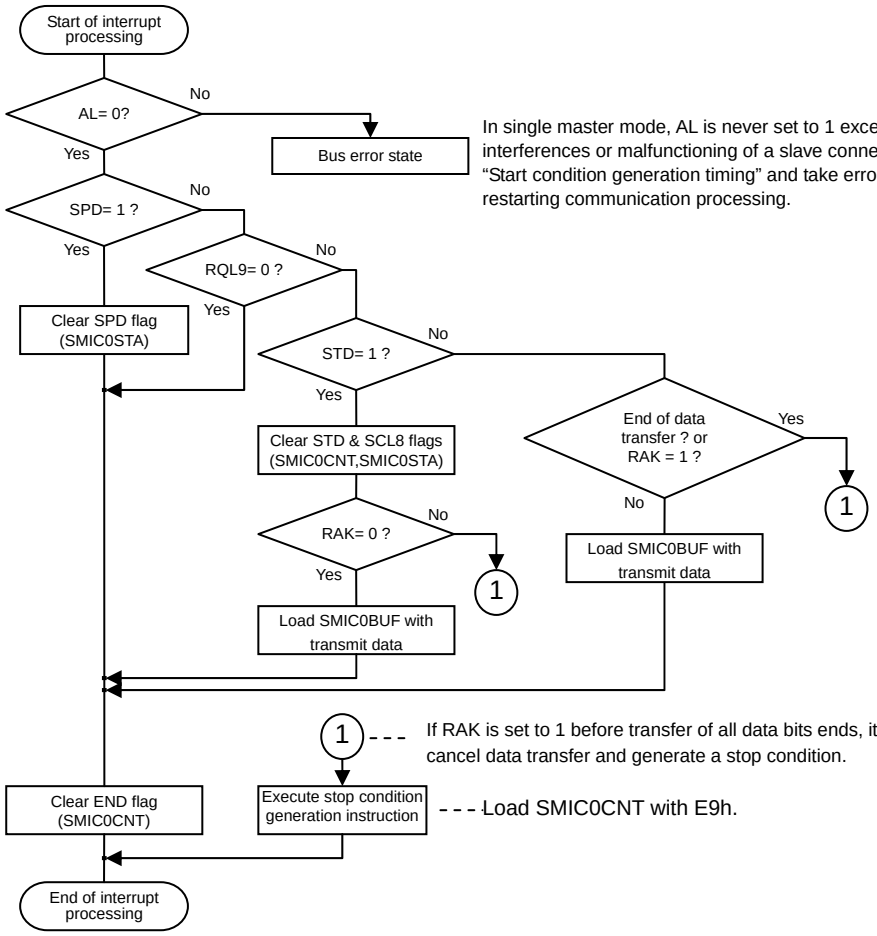


- - - See 3.16.5.3 “Start condition generation timing.”

- - - Set IE(FE08) bit 7(IE7) to 1.

- - - Set slave address + $R/\overline{W}$. Set ($R/\overline{W}$)=0 for transmission.

- - - Load SMIC0CNT with EDh.



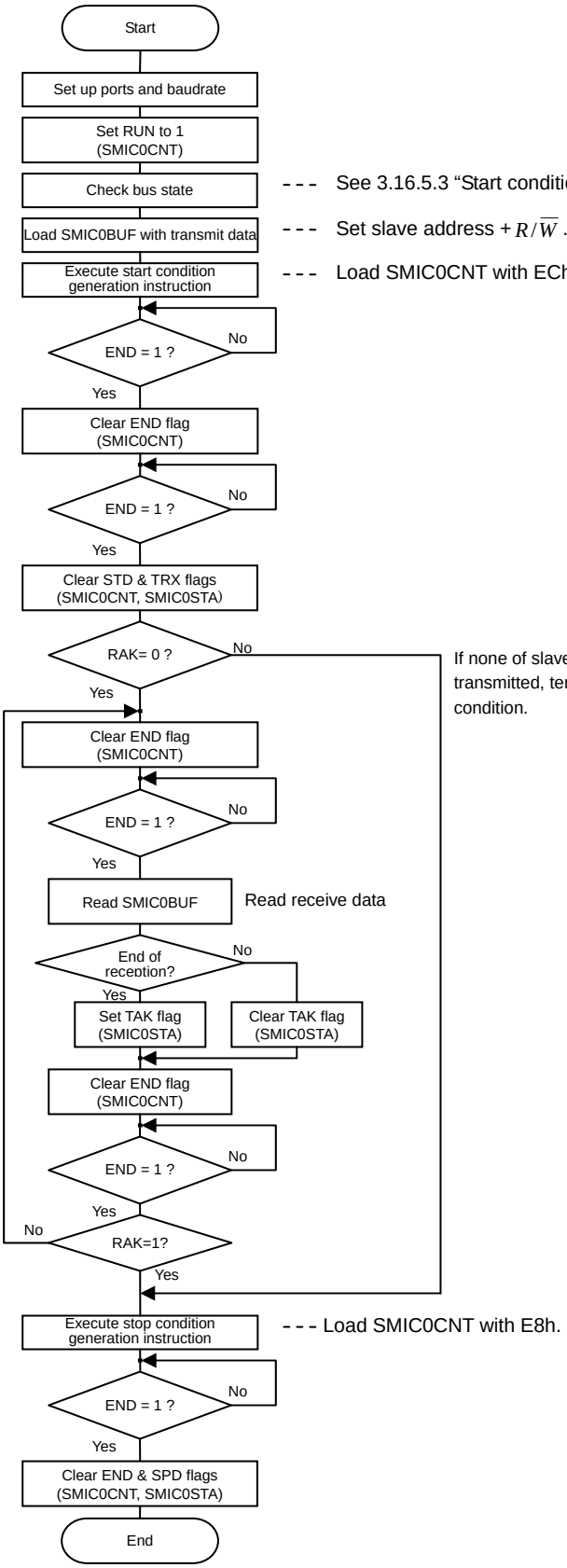
In single master mode, AL is never set to 1 except by external noise interferences or malfunctioning of a slave connected to the bus. See 3.16.5.3 “Start condition generation timing” and take error recovery measures before restarting communication processing.

If RAK is set to 1 before transfer of all data bits ends, it is necessary to cancel data transfer and generate a stop condition.

- - - Load SMIC0CNT with E9h.

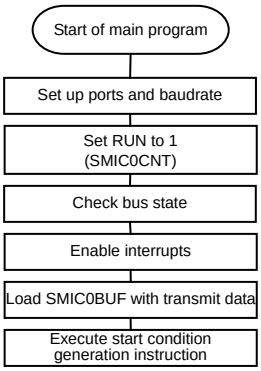
3.16.8.3 Example of receiving data in single master mode (using no interrupt)

Below is the flowchart for receiving data without using an interrupt.



3.16.8.4 Example of receiving data in single master mode (using interrupts)

Below is the flowchart for receiving data using interrupts.

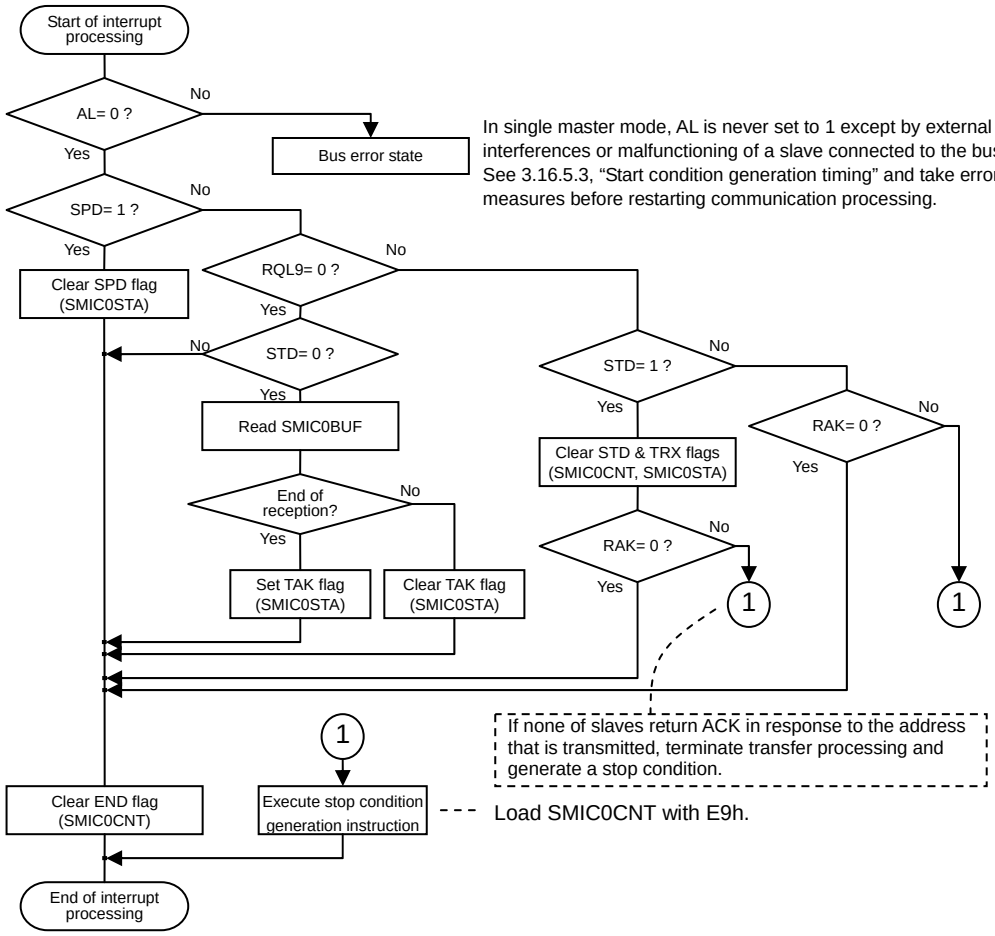


- - - See 3.16.5.3 “Start condition generation timing.”

- - - Set IE(FE08) bit 7(IE7) to 1.

- - - Set slave address + $R/\overline{W}$. Set ($R/\overline{W}$)=1 for reception.

- - - Load SMIC0CNT with EDh.



In single master mode, AL is never set to 1 except by external noise interferences or malfunctioning of a slave connected to the bus. See 3.16.5.3, “Start condition generation timing” and take error recovery measures before restarting communication processing.

If none of slaves return ACK in response to the address that is transmitted, terminate transfer processing and generate a stop condition.

- - - Load SMIC0CNT with E9h.

3.17 Asynchronous Serial Interface 1 (UART1)

3.17.1 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 bit: 1 bit (2 bits in continuous transmission mode)
3) Parity bit: None/even/odd (data length 8 bits)
   None (data length 7 or 9 bits)
4) Transfer rate: $\frac{16}{3}$ to $\frac{8192}{3}$ Tcyc
5) Operating mode: Programmable transfer mode, fixed-rate transfer mode
6) Transmit data conversion: Normal (NRZ), Manchester encoding
7) Full duplex communication

   The independent transmitter and receiver blocks allow both transmit and receive operations to be performed at the same time. Both transmitter and receiver blocks adopt a double buffer configuration, so that data can be transmitted and received continuously.

3.17.2 Functions

1) Programmable transfer mode
   - Performs full duplex asynchronous serial communication using a data length of 7, 8, or 9 bits with 1 stop bit.
   - The transfer rate of the UART1 is programmable within the range of $\frac{16}{3}$ to $\frac{8192}{3}$ Tcyc.

2) Fixed-rate transfer mode

   Functions as described below only when the system clock is set to a $\frac{1}{1}$ or $\frac{1}{2}$ frequency division of the subclock (X'tal resonator = 32.768 kHz):
   - Performs full duplex asynchronous serial communication using a data length of 8 bits with 1 stop bit.
   - The transfer rate of the UART1 is selectable from among 9600, 4800, and 2400 bps in the $\frac{1}{1}$ frequency division mode and from among 4800, 2400, or 1200 bps in the $\frac{1}{2}$ frequency division mode (Note 1).

3) Continuous data transmission/reception
   - Performs continuous transmission and reception of serial data whose data length and transfer clock rate are fixed. The number of stop bits used in the continuous transmission mode is 2 bits (see Figure 3.17.4) (Note 2).
   - The transfer rate of the UART1 depends on the operating mode.
   - The transmit data is read from the transmit data register (TBUF) and the received data is stored in the receive data register (RBUF).

4) Transmit data conversion
   - The data type of the contents of the transmit data register (TBUF) can be selected from normal output (NRZ) with no conversion and Manchester encoding conversion output.

5) Interrupt generation

Interrupt requests are generated at the end of transmit data transfer, at the end of transmission, and at the end of reception if the interrupt request enable bit is set.

6) It is necessary to manipulate the following special function registers to control the asynchronous serial interface 1 (UART1).
   - UCON0, UCON1, UBR, TBUF, RBUF, UMDSL
   - P3, P3DDR

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	RECENIE
FED1	0000 0000	R/W	UCON1	TRUN	8/9BIT	TDDR	TCMOS	8/7BIT	TBIT8	TEMPTY	TEMPIE
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	0000 0000	R/W	UMDSL	UMB7	UMB6	UMB5	UMXTS1	UMXTS0	UMMCS	TEND	TENIE

Note 1: The values of the transfer rate frequency division ratio select bit (UCON0:UBRSEL) and the baudrate control register (UBR) are invalid in the fixed-rate transfer mode.

Note 2: The number of stop bits of continuous data transmission in the fixed-rate transfer mode is variable between 2 and 4 bits. The number of stop bits with which data can be received continuously is 3 bits or more. These should be taken into consideration when using the UART1.

3.17.3 Circuit Configuration

3.17.3.1 UART1 control register 0 (UCON0) (8-bit register)

1) This register controls the receive operation and interrupts of UART1.

3.17.3.2 UART1 control register 1 (UCON1) (8-bit register)

1) This register controls the transmit operation, data length, and interrupts of UART1.

3.17.3.3 UART1 baudrate control register (UBR) (8-bit register)

1) This is an 8-bit register that defines the transfer rate of UART1 in the programmable transfer mode.
2) It can generate clocks at intervals of $\frac{(n+1)\times 8}{3}$ Tcyc or $\frac{(n+1)\times 32}{3}$ Tcyc (n = 1 to 255, Note: n = 0 is inhibited).

3.17.3.4 UART1 transmit data register (TBUF) (8-bit register)

1) This register is an 8-bit register for storing the data to be transmitted via UART1.

3.17.3.5 UART1 transmit shift register (TSFT) (11-bit shift register)

1) This register is used to send the transmit data via UART1.
2) This register cannot be accessed directly with an instruction. It must be accessed through the transmit data register (TBUF).

3.17.3.6 UART1 receive data register (RBUF) (8-bit register)

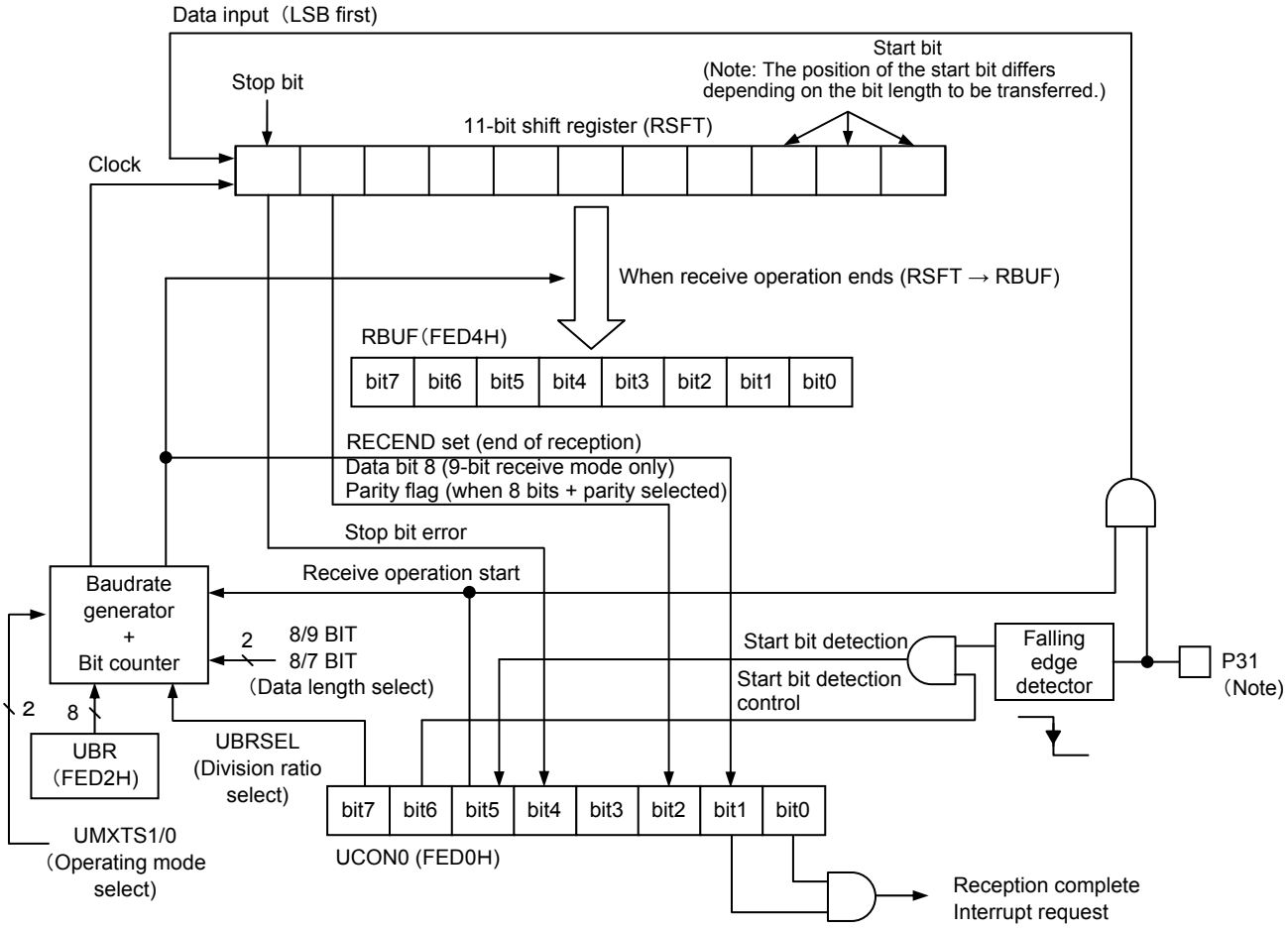
1) This register is an 8-bit register for storing the UART1 receive data.

3.17.3.7 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).

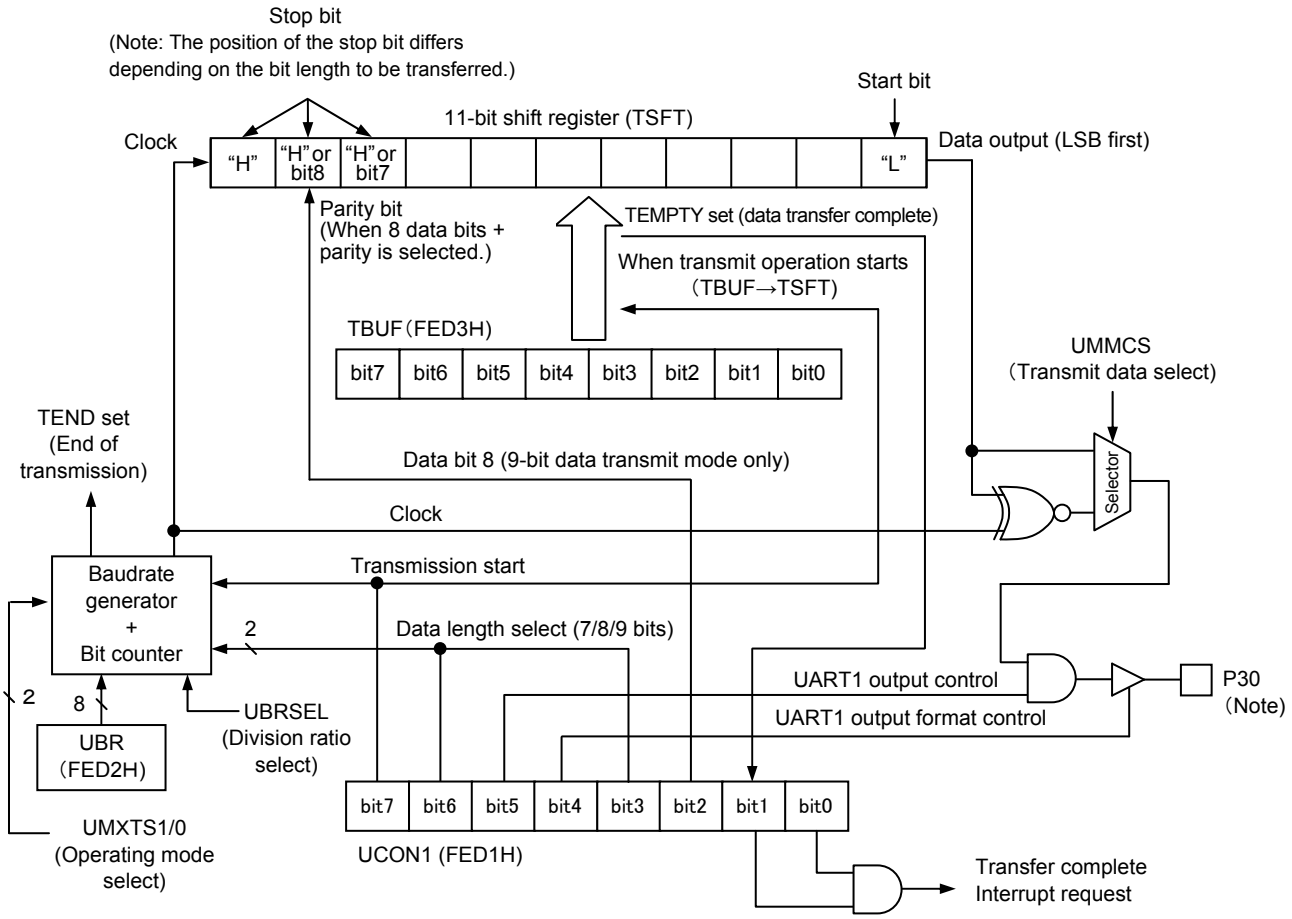
3.17.3.8 UART1 mode select register (UMDSL) (8-bit register)

1) This register is used to select the operating mode, to select the transmit data conversion mode, and to control the transmit interrupt processing of UART1.



Note: Bit 1 of P3DDR (at FE4DH) must be set to 0 when UART1 is to be used in reception mode. UART1 will not function normally if bit 1 is set to 1.

Figure 3.17.1 UART1 Block Diagram (Reception Mode)



Note: Bit 0 of P3DDR (at FE4DH) must be set to 0 when UART1 is to be used in transmission mode. If bit 0 is set to 1, the transmit data is not output.

Figure 3.17.2 UART1 Block Diagram (Transmission mode)

3.17.4 Related Registers

3.17.4.1 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	RECENIE

UBRSEL (bit 7): UART1 transfer rate frequency division ratio select

This bit selects the frequency division ratio of the clock rate in the programmable transfer mode.

<1> When this bit is set to 1, the value range of the transfer rate is from $\frac{64}{3}$ to $\frac{8192}{3}$ Tcyc.

<2> When this bit is set to 0, the value range of the transfer rate is from $\frac{16}{3}$ to $\frac{2048}{3}$ Tcyc.

* UART1 will not run normally if the transfer rate is changed during transmit or receive operation. Be sure to stop UART1 before setting a new transfer rate.
* This bit is disabled in the fixed-rate transfer mode.

STRDET (bit 6): UART1 start bit detection control

<1> Setting this bit to 1 enables the start bit detection (falling edge detection) function and places UART1 in the receive wait state.

<2> Setting this bit to 0 disables the start bit detection (falling edge detection) function.

RECRUN (bit 5): UART1 receive operation flag

<1> This bit is set and a receive operation starts when a falling edge of the signal at the receive port (P31) 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 (when a stop bit is received).

* Set STRDET and RECRUN to 0 at the same time when stopping the receive operation in the receive wait state (STRDET = 1/RECRUN = 0) or during a receive operation (STRDET=1/RECRUN=1).

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 is low.

<2> This bit must be cleared with an instruction.

U0B3 (bit 3): UART1 general-purpose flag/parity select

<1> This bit can be used as a general-purpose flag when either 7- or 9-bit data length is selected.

* Any attempt to manipulate this bit exerts no influence on the operation of this functional block.

<2> This bit specifies the even or odd parity mode when 8 data bits + parity is selected.

U0B3 (When 8 data bits + parity is selected)	Parity
0	Even
1	Odd

* In the transmission mode, the parity (even or odd) is automatically calculated from the TBUF value and appended to the transmit data.

RBIT8 (bit 2): UART1 receive data bit 8 storage bit/receive parity error flag

<1> This bit is loaded with bit 8 of the receive data at the end of receive operation when 9-bit data length is selected.

<2> This bit is loaded with the result of receive parity error check when 8 data bits＋parity is selected.

RBIT8 (when 8 data bits + parity is selected)	Parity Error
0	No error
1	Error occurred

RECEND (bit 1): UART1 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 (RSFT) to the receive data register (RBUF).

<2> This bit must be cleared with an instruction.

RECENIE (bit 0): UART1 receive end interrupt request enable control

<1> When this bit and RECEND are set to 1, an interrupt request to vector address 0033H is generated.

3.17.4.2 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	TEMPTY	TEMPIE

TRUN (bit 7): UART1 transmit 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 (when the transmission of the stop bit (s) finished). (If this bit is cleared in the middle of the transmit operation, the operation is aborted immediately.)

* In the continuous transmission mode, this bit is cleared at the end of a transmit operation, but is automatically set within the same cycle (Tcyc). Consequently, transmit operations occur with intervening 1-Tcyc delays.

8/9BIT (bit 6): UART1 transfer data length select

8/7BIT (bit 3): UART1 transfer data length select

<1> When 8/9BIT is set to 1, the transfer data length is set to 9 bits.

<2> When 8/9BIT is set to 0 and 8/7BIT to 0, the transfer data length is set to 8 bits.

<3> When 8/9BIT is set to 0 and 8/7BIT to 1, the transfer data length is set to 7 bits.

<4> When 8/9Bit is set to 1 and 8/7BIT is set to 1, the transfer data length is set to 8 bits plus parity bit.

* UART1 will not run normally if the data length is changed in the middle of a transmit or receive operation. Be sure to set this bit after stopping the operation.
* The same data length is used when both transmit and receive operations are to be performed at the same time.
* The set values of these bits are disabled in the fixed-rate transfer mode.

8/9BIT	8/7BIT	Data Length
0	0	8 bits
0	1	7 bits
1	0	9 bits
1	1	8 bits + parity

TDDR (bit 5): UART1 transmit port output control

<1> When this bit is set to 1, the transmit data is sent to the transmit port (P30). (No transmit data is generated if bit 0 of P3DDR (FE4DH) is set to 1.)

<2> When this bit is set to 0, no transmit data is placed at the transmit port (P30).

* When this bit is set to 1 in the transmission stopped state (TRUN=0), the transmit port is set to high/open (CMOS/N-channel open drain) if the normal output type is selected (UMDSL: UMMCS=0), and set to low if the Manchester encoding output type is selected (UMDSL: UMMCS=1).
* This bit must always be set to 0 when the transmit 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 (P30) is set to CMOS.

<2> When this bit is set to 0, the output type of the transmit port (P30) is set to N-channel open drain.

TBIT8 (bit 2): UART1 transmit data bit 8 storage bit

<1> This bit stores bit 8 of the transmit data when the data length is set to 9 bits (8/9BIT = 1).

TEMPTY (bit 1): UART1 transmit data transfer end flag

<1> When transmit operation is started, this bit is set when the data transfer from the transmit data register (TBUF) to the transmit shift register (TSFT) ends.

<2> This bit must be cleared with an instruction.

* When performing a continuous 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 before the transmit operation ends, the transmit control bit (TRUN) is automatically set at the end of the transmit operation, starting the next transmit operation.

TEMPIE (bit 0): UART1 transmit-data-transfer-end interrupt request enable control

<1> An interrupt request to vector address 003BH is generated when this bit and TEMPTY are set to 1.

3.17.4.3 UART1 baudrate control register (UBR)

1) This is an 8-bit register that defines the transfer rate of UART1 to be used in the programmable transfer mode.

2) The counter for each baudrate generator is initialized when a transmit or receive operation is stopped (UCON0:RECRUN = 0 or UCON1:TRUN = 0).

3) The transfer rate range can be changed using the transfer rate frequency division ratio select bit (UCON0:UBRSEL).

UBRSEL	TUBR1	Range
0	(UBR value＋1) × $\frac{8}{3}$ Tcyc	$\frac{16}{3}$ to $\frac{2048}{3}$ Tcyc
1	(UBR value＋1) × $\frac{32}{3}$ Tcyc	$\frac{64}{3}$ to $\frac{8192}{3}$ Tcyc

* UART1 will not run normally if the transfer rate is changed in the middle of a transmit or receive operation. Be sure to stop UART1 before setting a new transfer rate.

* The same transfer rate is used when both transmit and receive operations are to be performed at the same time.

* The value of this register is disabled in the fixed-rate transfer mode.

* 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

3.17.4.4 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 TBUF is transferred to the transmit shift register (TSFT) at the beginning of a transmit operation.
   * When performing continuous transmit operation, check the UART1 transmit data transfer end flag (UCON1:TEMPTY) before loading this register with the next transmit data.
   * If the data length is set to 9 bits (UCON1:8/9BIT=1), bit 8 of the transmit data must be placed in 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

3.17.4.5 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.
   * If the data length is set to 9 bits (UCON1:8/9BIT=1), bit 8 of the receive data is transferred to the receive data bit 8 storage bit (UCON0:RBIT8).
   * If the data length is set to 7 bits (UCON1:8/9BIT=0, 8/7BIT= 1), a 0 is transferred to bit R1BUF7.

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

3.17.4.6 UART1 mode select register (UMDSL)

1) This register is an 8-bit register used to select the operating mode, to select the transmit data conversion mode, and to control transmit interrupt processing of UART1.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FED5	0000 0000	R/W	UMDSL	UMB7	UMB6	UMB5	UMXTS1	UMXTS0	UMMCS	TEND	TENIE

UMB7 to UMB5 (bits 7 to 5): UART1 general-purpose flags

<1> These bits can be used as general-purpose flag bits.
   * Any attempt to manipulate these bits exerts no influence on the operation of this functional block.

UMXTS1 (bit 4): UART1 operating mode select

UMXTS0 (bit 3): UART1 operating mode select

The relationship between the UMXTS settings and the operating modes is shown below.

UMXTS1	UMXTS0	Operating Mode	Transfer Rate
0	0	Programmable transfer mode	Variable with register settings
0	1	Fixed-rate transfer mode (1)	9600 bps (4800 bps)
1	0	Fixed-rate transfer mode (2)	4800 bps (2400 bps)
1	1	Fixed-rate transfer mode (3)	2400 bps (1200 bps)

<1> In programmable transfer mode, the data length is variable with the values of the transfer data length select bits (UCON1:8/9BIT, 8/7BIT). The transfer rate can also be changed using the transfer rate frequency division ratio select bit (UCON0:UBRSEL) and the baudrate control register (UBR).

<2> In fixed-rate transfer mode, the data length is fixed at 8 bits and the transfer rate that can be selected is either of 9600/4800/2400 bps.

* UART1 will not run normally if the operating mode is changed during a transmit or receive operation. Be sure to stop UART1 before setting these bits.
* The fixed-rate transfer mode can be used only when a clock with a frequency of $\frac{1}{1}$ or $\frac{1}{2}$ of the subclock frequency (X'tal resonator=32.768 kHz) is selected as the system clock. UART1 will not run normally with any other clock settings. The transfer rates when the $\frac{1}{2}$ frequency division ratio is used are shown in the parentheses.

UMMCS (bit 2): UART1 transmit data conversion select

<1> When this bit is set to 1, the transmit data is subject to Manchester encoding before being transmitted from UART1.

<2> When this bit is set to 0, the transmit data is transmitted as normal (NRZ) output without being subject to conversion.

TEND (bit 1): UART1 transmit end flag

<1> This bit is set and the UART1 operation is stopped at the end of transmission if the UART1 transmit data transfer end flag (UCON1:TEMPTY) is set to 1. If the flag (UCON1:TEMPTY) is set to 0, the continuous transmission mode is on.

<2> This bit must be cleared with an instruction.

TENIE (bit 0): UART1 transmit end interrupt request enable control

<1> An interrupt request to vector address 003BH is generated when this bit and TEND are set to 1.

3.17.5 UART1 Continuous Communication Operation Examples

3.17.5.1 Continuous 8-bit data reception mode (receive data = 55H)

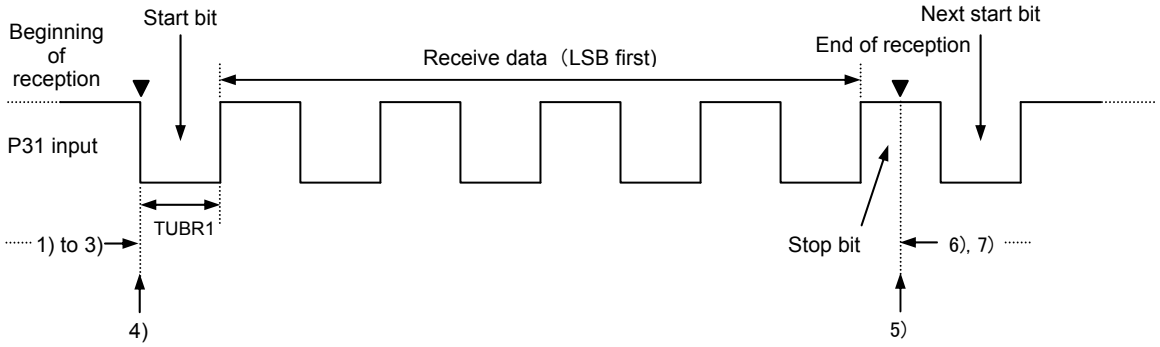


Figure 3.17.3 Example of Continuous 8-bit Data Reception Mode Operation (Programmable Transfer Mode)

1) Setting the transfer rate
   - Set up UCON0:UBRSEL and the UBR register.
   * In the fixed-rate transfer mode, use the values of UMDSL:UMXTS1 and UMXTS0 to define the transfer rate.
2) Setting the data length
   - Set UCON1:8/9BIT to 0 and 8/7BIT to 0.
   * In the fixed-rate transfer mode, the data length is fixed at 8 bits and cannot be set to any other values.
3) Setting the receive port, start bit detection, and interrupts
   - Set P3DDR:P31DDR to 0 and P3:P31 to 0.

     Load UCON0 with X1000001b.
4) Starting a receive operation
   - UCON0:RECRUN is set and UART1 starts a receive operation when a falling edge of the signal at the receive port (P31) is detected.
5) End of a receive operation
   - When the receive operation ends, UCON0:RECRUN is automatically cleared and UCON0: RECEND is set. UART1 then waits for the start bit of the next received data.
6) Receive end interrupt
   - Read the received data from RBUF.
   - Read UCON0:STPERR to check for any communication error.

     (If a communication error is found, clear UCON0:STPERR with the error processing routine.)
   - Clear UCON0:RECEND and exit the interrupt processing routine.
7) Receiving the next data
   - Repeat steps 4) to 6) as shown above.
   * When stopping a continuous receive operation, set UCON0:STRDET and RECRUN to 0 at the same time, and UART1 will stop the receive operation immediately.

Note: The number of stop bits that can be received continuously in the fixed-rate transfer mode is 3 bits or more.

3.17.5.2 Continuous 8-bit data transmission mode (transmit data = 55H)

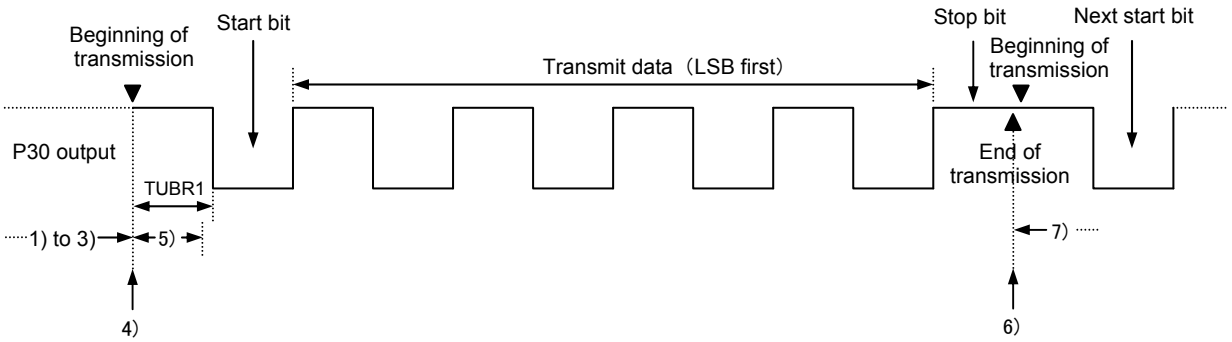


Figure 3.17.4 Example of Continuous 8-bit Data Transmission Mode Operation (Programmable Transfer Mode)

1) Setting the transfer rate
   - Set up UCON0:UBRSEL and the UBR register.
   * In the fixed-rate transfer mode, use the values of UMDSL:UMXTS1 and UMXTS0 to define the transfer rate.
2) Setting the transmit data
   - Load TBUF with 55H.
3) Setting the transmit port, data length, and interrupts
   - Set P3DDR:P30DDR to 0 and P3:P30 to 0.
   - Set UMDSL:TENIE to 1.
   - Load UCON1 with 00110001b.
   * In the fixed-rate transfer mode, the data length is fixed at 8 bits and cannot be set to any other values.
4) Starting a transmit operation
   - Set UCON1:TRUN, and UART1 will start a transmit operation.
5) Transmit data transfer end interrupt
   - Load TBUF with the next transmit data.
   - Clear UCON1:TEMPTY and exit the interrupt processing routine.
6) End of a transmit operation
   - UCON1:TRUN is automatically cleared when UART1 finishes the transmit operation. It is, however, automatically set within the same cycle (Tcyc) (this processing takes 1 Tcyc), after which the transmission of the next data starts.
7) Transmitting the next data
   - Repeat steps 5) and 6) as shown above.
   * If the interrupt processing routine is exited after clearing UCON1:TEMPIE but not clearing UCON1:TEMPTY when terminating a continuous transmit operation in step 5) above, UMDSL: TEND is set at the end of that transmit operation and the transmit operation is stopped on the occurrence of a transmit end interrupt.

Note: The number of stop bits of continuous data transmission in the fixed-rate transfer mode is variable between 2 and 4 bits.

3.17.6 Supplementary Notes on UART1

3.17.6.1 About transmit data conversion

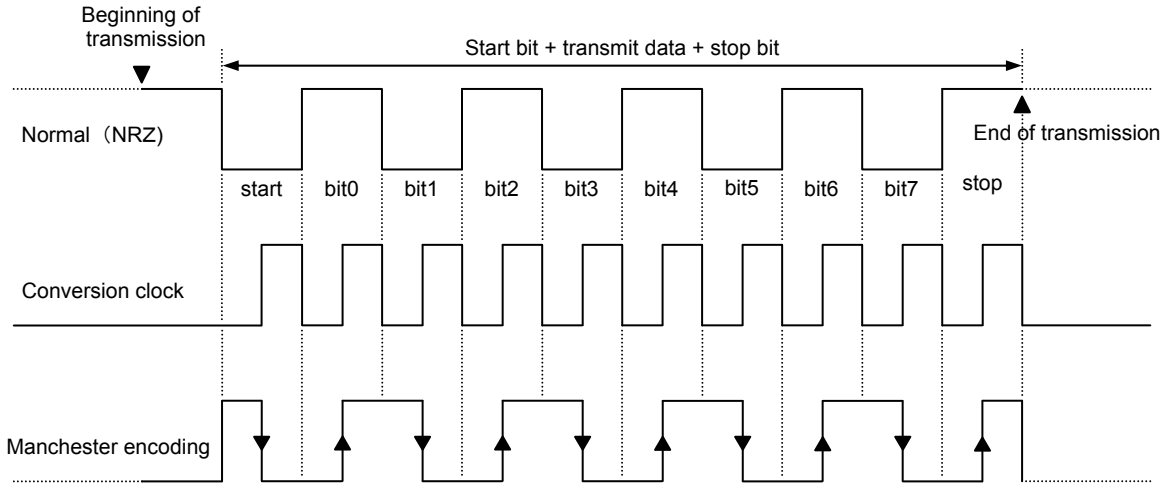


Figure 3.17.5 Example of Transmit Data Conversion (Transmit Data = 55H)

1) The type of the transmit data can be selected from normal (NRZ) output or from Manchester encoding output according to the 0/1 value of UMDSL:UMMCS.

2) The transmit data to be subjected to Manchester encoding consists of 1 start bit + 7/8/9 bits of transmit data + 1 stop bit (one high data bit occurring between the beginning of transmission and the start bit is not subjected to encoding).

* In the fixed-rate transfer mode, the data length is fixed at 8 bits and cannot be set to any other values.

3) When Manchester encoding output is selected, the pre- and post-edge duty cycle within a bit is 50% in the programmable transfer mode but varies with the bit being processed in the fixed-rate transfer mode. The relationship between the bits being processed and the duty cycle is shown below.

- For the start bit and data bit 4, the pre- and post-edge duty cycle within the bit is 50%.
- For data bits 0 to 3, 5 to 7, and stop bit, the pre-and post-edge duty cycle within the bit is approx. 43% vs. 57%.

3.17.6.2 About the fixed-rate transfer mode

1) In the fixed-rate transfer mode, UART1 generates the transfer rate in a special way; the period (bit width) of the data output varies with the bit being transmitted. The relationship between the bits being transmitted and the period is shown below.

When the transfer rate is set to 9600 bps (bit period≒104.16 μs)

- The bit period of the start bit and data bit 4 is approximately 91.55 μs (approx. 87.9% of bit period ≒104.16 μs).
- The bit period of data bits 0 to 3, 5 to 7, and the stop bit is approximately 106.76 μs (approx. 102.5% of bit period≒104.16 μs).

* The above bit period ratio holds for the transfer rate set to 4800, 2400, or 1200 bps.

3.17.6.3 UART1 communication port settings

1) Receive port (P31) settings

Register Data		Receive Port (P31) State	Internal Pull-up Resistor
P31	P31DDR		
0	0	Input	OFF
1	0	Input	ON

* UART1 cannot receive data normally if P31DDR is set to 1.

2) Transmit port (P30) settings

Register Data				Transmit Port (P30) State	Internal Pull-up Resistor
P30	P30DDR	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

* UART1 transmits no data if P30DDR is set to 1.

3.17.7 UART1 HALT Mode Operation

3.17.7.1 Reception mode

1) A UART1 reception mode operation 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 to the receive port after the end of a receive operation.)
2) HALT mode can be released using the UART1 receive interrupt.

3.17.7.2 Transmission mode

1) A UART1 transmission mode operation is enabled in HALT mode. (If the continuous transmision mode is specified when the microcontroller enters HALT mode, UART1 will restart transmission after terminating a transmit operation. Since UCON1:TEMPTY cannot be cleared in this case, UART1 stops operation after completing that transmission.)
2) HALT mode can be released using the UART1 transmit interrupt.

3.18 PWM0 and PWM1

3.18.1 Overview

This series of microcontrollers incorporates two 12-bit PWMs, i.e., PWM0 and PWM1. Each PWM consists of a PWM generator circuit that generates multifrequency 8-bit fundamental PWM waves and a 4-bit additional pulse generator.

PWM0 and PWM1 have dedicated input/output pins PWM0 and PWM1, respectively.

3.18.2 Functions

1) PWM0: Fundamental wave 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 pin function
   - The PWM0 and PWM1 pins can also be used as input ports.
   - PWM0 can also serve as large-current P-channel driver output and PWM1 as large-current N-channel driver output.
   - PWM0 and PWM1 are multiplexed with AN9 and AN8 analog input channels, respectively.

7) It is necessary to manipulate the following special function registers to control PWM0 and PWM1.

- 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

3.18.3 Circuit Configuration

3.18.3.1 PWM0/PWM1 control register (PWM0C) (8-bit register)

1) This register controls the operation and interrupts of PWM0 and PWM1.

3.18.3.2 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 low-order 4 bits are read as 1.
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.

3.18.3.3 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.

3.18.3.4 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 low-order 4 bits are read as 1.
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.

3.18.3.5 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.

3.18.3.6 PWM01 port input register (PWM01P) (2-bit register)

1) PWM0 pin data can be read into this register as bit 0.
2) PWM1 pin data can be read into this register as bit 1.

3.18.4 Related Registers

3.18.4.1 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) × $\frac{16}{3}$ Tcyc

Overall period = Fundamental wave period × 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.

3.18.4.2 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 the low-order 4 bits are read as 1.
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

3.18.4.3 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) × $\frac{1}{3}$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

3.18.4.4 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 low-order 4 bits are read as 1.
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

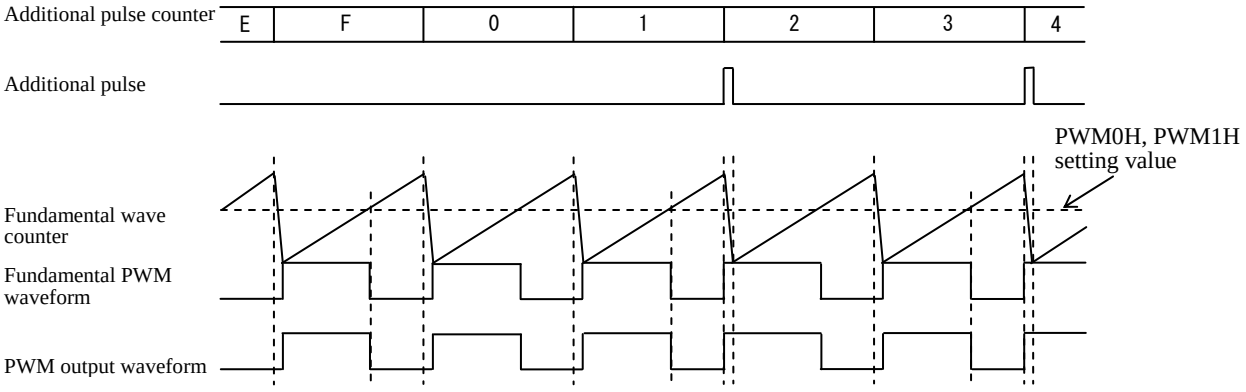
3.18.4.5 PWM1 compare register H (PWM1H) (8-bit register)

1) This register controls the fundamental wave pulse width of PWM1.

Fundamental wave pulse width = (Value represented by PWM1H7 to PWM1H0) × $\frac{1}{3}$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.

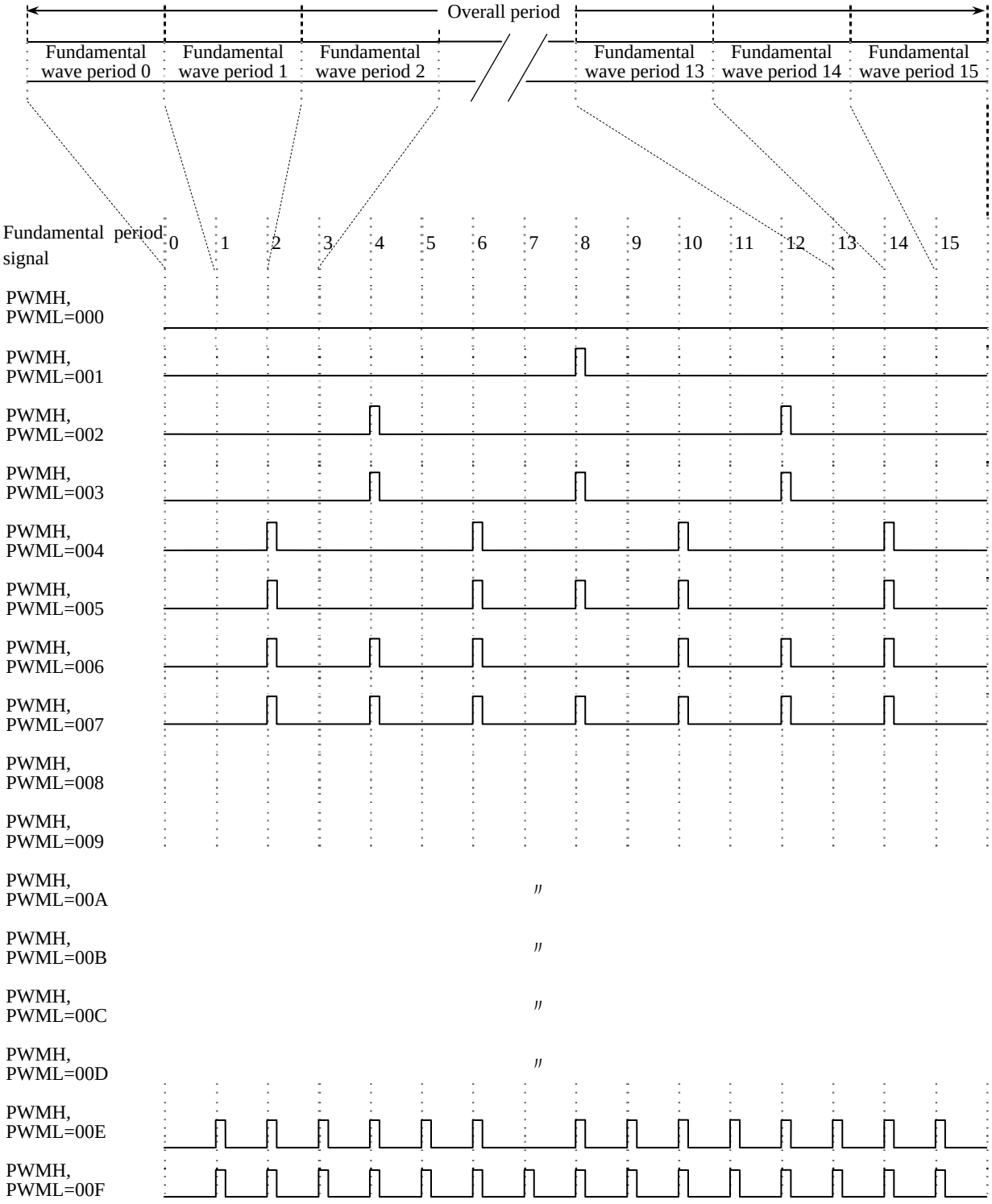
Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE23	0000 0000	R/W	PWM1H	PWM1H7	PWM1H6	PWM1H5	PWM1H4	PWM1H3	PWM1H2	PWM1H1	PWM1H0



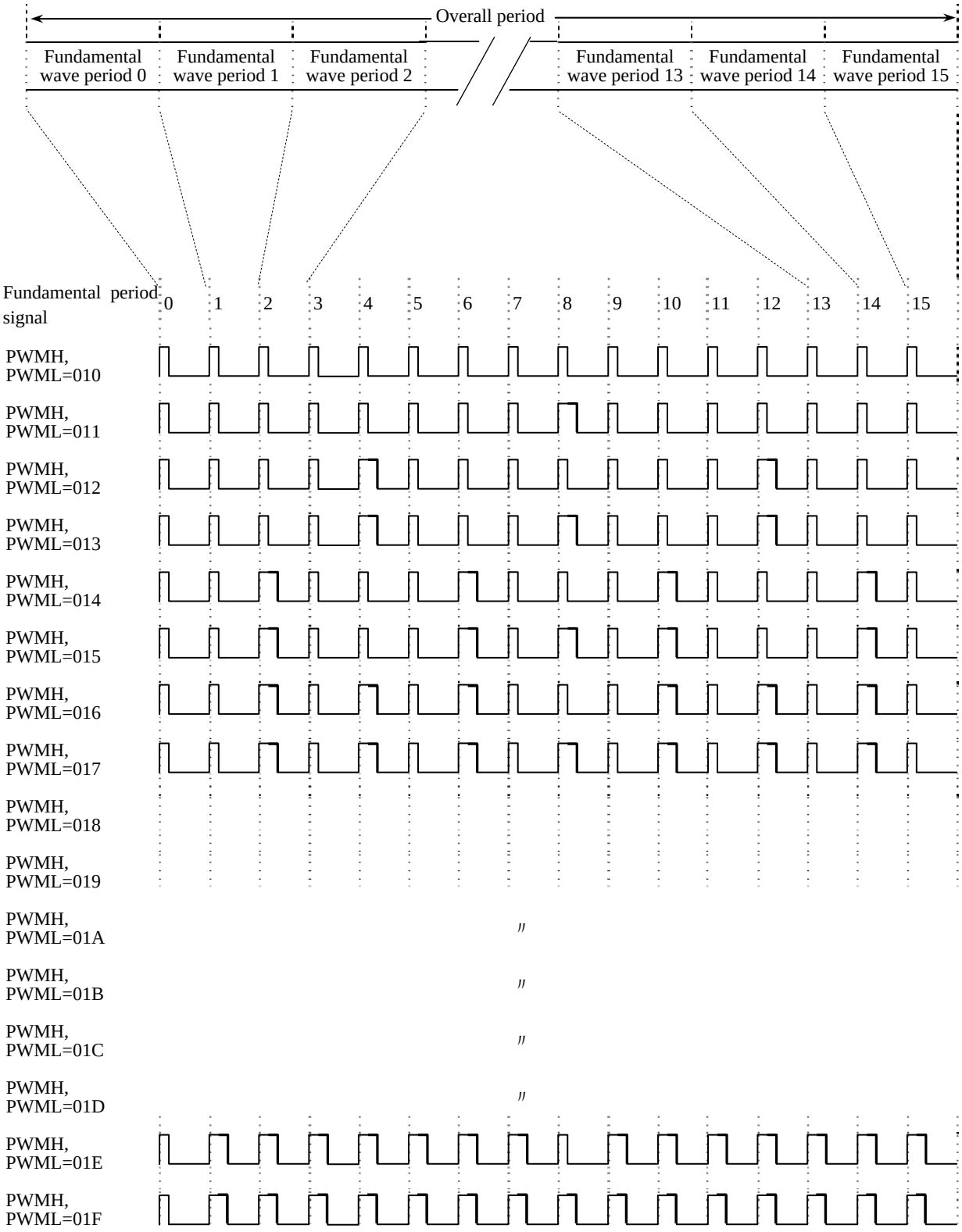
PWM

- 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]



- 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]

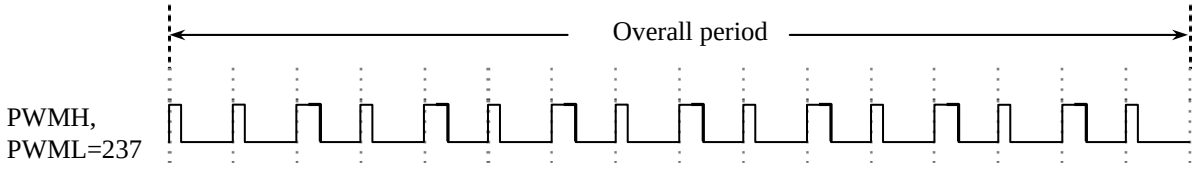


- The fundamental wave period is variable within the range of $\frac{(16 \text{ to } 256)}{3}$ Tcyc.
  Fundamental wave period = (Value represented by PWM0C7 to PWM0C4 + 1) × $\frac{16}{3}$ Tcyc
  - The overall period can be changed by changing the fundamental wave period.
  - The overall period is made up of 16 fundamental wave periods.

Examples:

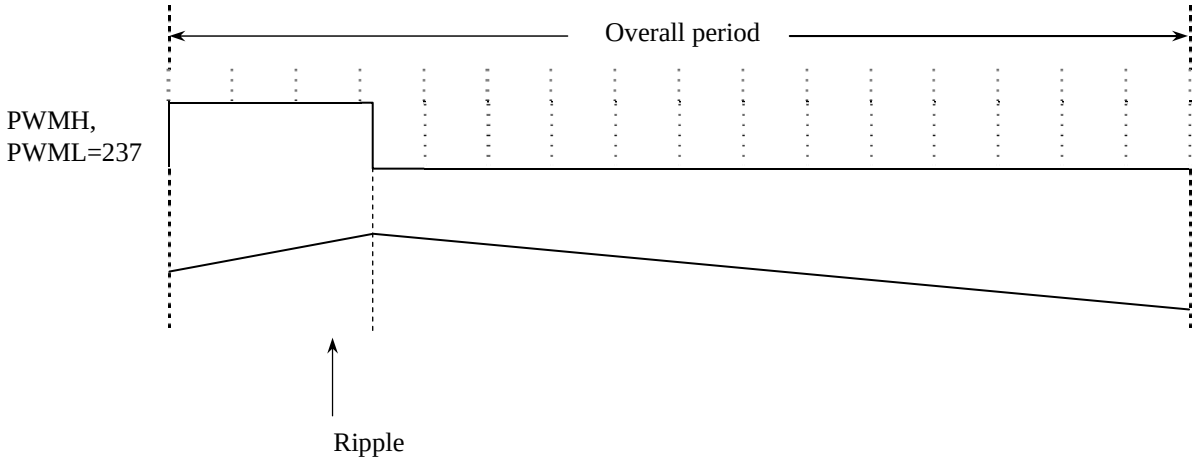
- Wave comparison when the 12-bit PWM contains 237[H].
  12-bit register configuration → (PWMH), (PWML) = 237[H]

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.



3.18.4.6 PWM01 port input register (PWM01P) (2-bit register)

1) PWM0 pin data can be read into this register as bit 0.
2) PWM1 pin 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)

3.19 AD Converter (ADC12)

3.19.1 Overview

This series of microcontrollers incorporates a 12-bit resolution AD converter that has the features listed below. It allows the microcontroller to capture 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

3.19.2 Functions

1) Successive approximation
   - The AD converter has a resolution of 12 bits.
   - Some conversion time is required 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. The AD mode register (ADMRC) is used to select the AD conversion mode.

3) 12-channel analog input

   The signal to be converted is selected using the AD control register (ADCRC) from 12 types of analog signals that are supplied from pins P0, P70, P71, XT1, and XT2.

4) Conversion time select

   The AD conversion time can be set from 1/1 to 1/128 (frequency division ratio). The AD mode register (ADMRC) and AD conversion result register low byte (ADRLC) are used to select the conversion time for appropriate AD conversion.

5) Automatic reference voltage generation control

   The AD converter incorporates a reference voltage generator that automatically generates the reference voltage when an AD conversion starts and stops generation when the conversion ends. For this reason, 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

3.19.3 Circuit Configuration

3.19.3.1 AD conversion control circuit

1) This circuit runs in two modes: 12- and 8-bit AD conversion modes.

3.19.3.2 Comparator circuit

1) This 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 conversion end flag (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).

3.19.3.3 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.

3.19.3.4 Automatic reference voltage generator circuit

1) The automatic 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 automatically stopped when the conversion ends. The reference voltage output ranges from VDD to VSS.

3.19.4 Related Registers

3.19.4.1 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):
ADCHSEL1 (bit 5):
ADCHSEL0 (bit 4):
} **AD conversion input signal select**

These 4 bits are used to select the signal to be subject to AD conversion.

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/AN9
1	0	1	0	XT1/AN10
1	0	1	1	XT2/AN11

ADCR3 (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 amount of time specified by the conversion time control register is required to complete the conversion. The conversion time is defined using three bits, i.e., ADTM2 (bit 0) of the AD conversion result register low byte (ADRLC) and 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 when AD conversion is in progress.

ADENDF (bit 1): AD conversion end 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 prohibited to set ADCHSEL3 to ADCHSEL0 to a 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.*

3.19.4.2 AD mode register (ADMRC)

1) This register is an 8-bit register that controls the operation mode of the AD converter.

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).

When this bit is set to 1, the AD converter operates 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.

When this bit is set to 0, the AD converter operates as a 12-bit AD converter. The conversion results are placed in the AD conversion result register high byte (ADRHC) and the high-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 bit 0 (ADTM2) 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

Conversion time calculation formulas

- 12-bit AD conversion mode: Conversion time = ((52/(AD division ratio)) + 2) × 1/3 × Tcyc
- 8-bit AD conversion mode: Conversion time = ((32/(AD division ratio)) + 2) × 1/3 × 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 required in the second and subsequent conversions or in AD conversions that are carried out in the 8-bit AD conversion mode.*

3.19.4.3 AD conversion result register low byte (ADRLC)

1) This register is used to hold the low-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):
} **Low-order 4 bits of AD conversion results**

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 ADTM1 (bit 1) and ADTM0 (bit 0) of the AD mode register (ADMRC) are used to control the conversion time. See the Subsection on the AD mode register for the procedure to set up the conversion time.

Note:

The conversion result data contains errors (quantization error + combination error). Be sure to use only valid conversion results based on specifications provided in the latest "SANYO Semiconductors Data Sheet."

3.19.4.4 AD conversion result register high byte (ADRHC)

1) This register is used to hold the high-order 8 bits of the results of an AD conversion that is carried out in 12-bit AD conversion mode. The register stores the entire 8 bits of an AD conversion that is carried out in 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

3.19.5 AD Conversion Example

3.19.5.1 12-bit AD conversion mode

1) Setting up the 12-bit AD conversion mode
   - Set ADMD3 (bit 6) of the AD mode register (ADMRC) to 0.

2) Setting up the conversion time
   - To set the conversion time to 1/32 frequency division, 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 the 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 when the AD conversion is carried out for the first time after a system reset or after the AD conversion mode is switched from 8-bit to 12-bit AD conversion mode. The conversion time determined by the formula is required in the second and subsequent conversions.

5) Detecting the AD conversion end flag
   - Monitor ADENDF (bit 1) of the AD control register (ADCRC) until it is set to 1.
   - Clear the conversion end 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 register high byte (ADRHC) and AD conversion result register low byte register (ADRLC). Since the conversion result data contains errors (quantization error + combination error), use only the valid part of the conversion data selected according to the specifications provided in the latest “SANYO Semiconductors Data Sheet.”
   - Send the above read data to application software processing.
   - Return to step 4) to repeat conversion processing.

3.19.6 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 operation.

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 AD conversion end 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 by the formula given in the paragraph entitled "Conversion time calculation formulas" is required in the second and subsequent conversions or in AD conversions that are carried out in the 8-bit AD conversion mode.

7) The conversion result data contains errors (quantization error + combination error). Be sure to use only valid conversion results based on specifications provided in 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 higher than VDD or lower than VSS to an input pin may exert an adverse influence on the converted value of the channel in question or of other channels.

9) Take the following measures to prevent a reduction in conversion accuracy due to noise interference, etc.:
   - Add external bypass capacitors of several μF plus thousands of 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, very close to the analog input pins. To avoid any adverse coupling influences, use a ground that is free of noise interference 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. Or, 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 interference if the state of port outputs is changing. To minimize the adverse influences of noise interference, it is necessary to keep line resistance across the power supply and the VDD pins of the microcontroller at a 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 several times, discard the maximum and minimum values of the conversion results, and use an average of the remaining data.

3.20 Audio Interface

3.20.1 Overview

The audio interface incorporated in this series of microcontrollers is a 3-wire serial digital audio interface with the following features:

1) Data length: 16 / 18 / 20 / 24 bits
2) Sampling frequency (fs): 8 / 11.025 / 12 / 16 / 22.05 / 24 / 32 / 44.1 / 48 kHz
3) Master clock: 256 / 384 fs
4) Bit clock: 48 / 64 fs
5) Left-justified / right-justified / I^2 S format selectable
6) LSB first / MSB first selectable

3.20.2 Functions

1) 3-wire serial digital audio interface
   - Can transmit or receive 2 to 8190 bytes of data consecutively.
   - The transmit data is automatically transferred from RAM to a shift buffer, and the receive data is automatically transferred from the shift buffer to RAM.
   - The RAM area for storing transmit and receive data can be allocated to arbitrary addresses.
   - The master clock can be selected from among the external input clock (PWM1 pin), the clock generated by the internal PLL circuit, and the system clock.

2) Synchronous serial interface

   The audio interface can also serve as a synchronous serial interface with parity check that can run either in master (transmit) or slave (receive) mode.

3) Interrupt generation

   An interrupt request is generated at the beginning and end of transmit or receive operation if the interrupt request enable bit is set.

4) It is necessary to manipulate the following special function registers to control the audio interface.
   - AIFRPG, AIFSTC, AIFDSL, AIFRAT, AIFCON, AIFMOD, AIFCLK, AIFPC, AIFCNL, AIFCNH, AIFADL, AIFADH
   - P0, P3, P3DDR, PWM0C, PWM0L

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA8	0000 0000	R/W	AIFRPG	MCKDV2	MCKDV1	MCKDV0	FMTCEN	PRTYEN	AEROE	AERDET	AERIE
FEA9	0000 0000	R/W	AIFSTC	ARUNOE	MCKSEL	ASTREN	ASTRSTP	ASTPCEN	ASTPCHI	ASTPSL1	ASTPSL0
FEAA	HHHH H000	R/W	AIFDSL	-	-	-	-	-	ADTOSL2	ADTOSL1	ADTOSL0
FEAB	0000 0000	R/W	AIFRAT	APLTST0	MCLKPL	BCLKSL	AIFSWP	AFLTSL1	AFLTSL0	RATESEL1	RATESEL0
FEAC	0000 0000	R/W	AIFCON	APLTST1	ACLKON	AIFRUN	AIFOFF	AIFWRT	AIFDIR	AENDIF	AENDIE
FEC9	0000 0000	R/W	AIFMOD	AIFLN1	AIFLN0	AIFFB	SDATISEL	IISEN	MCLKIN	ASTAIF	ASTAIE
FECA	0000 0000	R/W	AIFCLK	AVCOON	ACMPON	AIFCK1	AIFCK0	AVCOSEL	LRCKPL	BCLKPL	BCKSEL
FECB	0000 0000	R/W	AIFPC	BCLKLO	LRCKIN	BCLKIN	SDATIN	MCLKDR	LRCKDR	BCLKDR	SDATDR
FECC	0000 0000	R/W	AIFCNL	ACNT7	ACNT6	ACNT5	ACNT4	ACNT3	ACNT2	ACNT1	ACNT0
FECD	HHH0 0000	R/W	AIFCNH	-	-	-	ACNT12	ACNT11	ACNT10	ACNT9	ACNT8
FECE	0000 0000	R/W	AIFADL	AADR7	AADR6	AADR5	AADR4	AADR3	AADR2	AADR1	AADR0
FECF	HHH0 0000	R/W	AIFADH	-	-	-	AADR12	AADR11	AADR10	AADR9	AADR8

3.20.3 Circuit Configuration

The audio interface control circuit consists of the functional blocks shown below.

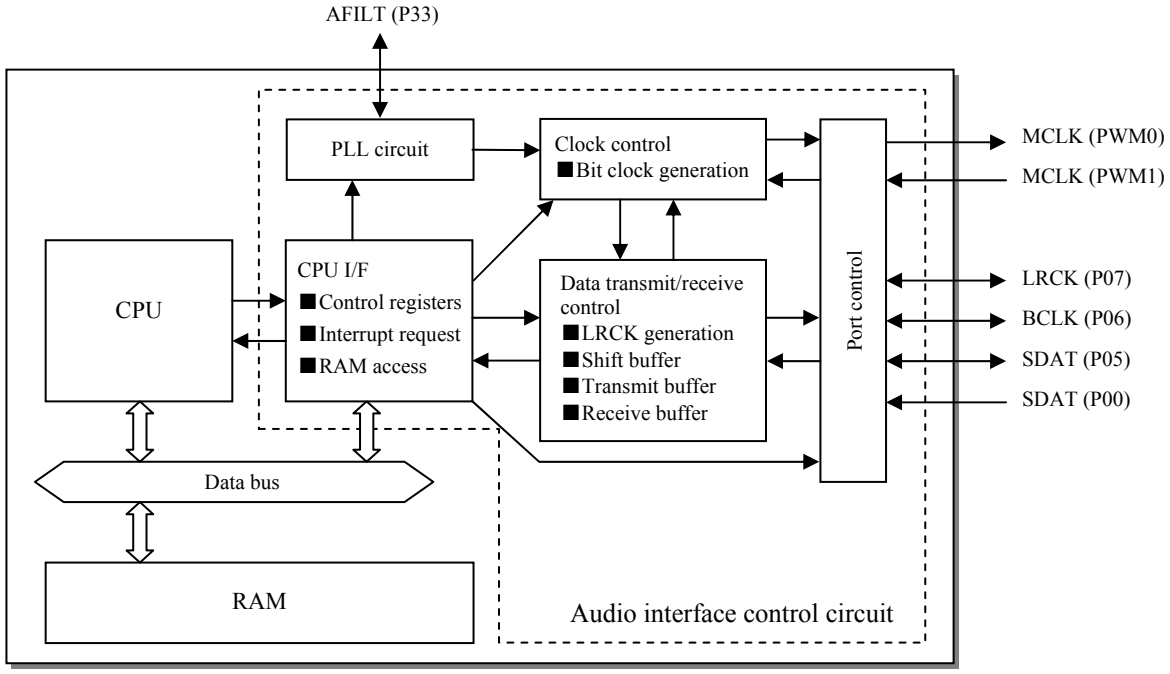


Figure 3.20.1 Audio Interface Control Circuit Block Diagram

3.20.3.1 Audio interface PLL circuit

1) The internal PLL generates the master clock for the audio interface.

3.20.3.2 Clock control circuit

1) The master clock for the audio interface is selected from the following clock sources:
   <1> Clock generated by the internal PLL
   <2> External clock supplied from the PWM1 pin
   <3> System clock
2) The master clock is frequency-divided to generate the bit clock.

3.20.3.3 CPU interface circuit

1) This circuit has registers for controlling the operation of the audio interface circuit.
2) This circuit generates interrupt requests to the CPU.
3) This circuit transfers transmit data from RAM to the transmit buffer.
4) This circuit transfers receive data from the receive buffer to RAM.

3.20.3.4 Data transmit/receive control circuit

1) This circuit generates the LR clock.
2) This circuit loads data from the transmit buffer into a shift register for serial transmission.
3) This circuit loads the received serial data into a shift register and saves it in the receive buffer.

3.20.3.5 Related I/O pins

1) The I/O pins related with the audio interface control circuit are shown in the table below.

Table 3.20.1 Related I/O Pins

Pin Name	I/O	Description
P00	I	Input pin for SDAT (serial data)
P05	I/O	I/O pin for SDAT (serial data)
P06	I/O	I/O pin for BCLK (bit clock)
P07	I/O	I/O pin for LRCK (LR clock)
P23	O	AIFRUN flag output pin
P70	I	Input pin for continuous transfer suspension control
P71	I	Input pin for continuous transfer suspension control
	O	AIFRUN flag output pin
P72	I	Input pin for continuous transfer suspension control
	O	AIFRUN flag output pin
P73	I	Input pin for continuous transfer suspension control
	O	AIFRUN flag output pin
PWM0	I	MCLK (master clock) input pin
PWM1	O	MCLK (master clock) output pin
P32	O	Error detection flag output pin
P33	I/O	Pin connected to internal PLL filter circuit (Figure 3.20.3)

3.20.4 Related Registers

3.20.4.1 Synchronous serial interface control register 1 (AIFRPG)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA8	0000 0000	R/W	AIFRPG	MCKDV2	MCKDV1	MCKDV0	FMTCEN	PRTYEN	AEROE	AERDET	AERIE

MCKDV2 (bit 7): External output clock select

<1> When this bit is set to 1, the clock source is transmitted from the PWM0 pin without being frequency-divided.

<2> When this bit is set to 0, a frequency-divided clock is transmitted from the PWM0 pin. The frequency division ratio (1/1 to 1/8) is determined by MCKDV1 and MCKDV0.

MCKDV1 (bit 6): Master clock frequency division ratio select

MCKDV0 (bit 5): Master clock frequency division ratio select

<1> These bits define the frequency division ratio (1/1 to 1/8) of the clock source to be supplied as the master clock.

Table 3.20.2 Master Clock Frequency Division Ratio Select

MCKDV[1:0]	Master Clock Frequency Division Ratio
00	1/1
01	1/2
10	1/4
11	1/8

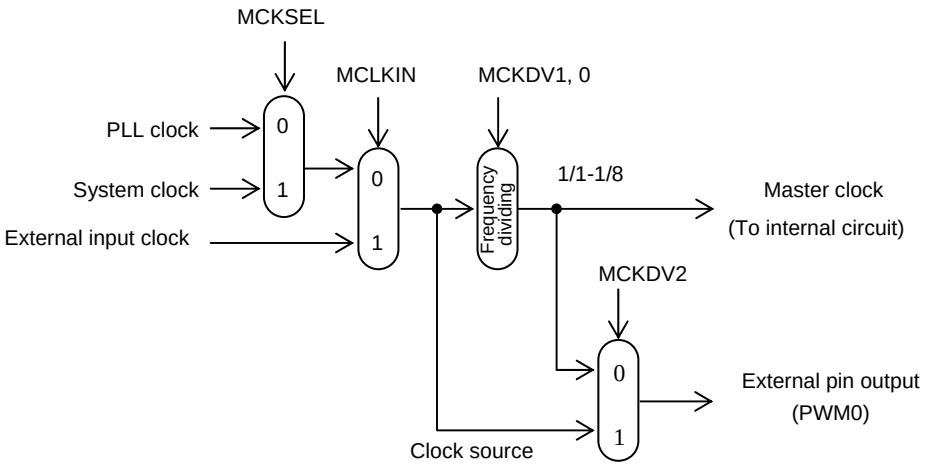


Figure 3.20.2 Master Clock/External Output Clock Select

FMTCEN (bit 4): LRCK format check setting

<1> Set this bit to 1 to use the LRCK format check function.

<2> This bit is used to check the format to determine whether the condition BCLK=48 fs or 64 fs is established in slave mode. The error flag (AERDET) is set when a format error is detected.

PRTYEN (bit 3): Parity check setting

<1> Set this bit to 1 to enable the parity check function of the synchronous serial interface (master/slave mode).

<2> In this case, the bit length must be set to 16 bits (AIFLN1=0, AIFLN0=0) and the BCLK frequency selection must be set to 48 fs (BCKSEL=0).

<3> In master mode, 1 byte of parity data (even parity) is appended to 2 bytes of transmit data. The parity data is placed in the third byte position if the left-justified format is used and in the first byte position if the right-justified format is used.

<4> In slave mode, parity check is performed on the receive data and the error flag (AERDET) is set if a parity error is detected.

(Example) Parity data is 26[H] if the transmit data is 1234[H].

AEROE (bit 2): Error detection flag output setting

<1> Set this bit to 1 to output the value of the error detection flag (AERDET) from the P32 pin. In this case, P32DDR and P32 must be set to 1 and 0, respectively.

AERDET (bit 1): Error detection flag

<1> This bit is set to 1 automatically when an LRCK format error or parity error is detected in slave mode.

<2> This bit must be cleared with an instruction.

AERIE (bit 0): Error interrupt request enable flag

<1> When this bit and AERDET are set to 1, an interrupt request to vector address 004BH is generated.

3.20.4.2 Synchronous serial interface control register 2 (AIFSTC)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA9	0000 0000	R/W	AIFSTC	ARUNOE	MCKSEL	ASTREN	ASTRSTP	ASTPCEN	ASTPCHI	ASTPSL1	ASTPSL0

ARUNOE (bit 7): AIFRUN flag output setting

<1> This bit must be set to 1 to transmit the value of the AIFRUN flag from an external pin.

<2> The output pin (P23, P71, P72, or P73) is selected by ASTPSL1 and ASTPSL0.
The direction register bit for the selected pin must be set to 1 and the related data latch to 0.

Table 3.20.3 AIFRUN Flag Output Pin Select

ASTPSL[1:0]	Pin Selection	Pin Register Settings
00	P23	P23DDR=1, P23=0
01	P71	P71DDR=1, P71=0
10	P72	P72DDR=1, P72=0
11	P73	P73DDR=1, P73=0

<3> It is possible to invert the value of the AIFRUN flag by setting ASTPCHI to 1.

Table 3.20.4 AIFRUN Flag Output Polarity Settings

ASTPCHI	Polarity Setting
0	Polarity is not inverted on output
1	Polarity is inverted on output

MCKSEL (bit 6): Master clock source select (see Figure 3.20.2)

<1> A 1 in this bit selects the system clock. This bit must be set to 1 when the synchronous serial interface is used in master mode.

<2> A 0 in this bit selects the clock that is generated by the PLL circuit.

ASTREN (bit 5): Synchronous serial interface master mode select

<1> Set this bit to 1 to select the synchronous serial interface master mode.

<2> This bit must be set to 0 for the audio interface mode or synchronous serial interface slave mode.

* Be sure to set ACLKON to 0 when the value of this bit is changed.

ASTRSTP (bit 4): Continuous transfer suspension control (software-controlled suspension)

<1> Set this bit to 1 to suspend the continuous transfer in the synchronous serial interface master mode. The continuous transfer operation is suspended after the transmission of the current L-channel or R-channel data is completed.

<2> Set this bit to 0 to cancel the suspension.

ASTPCEN (bit 3): Continuous transfer suspension pin control (hardware-controlled suspension)

<1> Set this bit to 1 to control the suspension of the continuous transfer in the synchronous serial interface master mode according to the input level (either H or L level) of the pin. The continuous transfer operation is suspended after the transmission of the current L-channel or R-channel data is completed.

<2> The suspension control pin (P70, P71, P72, or P73) is selected by ASTPSL1 and ASTPSL0.

Table 3.20.5 Continuous Transfer Suspension Pin Select

ASTPSL[1:0]	Pin Selection
00	P70
01	P71
10	P72
11	P73

<3> The control polarity (H or L level) can be changed by ASTPCHI.

Table 3.20.6 Continuous Transfer Suspension Pin Polarity Setting

ASTPCHI	Polarity Setting
0	Suspended by L level, released by H level
1	Suspended by H level, released by L level

ASTPCHI (bit 2):

<1> AIFRUN flag output polarity setting (see Table 3.20.4)

<2> Continuous transfer suspension pin polarity setting (see Table 3.20.6)

ASTPSL1 (bit 1):

ASTPSL0 (bit 0):

<1> AIFRUN flag output pin select (see Table 3.20.3.)

<2> Continuous transfer suspension pin select (see Table 3.20.5)

3.20.4.3 Output select control register (AIFDSL)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEAA	HHHH H000	R/W	AIFDSL	-	-	-	-	-	ADTOSL2	ADTOSL1	ADTOSL0

ADTOSL2 (bit 2): P07 pin output select

<1> When this bit is set to 1, the input value from the UHBD+ pin is through output.

<2> When this bit is set to 0, LRCK from the audio interface is output.

ADTOSL1 (bit 1): P06 pin output select

<1> When this bit is set to 1, the input value from the UHBD- pin is through output.

<2> When this bit is set to 0, BCLK from the audio interface is output.

ADTOSL0 (bit 0): P05 pin output select

<1> When this bit is set to 1, the input value from the P25 pin is through output.

<2> When this bit is set to 0, SDAT from the audio interface is output.

3.20.4.4 Master clock frequency control register (AIFRAT)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEAB	0000 0000	R/W	AIFRAT	APLTST0	MCLKPL	BCLKSL	AIFSWP	AFLTSL1	AFLTSL0	RATESEL1	RATESEL0

APLTST0 (bit 7): Test bit

This bit must always be set to 0.

MCLKPL (bit 6): Master clock polarity select

<1> When this bit is set to 1, bit clocks are generated in synchronization with the falling edges of the master clock.

<2> When this bit is set to 0, bit clocks are generated in synchronization with the rising edges of the master clock.

BCLKSL (bit 5):

This bit must always be set to 0.

AIFSWP (bit 4): High-order/low-order bit switching control (endian switching control)

<1> Set this bit to 1 to switch the high-order and low-order bits in 1-byte units when transferring data between RAM and transmit/receive buffer.

<2> Using this bit with AIFDIR makes it possible to transmit RAM data in big endian mode or store receive data in RAM in big-endian mode.

Table 3.20.7 Endian Switching

AIFSWP	AIFDIR	Endian	LSB/MSB
0	0	Little endian	LSB first
	1		MSB first
1	0	Big endian	MSB first
	1		LSB first

AFLTSL1 (bit 3): Charge pump output resistance change

AFLTSL0 (bit 2): Charge pump output resistance change

Table 3.20.8 Charge Pump Output Resistance Change

AFLTSL[1:0]	Charge Pump Output Resistance Value
00	Approx. 1.2 kΩ
01	Approx. 3.3 kΩ
10	Approx. 0.6 kΩ
11	Inhibited

RATESEL1 (bit 1): Master clock/bit clock frequency division ratio setting

RATESEL0 (bit 0): Master clock/bit clock frequency division ratio setting

<1> Bit clocks are generated by frequency-dividing the master clock.

<2> These 2 bits and bit 0 (BCKSEL) of AIFCLK determine the frequency division ratio of the master clock/bit clock.

Table 3.20.9 Master Clock/Bit Clock Frequency Division Ratio

BCKSEL	RATESEL	Ratio	Bit Clock	Master Clock
0	00	1/8	48 fs	384 fs
	01	1/4		192 fs
	1X	1/2		96 fs
1	00	1/6	64 fs	384 fs
	01	1/4		256 fs
	1X	1/2		128 fs

3.20.4.5 Audio interface control register (AIFCON)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEAC	0000 0000	R/W	AIFCON	APLTST1	ACLKON	AIFRUN	AIFOFF	AIFWRT	AIFDIR	AENDIF	AENDIE

APLTST1 (bit 7): Test bit

This bit must always be set to 0.

ACLKON (bit 6): Operation start control

<1> Setting this bit to 1 places the audio interface circuit into the operating state. In audio interface mode, generation of LRCK and BCLK is started. This bit must also be set to 1 when the audio interface circuit is to be used in synchronous serial interface mode.

<2> Setting this bit to 0 initializes the audio interface circuit and places it into the stopped state.

AIFRUN (bit 5): Data transfer start control

<1> Setting this bit to 1 starts data transfer between RAM and the transmit/receive buffer.

<2> When LRCK and BCLK are to be supplied from external clock inputs in audio interface mode, set ACLKON to 1 and, after a lapse of 2 or more LRCK cycles, set AIFRUN to 1.

<3> Clearing this bit during operation stops the data transfer between RAM and transmit/receive buffer in the middle of the operation.

AIFOFF (bit 4): Automatic AIFRUN flag clear control

<1> When this bit is set to 1, AIFRUN is automatically cleared at the end of the transfer of the specified number of data bytes between RAM and the transmit/receive buffer.

<2> AIFRUN is not automatically cleared when this bit is set to 0.

<3> This bit must be set to 1 when the audio interface circuit is to be used in synchronous serial interface mode (master or slave mode).

AIFWRT (bit 3): Transmit/receive switching control

<1> This bit must be set to 1 to perform receive operations (data transfer between the receive buffer and RAM) (including simultaneous transmit and receive operations).

<2> This bit must be set to 0 when only transmit operations are to be performed.

AIFDIR (bit 2): MSB first /LSB first select

<1> A 1 in this bit selects MSB first.

<2> A 0 in this bit selects LSB first.

AENDIF (bit 1): Data transfer end flag

<1> This bit is set to 1 automatically when the transfer of the specified number of data bytes between RAM and the transmit/receive buffer is completed.

<2> After this bit is set to 1, 1 LRCK cycle lapsed so that the data transfer operation in the transmit mode is completed.

<3> This bit must be cleared with an instruction.

AENDIE (bit 0): Transfer end interrupt request enable flag

<1> When this bit and AENDIF are set to 1, an interrupt request to vector address 003BH is generated.

3.20.4.6 Mode setting register (AIFMOD)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC9	0000 0000	R/W	AIFMOD	AIFLN1	AIFLN0	AIFFB	SDATISEL	UPLSEL	MCLKIN	ASTAIF	ASTAIE

AIFLN1 (bit 7): Data length select

AIFLN0 (bit 6): Data length select

Table 3.20.10 Data Length Select

AIFLN[1:0]	Data Length
00	16 bits
01	18 bits
10	20 bits
11	24 bits

AIFFB (bit 5): Left-justified/right-justified select

<1> A 1 in this bit selects right-justified mode.

<2> A 0 in this bit selects left-justified mode.

SDATISEL (bit 4): SDAT input pin select

<1> Setting this bit to 1 assigns the SDAT input pin to P00.

<2> Setting this bit to 0 assigns the SDAT input pin to P05.

IISEN (bit 3): I^2S mode select

<1> Setting this bit to 1 with AIFDIR=1, AIFFB=0, and LRCKPL=1 selects the I^2S format.

<2> When this bit is set to 0, the left- or right-justified format is selected.

MCLKIN (bit 2): Master clock source select (see Figure 3.20.2)

<1> When this bit is set to 1, the external input clock from the PWM1 pin is selected as the clock source for the master clock.

<2> When this bit is set to 0, the clock designated by MCKSEL (PLL clock/system clock) is selected as the clock source for the master clock.

ASTAIF (bit 1): Transfer start flag

<1> This bit is automatically set to 1 when data transfer between RAM and the transmit/receive buffer is started (when the first transmit data is transferred from RAM to the transmit buffer or when the first receive data is transferred from the receive buffer to RAM).

<2> When this bit is automatically set, the values of the transfer count setting register and transfer RAM address setting register are saved in internal registers. Consequently, it becomes possible to alter the values of the transfer count setting register and transfer RAM address setting register in preparation for the next transfer.

<3> This bit must be cleared with an instruction.

ASTAIE (bit 0): Transfer start interrupt request enable flag

<1> When this bit and ASTAIF are set to 1, an interrupt request to vector address 002BH is generated.

3.20.4.7 Clock control register (AIFCLK)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FECA	0000 0000	R/W	AIFCLK	AVCOON	ACMPON	AIFCK1	AIFCK0	AVCOSEL	LRCKPL	BCLKPL	BCKSEL

AVCOON (bit 7): VCO operation control

ACMPON (bit 6): Phase comparator operation control

<1> AVCOON and ACMPON must be set to 1 when the clock generated by the internal PLL is used as the master clock .

<2> Connect an external filter circuit to P33 as shown in Figure 3.20.3.

<3> In this case, set bit 2 (MCLKIN) of AIFMOD (FEC9H), bit 3 (P33DDR) of P3DDR (FE4DH), and bit 3 (P33) of P3 (FE4CH) to 0.

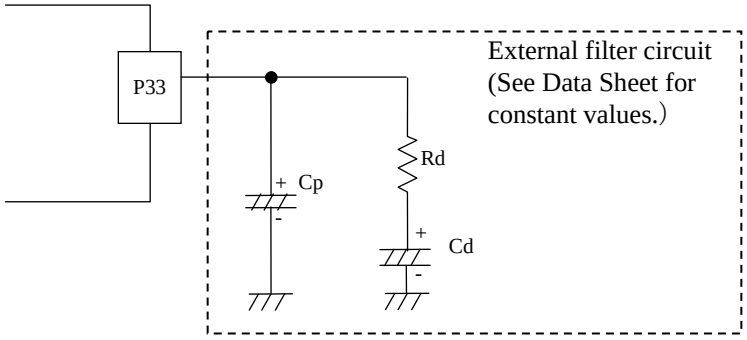


Figure 3.20.3 Internal PLL External Filter Circuit for the Audio Interface

AIFCK1 (bit 5): PLL oscillation frequency select

AIFCK0 (bit 4): PLL oscillation frequency select

AVCOSEL (bit 3): PLL oscillation frequency select

<1> These bits are used to select the frequency of the master clock generated by the internal PLL circuit according to the sampling frequency and master clock/bit clock frequency division ratio (Table 3.20.9).

Table 3.20.11 PLL Oscillation Frequency Settings

Master Clock Frequency	AIFCK[1:0]	AVCOSEL
12.288 MHz	00	0
11.2896 MHz	11	0
16.9344 MHz	01	1
18.432 MHz	10	1
8.192 MHz	11	1

Any setting other than those listed above is inhibited.

LRCKPL (bit 2): LRCK polarity select

<1> When this bit is set to 1, LRCK=L selects the L channel and LRCK=H selects the R channel.

<2> When this bit is set to 0, LRCK=H selects the L channel and LRCK=L selects the R channel.

BCLKPL (bit 1): BCLK polarity select

<1> When this bit is set to 1, data is transmitted on the rising edge of BCLK and taken in on the falling edge of BCLK.

<2> When this bit is set to 0, data is transmitted on the falling edge of BCLK and taken in on the rising edge of BCLK.

BCKSEL (bit 0): BCLK frequency select

<1> A 1 in this bit sets the BCLK frequency to 64 fs.

<2> A 0 in this bit sets the BCLK frequency to 48 fs.

3.20.4.8 Port control register (AIFPC)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FECB	0000 0000	R/W	AIFPC	BCLKLO	LRCKIN	BCLKIN	SDATIN	MCLKDR	LRCKDR	BCLKDR	SDATDR

BCLKLO (bit 7):

This bit must always be set to 0.

LRCKIN (bit 6): LRCK input control

<1> Set this bit to 1 to configure the microcontroller as a slave and to assign LRCK to an external input.

BCLKIN (bit 5): BCLK input control

<1> Set this bit to 1 to configure the microcontroller as a slave and to assign BCLK to an external input.

SDATIN (bit 4): SDAT input control

<1> Set this bit to 1 to assign SDAT to an external input (in data receive mode).

MCLKDR (bit 3): Master clock output control

<1> This bit must be set to 1 after setting bit 6 (PWM0L2) of PWM0L (FE20H) to 1 to transmit the master clock from the PWM0 pin.

<2> In such a case, set bit 2 (ENPWM0) of PWM0C (FE24H) and bit 7 (PWM0L3), bit 5 (PWM0L1), and bit 4 (PWM0L0) of PWM0L (FE20H) to 0.

<3> The PWM0 pin is fixed at the low level when bit 6 (PWM0L2) of PWM0L (FE20H) and this bit are set to 0.

LRCKDR (bit 2): LRCK output control

<1> Set this bit to 1 to configure the microcontroller as the master and to transmit LRCK from the P07 pin.

<2> In such a case, set bit 7 (P07) of P0 (FE40H) to 0. The value of bit 7 (P07DDR) of P0DDR (FE41H) is arbitrary.

BCLKDR (bit 1): BCLK output control

<1> Set this bit to 1 to configure the microcontroller as the master and to transmit BCLK from the P06 pin.

<2> In such a case, set bit 6 (P06) of P0 (FE40H) to 0. The value of bit 6 (P06DDR) of P0DDR (FE41H) is arbitrary.

SDATDR (bit 0): SDAT output control

<1> Set this bit to 1 to transmit SDAT from the P05 pin (in data transmit mode).

<2> In such case, set bit 5 (P05) of P0 (FE40H) to 0. The value of bit 5 (P05DDR) of P0DDR (FE41H) is arbitrary.

3.20.4.9 Transfer count setting register (AIFCNL, AIFCNH)

1) This register defines the number of data bytes to be continuously transmitted (or received).
2) The legitimate value range is from 002[H] to 1FFE[H] (2 to 8190 bytes).
3) When this register is read with AIFRUN set to 0, the contents of this register are read out.
4) When this register is read with AIFRUN set to 1, the number of data bytes that has been transferred is read out.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FECC	0000 0000	R/W	AIFCNL	ACNT7	ACNT6	ACNT5	ACNT4	ACNT3	ACNT2	ACNT1	ACNT0
FECD	HHH0 0000	R/W	AIFCNH	-	-	-	ACNT12	ACNT11	ACNT10	ACNT9	ACNT8

3.20.4.10 Transfer RAM address setting register (AIFADL, AIFADH)

1) This register defines the starting address of the area in RAM for storing the data to be continuously transmitted (or received).
2) The legitimate value range is from 0000[H] to 1FFF[H].

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FECE	0000 0000	R/W	AIFADL	AADR7	AADR6	AADR5	AADR4	AADR3	AADR2	AADR1	AADR0
FECF	HHH0 0000	R/W	AIFADH	-	-	-	AADR12	AADR11	AADR10	AADR9	AADR8

3.20.5 Audio Interface Communication Example

3.20.5.1 Selecting the master clock

1) When configuring the microcontroller as a master and sending BCLK and LRCK, it is necessary to supply the master clock using one of the methods summarized below (see Figure 3.20.2).

Table 3.20.12 Master Clock Source Select

Master Clock Selection Method	Register Settings			
	MCKSEL	MCLKIN	AVCOON	ACMPON
Internal PLL	0	0	1	1
PWM1 external input	X	1	0	0

2) If the internal PLL is selected, select the PLL oscillation frequency (see Table 3.20.11) and the frequency division ratio of the master clock (see Table 3.20.2) according to the required sampling frequency (fs).

Table 3.20.13 PLL Frequency and Division Ratio for a Master Clock Frequency of 384 fs

Sampling Frequency (fs)	PLL Frequency	MCKDV[1:0]	Master Clock Frequency
8 kHz	12.288 MHz	10	3.072 MHz
11.025 kHz	16.9344 MHz		4.2336 MHz
12 kHz	18.432 MHz		4.608 MHz
16 kHz	12.288 MHz	01	6.144 MHz
22.05 kHz	16.9344 MHz		8.4672 MHz
24 kHz	18.432 MHz		9.216 MHz
32 kHz	12.288 MHz	00	12.288 MHz
44.1 kHz	16.9344 MHz		16.9344 MHz
48 kHz	18.432 MHz		18.432 MHz

Either 48 fs or 64 fs can be selected as the bit clock rate.

Table 3.20.14 PLL Frequency and Division Ratio for a Master Clock Frequency of 256 fs

Sampling Frequency (fs)	PLL Frequency	MCKDV[1:0]	Master Clock Frequency
8 kHz	8.192 MHz	10	2.048 MHz
11.025 kHz	11.2896 MHz		2.8224 MHz
12 kHz	12.288 MHz		3.072 MHz
16 kHz	8.192 MHz	01	4.096 MHz
22.05 kHz	11.2896 MHz		5.6448 MHz
24 kHz	12.288 MHz		6.144 MHz
32 kHz	8.192 MHz	00	8.192 MHz
44.1 kHz	11.2896 MHz		11.2896 MHz
48 kHz	12.288 MHz		12.288 MHz

The bit clock rate is fixed at 64 fs.

3) Set up the master clock/bit clock division ratio (Table 3.20.9)

3.20.5.2 Selecting the audio format

1) It is necessary to determine the audio interface format before starting data transfer. It is not possible to change the format in the middle of data transfer.

Table 3.20.15 Audio Format

Register Name	Bit	Bit Name	Description
AIFCLK	2	LRCKPL	Selects the LRCK polarity. 0: L-channel when high, R-channel when low 1: L-channel when low, R-channel when high
	0	BCKSEL	Selects the BCLK frequency. 0: 48 fs (fs: sampling frequency) 1: 64 fs
AIFMOD	7 6	AIFLN1 AIFLN0	Data length select 00: 16 bits 01:18 bits 10: 20 bits 11: 24 bits
	5	AIFFB	0: Left justified 1: Right justified
	3	IISEN	0: Normal mode (left-justified/right -justified) 1: I²S mode
AIFCON	2	AIFDIR	0: LSB first 1: MSB first

2) The figure shown below is the timing relationship of the signals when 64fs, MSB first, 20 bits, and left-justified are selected.

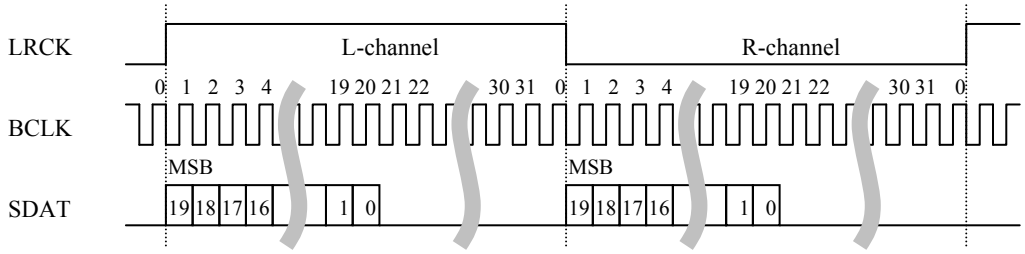


Figure 3.20.4 Audio Interface Format Timing Diagram (64 fs, MSB first, 20 bits, left-justified)

3) The figure shown below is the timing relationship of the signals when 48fs, MSB first, 16 bits, and right-justified are selected.

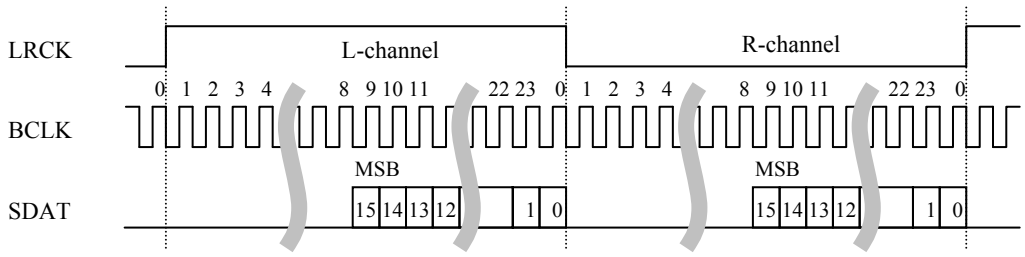


Figure 3.20.5 Audio Interface Format Timing Diagram (48 fs, MSB first, 16 bits, right-justified)

4) The figure shown below is the timing relationship of the signals when 64 fs, 24 bits, and I²S format are selected.

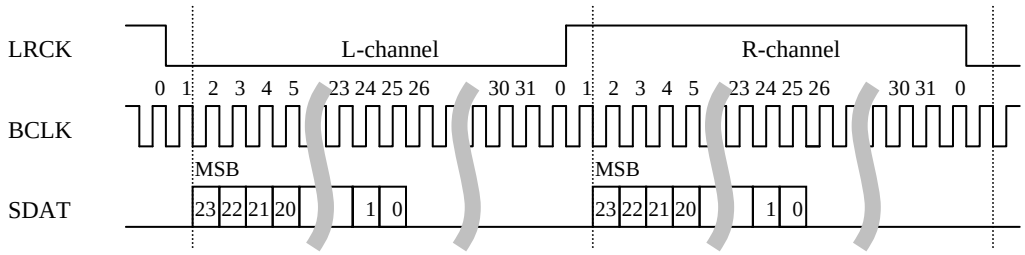


Figure 3.20.6 Audio Interface Format Timing Diagram (64 fs, 24 bits, I²S format)

3.20.5.3 Setting up I/O ports

1) Select the I/O direction of the relevant ports.
2) Select the pin to be used for receiving serial data when performing receive processing.

Table 3.20.16 Port I/O Settings

Register Name	Bit	Bit Name	Settings
AIFPC	6	LRCKIN	LR clock input control 0: Not input 1: Input
	5	BCLKIN	Bit clock input control 0: Not input 1: Input
	4	SDATIN	Serial data input control 0: Not input 1: Input
	3	MCLKDR	Master clock output control 0: Not output 1: Output (Set PWM0L to 40H.)
	2	LRCKDR	LR clock output control 0: Not output 1: Output (Set P07 to 0.)
	1	BCLKDR	Bit clock output control 0: Not output 1: Output (Set P06 to 0.)
	0	SDATDR	Serial data output control 0: Not output 1: Output (Set P05 to 0.)
AIFMOD	4	SDATISEL	Serial data input pin select 0: P05 pin 1: P00 pin

3.20.5.4 Audio interface transmit/receive processing

Example 1.

Below is an example of playing back (recording) audio data for the period equivalent to 65 samples with a bit clock frequency of 64 fs, a bit length of 20 bits, MSB first, left-justified, and a sampling frequency of 32 kHz. It is assumed that the first block of audio data consisting of 32 samples (3 bytes × 2 channels × 32=192 bytes) is stored in the data RAM area with a starting address of 0400H, and that the remaining 33 samples (3 bytes × 2 channels × 33=198 bytes) are stored in the data RAM area with a starting address of 0500H (store when recording).

<1> Setting up the master clock
- When configuring the microcontroller as a master and sending BCLK and LRCK, it is necessary to supply the master clock either from an external clock source or from the internal PLL (see 3.20.5.1).
- It is not necessary to set up the master clock when configuring the microcontroller as a slave and receiving BCLK and LRCK from an external device.

<2> Setting up the audio interface format
- Select the audio interface format before starting transfer (see 3.20.5.2).

<3> Setting up ports
- Set up the relevant ports according to whether the audio interface is to be configured as a master or slave or whether transmission or reception processing is to be performed (see 3.20.5.3).

<4> Generating LRCK and BCLK
- Set ACLKON to start the generation of LRCK and BCLK.
- Set ACLKON also when receiving LRCK and BCLK from an external device.

<5> Switching between transmit and receive modes
- Set AIFWRT to 0 in transmit (playback) mode.
- Set AIFWRT to 1 in receive (recording) mode.
- Set AIFWRT to 1 to perform transmit and receive processing (playback and recording) simultaneously.

<6> Setting the transfer byte count and transfer RAM address
- Set the values for the first data area (0400H).

  AIFCNH=00H, AIFCNL=C0H
  AIFADH=04H, AIFADL=00H

<7> Starting transfer processing
- Set ASTAIE (to enable an interrupt to be generated at the beginning of transfer).
- Set AIFRUN. (When LRCK and BCLK are being received from an external device, set AIFRUN after completion of 2 or more LRCK cycles after setting ACLKON to 1.)

<8> First transfer start interrupt
- ASTAIF is set and an interrupt is generated when the first data from the first data area is transferred to the transmit buffer (when data from the receive buffer is transferred to RAM in the recording mode). At this moment, since the values of the transfer count setting register and transfer RAM address setting register are saved in internal registers, it is possible to change their settings.
- Setting up the values for the second data area (0500H)

  AIFCNH=00H, AIFCNL=C6H
  AIFADH=05H, AIFADL=00H

- Clear ASTAIF.

<9> Second transfer start interrupt
- ASTAIF is set and an interrupt is generated when the first data from the second data area is transferred to the transmit buffer (when data from the receive buffer is transferred to RAM in the recording mode).
- Clear ASTAIF.
- Set AIFOFF (to have AIFRUN automatically cleared at the end of data transfer).
- Set AENDIE (to enable an interrupt to be generated at the end of data transfer).

<10> Transfer end interrupt

- AENDIF is set and an interrupt is generated when the last data byte (198th byte) from the second data area is transferred to the transmit buffer (when data from the receive buffer is transferred to RAM in the recording mode).
- Clear AENDIF.

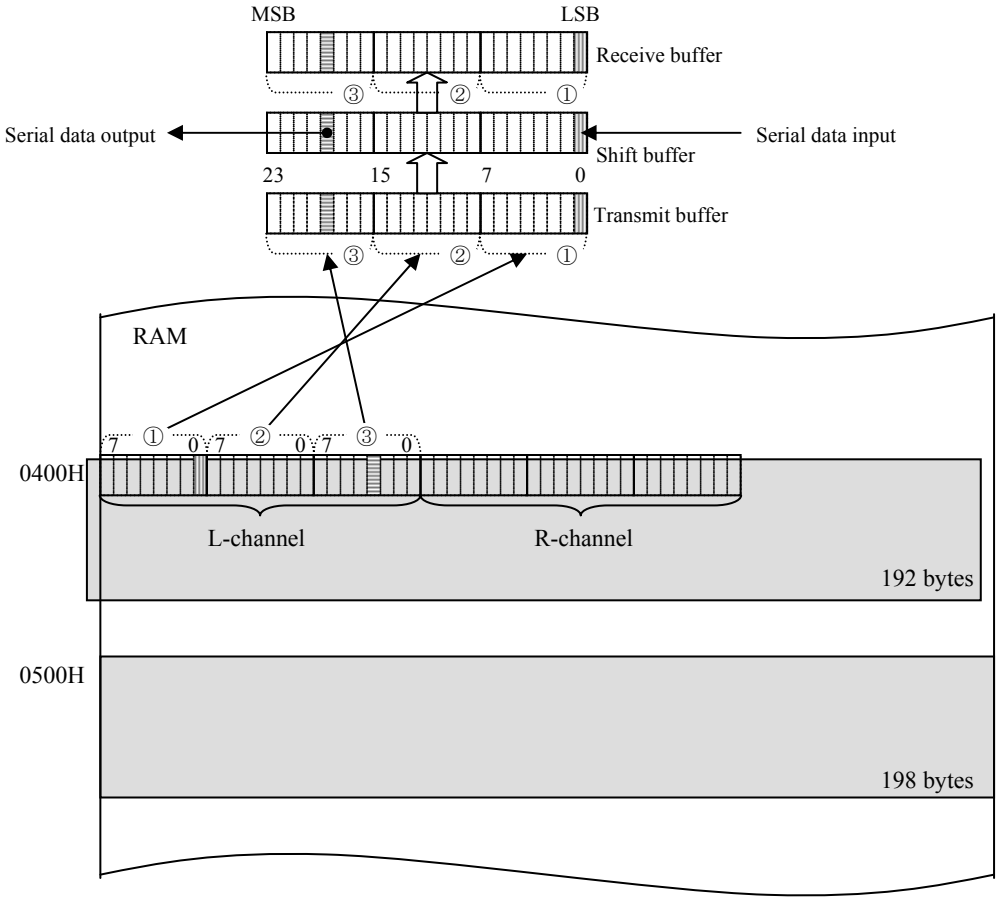


Figure 3.20.7 Audio Interface Transfer Processing

3.20.6 Synchronous Serial Interface Communication Example

3.20.6.1 Master mode operation

1) Synchronous clock/transfer mode
   - Select the system clock as the master clock (see Figure 3.20.2).
   - Set up RATESEL[1:0] and select the frequency of the synchronous clock (a system clock frequency division ratio of 1/2 to 1/8).
   - For the bit length, it is possible to set AIFLN[1:0] to 00 (16 bits) or 11 (24 bits). To enable the parity check function, set AIFLN[1:0] to 00 (16 bits).

Table 3.20.17 Master Mode Clock/Transfer Mode Settings

Register Name	Bit	Bit Name	Settings
AIFRPG	6 5	MCKDV1 MCKDV0	00 (No frequency division)
	4	FMTCEN	0: No LR format check 1: LR format check
	3	PRTYEN	0: No parity check 1: Parity check
AIFSTC	6	MCKSEL	1 (System clock)
	5	ASTREN	1 (Master mode)
	4	ASTRSTP	0 (Disable software-controlled suspension)
	3	ASTPCEN	Continuous transfer suspension control 0: Disabled 1: Enabled
	2	ASTPCHI	Continuous transfer suspension polarity 0: Data request when high 1: Data request when low
	1 0	ASTPSL1 ASTPSL0	Continuous transfer suspension control pin 00: P70 01: P71 10: P72 11: P73
AIFRAT	6	MCLKPL	0
	5	BCLKSL	0
	4	AIFSWP	Switching between high-order bit and low-order bit 0: No 1: Yes
	3	AFLTSL1	0
	2	AFLTSL0	0
	1 0	RATESEL1 RATESEL0	Synchronous clock frequency 00: 1/2 of system clock 01: 1/4 of system clock 1X: 1/8 of system clock
AIFMOD	7 6	AIFLN1 AIFLN0	00: 16 bits 11: 24 bits
	5	AIFFB	0: Left justified 1: Right justified
	4	SDATISEL	0
	3	IISEN	0 (Left-/right-justified format select)
	2	MCLKIN	0 (Internal clock select)

Register Name	Bit	Bit Name	Settings
AIFCLK	7	AVCOON	0
	6	ACMPON	0
	5	AIFCK1	0
	4	AIFCK0	0
	3	AVCOSEL	0
	2	LRCKPL	0
	1	BCLKPL	0
	0	BCKSEL	0 (48 fs)

2) Setting up ports
   - It is possible to run the interface without generating LRCK.
   - Set the data latch value for the output pins (P05, P06, and P07) to 0.
   - The values of P05DDR, P06DDR, and P07DDR are arbitrary.

Table 3.20.18 Master Mode Port Settings

Register Name	Bit	Bit Name	Settings
AIFPC	7	BCLKLO	0
	6	LRCKIN	0
	5	BCLKIN	0
	4	SDATIN	0
	3	MCLKDR	0
	2	LRCKDR	0: No output 1: Output
	1	BCLKDR	1 (Output)
	0	SDATDR	1 (Output)

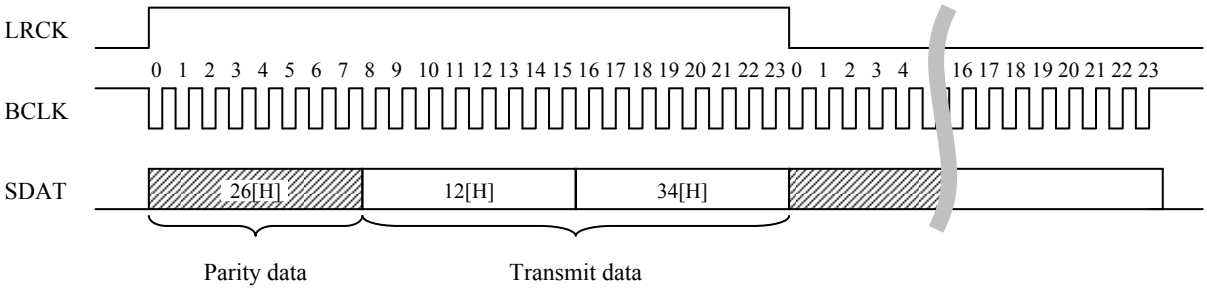
3) Setting the transfer count and transfer RAM address
   - Set (2 × natural number) as the transfer byte count (2 to 8190) when a bit length of 16 bits is selected.
   - Set (3 × natural number) as the transfer count (3 to 8190) when a bit length of 24 bits is selected.

4) Starting transfer processing

Table 3.20.19 Master Mode Transfer Startup Setting

Register Name	Bit	Bit Name	Settings
AIFCON	6	ACLKON	1
	5	AIFRUN	1
	4	AIFOFF	1 (AIFRUN automatically cleared at end of transmission)
	3	AIFWRT	0 (Transmission only)
	2	AIFDIR	0: LSB first 1: MSB first
	1	AENDIF	0
	0	AENDIE	1 (Interrupt requests enabled)

5) End of transfer processing
   - AIFRUN is set to 0 and AENDIF is set to 1, and an interrupt is generated when the transfer of the specified number of data bytes between RAM and the transmit buffer is completed.
   - The data transmission completes in 1 LRCK cycle after AENDIF is set to 1.
   - After the data transfer is completed, set ACLKON to 0 and initialize the internal circuit.



(48 fs, 16 bits, right-justified, with parity check)
Figure 3.20.8 Master Mode Transfer Timing

3.20.6.2 Slave mode operation

1) Transfer mode
   - For the bit length, it is possible to set AIFLN[1:0] to 00 (16 bits) or 11 (24 bits). To enable the parity check function, set AIFLN[1:0] to 00 (16 bits).
   - When AIFLN[1:0] is set to 00 (16 bits) and no parity check is to be used, set AIFFB to 0 (left-justified).
   - Set LRCKPL to 0 if there is LRCK input during receive processing or if LRCK is fixed at the high level. If LRCK is fixed at the low level, set LRCKPL to 1.

Table 2.20.20 Slave Mode Transfer Mode Settings

Register Name	Bit	Bit Name	Settings
AIFRPG	4	FMTCEN	LR format check 0: No 1: Yes
	3	PRTYEN	Parity check 0: No 1: Yes
AIFSTC	7	ARUNOE	AIFRUN flag pin output 0: No 1: Yes
	6	MCKSEL	0
	5	ASTREN	0 (Slave mode)
	4	ASTRSTP	0
	3	ASTPCEN	0
	2	ASTPCHI	AIFRUN flag output polarity 0: Not inverted 1: Inverted
	1 0	ASTPSL1 ASTPSL0	AIFRUN flag output pin 00: P23 01: P71 10: P72 11: P73

Register Name	Bit	Bit Name	Settings
AIFRAT	6	MCLKPL	0
	5	BCLKSL	0
	4	AIFSWP	Switching between high-order bit and low-order bit 0: No 1: Yes
	3	AFLTSL1	0
	2	AFLTSL0	0
	1	RATESEL1	0
	0	RATESEL0	0
AIFMOD	7 6	AIFLN1 AIFLN0	00: 16 bits 11: 24 bits
	5	AIFFB	0: Left justified 1: Right justified
	4	SDATISEL	Serial data input pin 0: P05 pin 1: P00 pin
	3	IISEN	0 (Left-/right-justified format select)
	2	MCLKIN	0
AIFCLK	7	AVCOON	0
	6	ACMPON	0
	5	AIFCK1	0
	4	AIFCK0	0
	3	AVCOSEL	0
	2	LRCKPL	0: LRCK input present or fixed at high level 1: LRCK fixed at low level
	1	BCLKPL	0
	0	BCKSEL	0 (48 fs)

2) Setting up ports
   - Configure all of LRCK, BCLK, and SDAT for input

Table 3.20.21 Slave Mode Port Settings

Register Name	Bit	Bit Name	Settings
AIFPC	7	BCLKLO	0
	6	LRCKIN	1
	5	BCLKIN	1
	4	SDATIN	1
	3	MCLKDR	0
	2	LRCKDR	0
	1	BCLKDR	0
	0	SDATDR	0

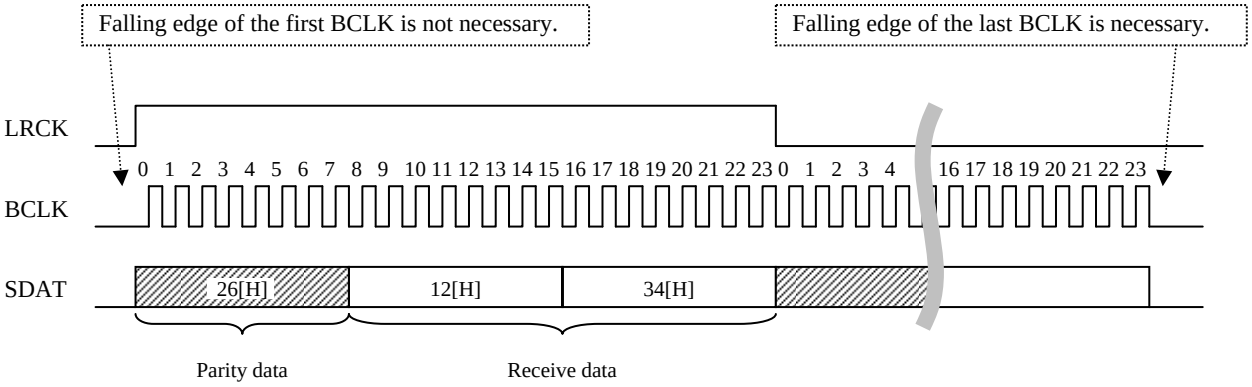
3) Setting the transfer count and transfer RAM address
   - Set (2 × natural number) as the transfer count (2 to 8190) if a bit length of 16 bits is selected.
   - Set (3 × natural number) as the transfer count (3 to 8190) when a bit length of 24 bits is selected.

4) Starting transfer processing

Table 3.20.22 Slave Mode Transfer Startup Settings

Register Name	Bit	Bit Name	Settings
AIFCON	6	ACLKON	1
	5	AIFRUN	1
	4	AIFOFF	1 (AIFRUN automatically cleared at end of transmission)
	3	AIFWRT	1 (Reception)
	2	AIFDIR	0: LSB first 1: MSB first
	1	AENDIF	0
	0	AENDIE	1 (Interrupt requests enabled)

5) End of transfer processing
- AIFRUN is set to 0 and AENDIF is set to 1, and an interrupt is generated when the transfer of the specified number of data bytes between RAM and the receive buffer is completed.
- After the data transfer is completed, set ACLKON to 0 and initialize the internal circuit.



(48 fs, 16 bits, right-justified, with parity check)
Figure 3.20.9 Slave Mode Transfer Timing

3.20.7 Audio Interface HALT Mode Operation

1) Data transmission and reception cannot be performed normally in HALT mode since the automatic data transfer between RAM and transmit buffer and between RAM and receive buffer is suspended once HALT mode is entered. Make sure that the microcontroller enters HALT mode after stopping the audio data transmission or reception processing by clearing AIFRUN.

2) Since the entry into HALT mode suspends the operation of the audio interface, it is impossible to exit HALT mode on an interrupt generated from the audio interface.

3.21 USB Host

3.21.1 Overview

This series of microcontrollers incorporates a USB (Universal Serial Bus) host controller circuit that has the following features:

1) Supports full-speed (FS) (12 Mbps) and low-speed (LS) (1.5 Mbps) transfers.
2) Supports control transfer, bulk transfer, interrupt transfer, and isochronous transfer modes.
3) Up to 1023-byte packet length
4) Can accommodate two USB downstream ports.

3.21.2 Functions

1) USB host control
   - Automatically performs transaction processing, SOF transmission processing, and device connection/disconnection detection processing.
   - The transmit/receive buffer (1023 bytes maximum) is mapped onto RAM.
   - The USB data sampling clock (48 MHz) is supplied from a clock that is generated by the internal PLL circuit.
2) Interrupt generation

   An interrupt request is generated on detection of a device connection, device disconnection, a resume signal, or the end of an SOF packet transmission or USB transaction if the corresponding interrupt request enable bit is set.
3) It is necessary to manipulate the following special function registers to control the USB host controller.
   - USBDIV, PLLCNT
   - UHCCTR, UHCPRT, UHCINT, TRSINT, UHFRML, UHFRMH, DEVADR, DEVEPN, BUSCTR, UHCSTA, UHCCNT, UHCRX, UHCADL, UHCADH, BACINT, UHTST0, UHTST1, BPTCTR, BPTINT
   - P2, P2DDR
   - P3, P3DDR

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE04	0000 0000	R/W	USBDIV	USBDIV7	DVCKON	DVCKDR	USBDIV4	USBDIV3	UDVSEL2	UDVSEL1	UDVSEL0
FE0D	0000 0000	R/W	PLLCNT	SELREF2	SELREF1	SELREF0	PLLTEST	VCOSTP	CMPSTP	PLLCNT1	PLLCNT0
FEB6	0000 0000	R/W	UHCCTR	UHCON	UHCRUN	NKRI1	NKRI0	DRIRST	FTSHRT	UHPIEZ	UHMIEZ
FEB7	0000 0000	R/W	UHCPRT	FTCKOE	UHPSEL1	UHPSEL0	UHSTBY	UHDRP	UHDRM	UHDTP	UHDTM
FEB8	0000 0000	R/W	UHCINT	DTCIF	DTCIE	RSMIF	RSMIE	SOFIF	SOFIE	ATCIF	ATCIE
FEB9	0000 0000	R/W	TRSINT	ACKIF	ACKIE	NAKIF	NAKIE	ERRIF	ERRIE	STLIF	STLIE
FEBA	0000 0000	R/W	UHFRML	HFRM7	HFRM6	HFRM5	HFRM4	HFRM3	HFRM2	HFRM1	HFRM0
FEBB	0000 0000	R/W	UHFRMH	HFRM15	HFRM14	HFRM13	HFRM12	HFRM11	HFRM10	HFRM9	HFRM8
FEBC	H000 0000	R/W	DEVADR	-	DADR6	DADR5	DADR4	DADR3	DADR2	DADR1	DADR0
FEBD	HHHH 0000	R/W	DEVEPN	-	-	-	-	DEPN3	DEPN2	DEPN1	DEPN0
FEBE	0000 0000	R/W	BUSCTR	RSTRUN	RSMRUN	SOFRUN	NKRTRY	ERRTRY	ALS	CBS1	CBS0
FEBF	0000 0000	R/W	UHCSTA	TRSRUN	TRSTGL	TRSOVR	TGLERR	TTKN1	TTKN0	TRSISO	TGLMON
FEC0	0000 0000	R/W	UHCCNT	UHCCN7	UHCCN6	UHCCN5	UHCCN4	UHCCN3	UHCCN2	UHCCN1	UHCCN0
FEC1	0000 0000	R/W	UHCRX	UHCRX7	UHCRX6	UHCRX5	UHCRX4	UHCRX3	UHCRX2	UHCRX1	UHCRX0
FEC2	0000 00HH	R/W	UHCADL	UHAD7	UHAD6	UHAD5	UHAD4	UHAD3	UHAD2	-	-
FEC3	HHH0 0000	R/W	UHCADH	-	-	-	UHAD12	UHAD11	UHAD10	UHAD9	UHAD8
FEC4	0000 0000	R/W	BACINT	UHCRX9	UHCRX8	UHCCN9	UHCCN8	BACBIF	BACBIE	BACIF	BACIE
FEC5	0000 0000	R/W	UHTST0	UHTST	UHCMPT	UHCMPK	VD3KIL	R2RAM	UHTAD2	UHTAD1	UHTAD0
FEC6	0000 0000	R/W	UHTST1	UHTDT7	UHTDT6	UHTDT5	UHTDT4	UHTDT3	UHTDT2	UHTDT1	UHTDT0
FEC7	HHHH 0000	R/W	BPTCTR	-	-	-	-	BLS	BPTRST	BPTRSM	BPTSOF
FEC8	0000 0000	R/W	BPTINT	BDTCIF	BDTCIE	BRSMIF	BRSMIE	BATCIF	BATCIE	BCBS1	BCBS0
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

3.21.3 Circuit Configuration

The USB host controller circuit consists of the following functional blocks:

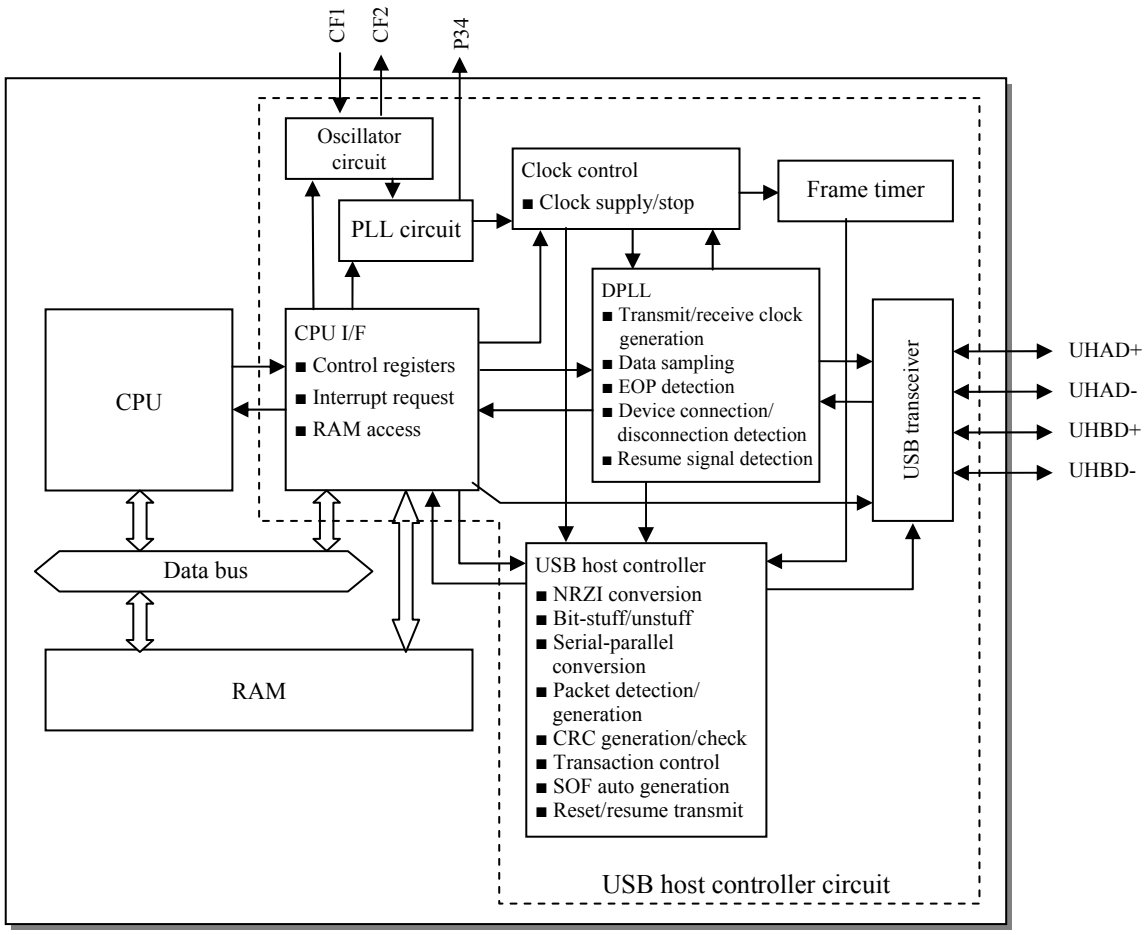


Figure 3.21.1 USB Host Controller Circuit Block Diagram

3.21.3.1 USB interface PLL circuit

The internal PLL generates the USB data sampling clock (48 MHz).

3.21.3.2 Clock control circuit

This circuit starts and stops supply of the clock signal to the DPLL, USB host controller, and frame timer.

3.21.3.3 Frame timer

The frame timer controls the transmission of SOF packets at the frame interval (1 ms).

3.21.3.4 DPLL circuit

1) Generates the transmit/receive clock (12 MHz).
2) Samples receive data.
3) Detects EOP.
4) Detects device connection and disconnection.
5) Detects the resume signal.

3.21.3.5 USB host controller circuit

1) Performs NRZI encoding/decoding.
2) Performs bit stuffing/unstuffing.
3) Performs serial-to-parallel conversion.
4) Detects and generates packets.
5) Generates and checks CRCs.
6) Controls transactions.
7) Transmits bus reset and resume signals.
8) Automatically generates SOF packets.

3.21.3.6 CPU interface circuit

1) Has registers for controlling the operation of the USB host controller circuit.
2) Issues 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 (endpoint buffer).

3.21.3.7 Related I/O pins

1) The table below lists the I/O pins that are related to the USB host controller circuit.

Table 3.21.1 Related I/O Pins

Pin Name	I/O	Description
UHAD+	I/O	USB downstream A port
UHAD-	I/O	USB downstream A port
UHBD+	I/O	USB downstream B port
UHBD-	I/O	USB downstream B port
P34	I/O	Filter circuit connection pin for the internal PLL

2) The USB downstream port peripheral circuit is shown below.
3) Connect the pull-down resistors (15kΩ) for the UHAD+, UHAD-, UHBD+, and UHBD- pins to GND.

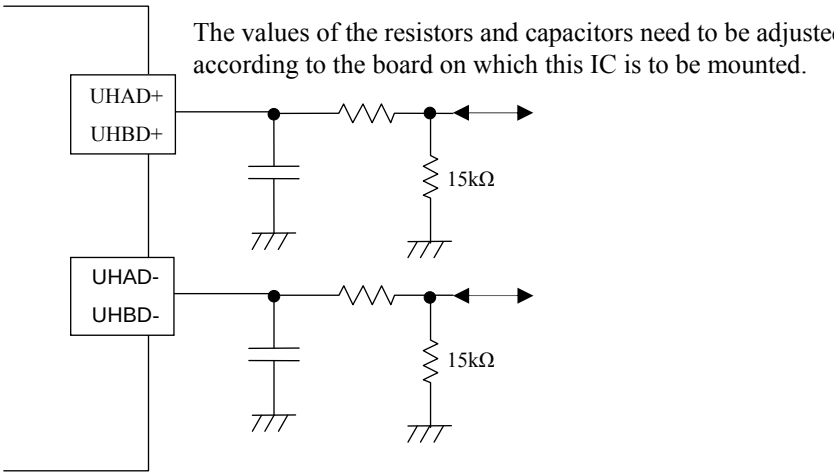


Figure 3.21.2 USB Downstream Port Peripheral Circuit

4) To generate the 48 MHz USB clock using the internal PLL circuit, it is necessary to connect the following external filter circuit to the P34/UFILT pin.

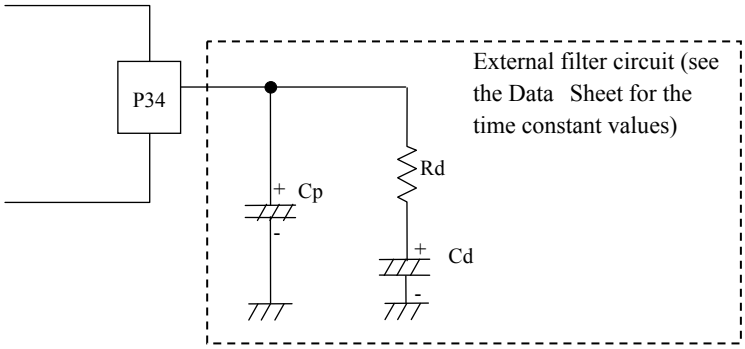


Figure 3.21.3 Internal PLL External Filter Circuit for the USB Interface

3.21.4 Related Registers

3.21.4.1 USB frequency-divided clock control register (USBDIV)

1) This register is used to select the frequency of the frequency-divided clock that is generated from the 48 MHz clock for USB.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE04	0000 0000	R/W	USBDIV	USBDIV7	DVCKON	DVCKDR	USBDIV4	USBDIV3	UDVSEL2	UDVSEL1	UDVSEL0

USBDIV7 (bit 7): Reserved bit.

This bit must always be set to 0.

DVCKON (bit 6): High-speed clock select

<1> A 1 in this bit selects the USB frequency-divided clock that is generated by frequency-dividing the 48 MHz clock for USB as the high-speed clock.

<2> A 0 in this bit selects the main clock (CF) as the high-speed clock.

DVCKDR (bit 5): USB frequency-divided clock external output control

<1> This bit must be set to 1 to transmit the USB frequency-divided clock that is generated by frequency-dividing the 48 MHz clock for USB from 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 configuration, the USB frequency-divided clock is transmitted from P73.

<2> Set this bit to 0 if the USB frequency-divided clock is not transmitted to any external device.

USBDIV4 to USBDIV3 (bits 4, 3): Reserved bit.

This bit must always be set to 0.

UDVSEL2 to UDVSEL0 (bits 2 to 0): Frequency-divided clock frequency select

<1> These bits select the frequency of the divided clock of the 48 MHz clock for USB.

<2> Set the frequency divided clock to 8 to 12 MHz to run the USB interface controller circuit by supplying the USB frequency-divided clock as the system clock.

<3> To change the USB frequency-divided clock frequency from any value other than its initial value (UDVSEL＝000), temporarily set it to the "frequency-divided clock stopped" 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 3.21.2 USB Frequency-divided Clock Frequency Select

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 (inhibited)

3.21.4.2 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 to SELREF0 (bits 7 to 5): PLL reference clock frequency select

<1> These bits select the oscillation frequency of the main clock connected to the CF pins (CF1 and CF2).

<2> Either 8 MHz or 12 MHz can be selected if “ENABLE” is selected in the user option "Main clock 8 MHz selection." If “DISABLE” is selected, only 12 MHz is selected.

Tale 3.21.3 Main Clock Oscillation Frequencies

Main clock 8MHz selection	SELREF[2: 0]	Main Clock Oscillation Frequency
ENABLE	000	8 MHz
	100	12 MHz
DISABLE	000	12 MHz
	100	12 MHz

* Any other settings than those are inhibited.

PLLTEST (bit 4): PLL test bit

This bit must always be set to 0.

VCOSTP (bit 3): PLLVCO operation control

CMPSTP (bit 2): PLL phase comparator operation control

<1> Set both VCOSTP and CMPSTP to 0 when generating the 48 MHz clock for USB using the internal PLL. In this case, set bit 4 (P34DDR) of P3DDR (FE4DH) and bit 4 (P34) of P3 (FE4CH) to 0. In addition, it is necessary to connect the external filter circuit shown in Figure 3.21.3 to the P34/UFILT pin.

<2> Set these bits to 1 when the PLL is not to be used.

PLLCNT1 to PLLCNT0 (bits 1, 0): Reserved bits

These bits must always be set to 0.

3.21.4.3 USB host operation control register (UHCCTR)

1) This register is used to control clock supply/stop.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEB6	0000 0000	R/W	UHCCTR	UHCON	UHCRUN	NKRI1	NKRI0	DRIRST	FTSHRT	UHPIEZ	UHMIEZ

UHCON (bit 7): USB host clock control

UHCRUN (bit 6): USB host clock control

<1> These bits control clock supply/stop.

Table 3.21.4 Clock Control

UHCON	UHCRUN	USB Host Operation
0	-	Enable only to detect the change in the USB bus state (bus active). Disable the other USB host functions.
1	0	Enable only to detect the device connection/disconnection, and resume signals. Disable the other USB host functions.
1	1	Enable the entire USB host block.

NKRI1 to NKRI0 (bits 5, 4): NAK automatic retry interval control

<1> These bits define the retry interval when the NAK automatic retry function is enabled (NKRTRY =1).

Table 3.21.5 NAK Automatic Retry Interval Control

NKRI[1: 0]	Retry Interval
00	Approx. 11 us
01	Approx. 43 us
10	Approx. 171 us
11	Retry is carried over to the next frame.

DRIRST (bit 3): Reserved bit

This bit must always be set to 0.

FTSHRT (bit 2): Reserved bit

This bit must always be set to 0.

UHPIEZ (bit 1): UHAD+ pin input enable flag

<1> A 1 in this bit disables reading in of the data on the UHAD+ pin.

<2> A 0 in this bit enables reading in of the data on the UHAD+ pin.

<3> Set this bit to 0 to perform USB communication.

UHMIEZ (bit 0): UHAD- pin input enable flag

<1> A 1 in this bit disables reading in of the data on the UHAD- pin.

<2> A 0 in this bit enables reading in of the data on the UHAD- pin.

<3> Set this bit to 0 to perform USB communication.

3.21.4.4 USB port control register (UHCPRT)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEB7	0000 0000	R/W	UHCPRT	FTCKOE	UHPSEL1	UHPSEL0	UHSTBY	UHDRP	UHDRM	UHDTP	UHDTM

FTCKOE (bit 7): 1 ms pulse output control

<1> Setting this bit to 1, bit 6 (P72DDR) of P7 (FE5CH) to 1, and bit 2 (P72) of P7 (FE5CH) to 0 enables pulses to be transmitted from P72 at 1 ms intervals (approx. 80 ns) when the frame timer is active (SOFRUN=1 or BPTSOF=1).

UHPSEL1 (bit 6): Reserved bit

This bit must always be set to 0.

UHPSEL0 (bit 5): Transaction port select

<1> This bit selects the USB downstream port (A port/B port) to be used for transaction processing. Transaction processing is carried out only on the selected port.

<2> A port/B port can be controlled separately for bus reset transmission, resume transmission, and SOF automatic transmission regardless of whether they are selected or not selected.

<3> Device connection and disconnection, and resume reception can be detected at A port/B port separately regardless of whether they are selected or not selected.

Table 3.21.6 USB Port Selection

UHPSEL0	USB Port Selection
0	A port selected
1	B port selected

UHSTBY (bit 4): USB transceiver operation control

<1> When this bit is set to 1, the USB transceiver is placed in the suspend state. It is possible to detect the change in the USB bus state even in this state.

<2> When this bit is set to 0, the USB transceiver is in the normal operating mode.

UHDRP (bit 3): UHAD+ pin general-purpose output control

<1> When this bit is set to 1, the UHDTP data is transmitted from the UHAD+ pin.

<2> This bit must be set to 0 when performing USB communication.

UHDRM (bit 2): UHAD- pin general-purpose output control

<1> When this bit is set to 1, the UHDTM data is transmitted from the UHAD- pin.

<2> This bit must be set to 0 when performing USB communication.

UHDTP (bit 1): UHAD+ port data latch

<1> When UHDRP is set to 1, data in this bit is transmitted from the UHAD+ pin.

<2> When this bit is read with an instruction, the data on the UHAD+ pin is read in. If UHCPRT is manipulated using a NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced instead of the data on the UHAD+ pin.

UHDTM (bit 0): UHAD- port data latch

<1> When UHDRM is set to 1, data in this bit is transmitted from the UHAD- pin.

<2> When this bit is read with an instruction, the data on the UHAD- pin is read in. If UHCPRT is manipulated using a NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, the contents of the register are referenced instead of the data on the UHAD- pin.

3.21.4.5 USB host interrupt control register (UHCINT)

1) This register controls the USB host interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEB8	0000 0000	R/W	UHCINT	DTCIF	DTCIE	RSMIF	RSMIE	SOFIF	SOFIE	ATCIF	ATCIE

DTCIF (bit 7): A port device disconnection detection flag

<1> This flag is set when the SE0 state (UHAD+ = L, UHAD- = L) continues for 2.5 µs or more at the USB downstream A port.

<2> This flag remains set until the above-mentioned bus state is reset.

<3> This flag must be cleared with an instruction.

DTCIE (bit 6): A port device disconnection detection interrupt request enable control

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

RSMIF (bit 5): A port FS resume/LS device connection detection flag

<1> This flag is set when the bus state (UHAD+ = L and UHAD- = H) continues for 2.5 µs or more at the USB downstream A port.

<2> This flag remains set until the above-mentioned bus state is reset.

<3> This flag must be cleared with an instruction.

RSMIE (bit 4): A port FS resume/LS device connection detection interrupt request enable control

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

SOFIF (bit 3): SOF packet transmission end flag

<1> This flag is set when the transmission of an SOF packet ends.

<2> This flag must be cleared with an instruction.

SOFIE (bit 2): SOF packet transmission end interrupt request enable control

<1> When this bit and SOFIF are set to 1, an interrupt request to vector address 004BH is generated.

ATCIF (bit 1): A port FS device connection/LS resume detection flag

<1> This flag is set when bus state (UHAD+ = H and UHAD- = L) continues for 2.5 µs or more at the USB downstream A port.

<2> This flag remains set until the above-mentioned bus state is reset.

<3> This flag must be cleared with an instruction.

ATCIE (bit 0): A port FS device connection/LS resume detection interrupt request enable control

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

3.21.4.6 Transaction interrupt control register (TRSINT)

1) This register controls the transaction interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEB9	0000 0000	R/W	TRSINT	ACKIF	ACKIE	NAKIF	NAKIE	ERRIF	ERRIE	STLIF	STLIE

ACKIF (bit 7): ACK end flag

<1> This flag is set to 1 when the transaction ends with an ACK. This bit is not set, however, if a data toggle mismatch is found in an IN transaction when TGLMON (UHCSTA, bit 0) is set to 0.

<2> This flag must be cleared with an instruction.

ACKIE (bit 6): ACK interrupt request enable control

<1> When this bit and ACKIF are set to 1, an interrupt request to vector address 0043H is generated.

NAKIF (bit 5): NAK end flag

<1> This bit is set to 1 if a transaction ends with a NAK when the NAK automatic retry function is disabled (NKRTRY=0). When NKRTRY=1, this flag is not set even if the transaction ends with a NAK.

<2> This flag must be cleared with an instruction.

NAKIE (bit 4): NAK interrupt request enable control

<1> When this bit and NAKIF are set to 1, an interrupt request to vector address 0043H is generated.

ERRIF (bit 3): Error end flag

<1> This bit is set to 1 if a transaction ends with an error when the automatic error retry function (ERRTRY=0) is disabled. When ERRTRY=1, this flag is not automatically set until two transaction errors occur in succession, and is automatically set if three transaction errors occur in succession.

<2> This flag must be cleared with an instruction.

ERRIE (bit 2): Error interrupt request enable control

<1> When this bit and ERRIF are set to 1, an interrupt request to vector address 0043H is generated.

STLIF (bit 1): STALL end flag

<1> This flag is set to 1 when a transaction ends with a stall.

<2> This flag must be cleared with an instruction.

STLIE (bit 0): STALL interrupt request enable control

<1> When this bit and STLIF are set to 1, an interrupt request to vector address 0043H is generated.

3.21.4.7 Frame number register (UHFRML, UHFRMH)

1) This register stores the frame number.
2) The value of bits 10 to 0 of HFRM is transmitted as the frame number of the SOF packet.
3) This register counts up every 1 ms if bit 5 (SOFRUN) of BUSCTR or bit 0 (BPTSOF) of BPTCTR is set to 1.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEBA	0000 0000	R/W	UHFRML	HFRM7	HFRM6	HFRM5	HFRM4	HFRM3	HFRM2	HFRM1	HFRM0
FEBB	0000 0000	R/W	UHFRMH	HFRM15	HFRM14	HFRM13	HFRM12	HFRM11	HFRM10	HFRM9	HFRM8

3.21.4.8 Device address setting register (DEVADR)

1) This register sets the USB device address.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEBC	H000 0000	R/W	DEVADR	-	DADR6	DADR5	DADR4	DADR3	DADR2	DADR1	DADR0

3.21.4.9 Endpoint setting register (DEVEPN)

1) This register sets the endpoint number.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEBD	HHHH 0000	R/W	DEVEPN	-	-	-	-	DEPN3	DEPN2	DEPN1	DEPN0

3.21.4.10 USB bus control register (BUSCTR)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEBE	0000 0000	R/W	BUSCTR	RSTRUN	RSMRUN	SOFRUN	NKRTRY	ERRTRY	ALS	CBS1	CBS0

RSTRUN (bit 7): A port bus reset transmission control

<1> When this bit is set to 1, a bus reset signal (SE0) is transmitted to the USB downstream A port.

<2> The bus reset signal continues to be transmitted until this bit is cleared with an instruction.

RSMRUN (bit 6): A port resume transmission control

<1> When this bit is set to 1, a resume signal (K state) is transmitted to the USB downstream A port.

<2> The resume signal continues to be transmitted until this bit is cleared with an instruction.

<3> When this bit is cleared, the USB host stops transmission of the K state signal and transmits a low-speed EOP at the end.

SOFRUN (bit 5): A port SOF transmission control

<1> When this bit is set to 1, the frame timer is started, which automatically transmits an SOF packet to the USB downstream A port every 1 ms.

<2> No SOF packet is transmitted to A port even when this bit is set to 1 if a bus reset or resume signal is transmitted to A port at the same timing as when an SOF packet is transmitted.

<3> When this bit is set to 1, the contents of frame number registers (UHFRML and UHFRMH) are counted up every 1 ms regardless of whether SOF packets are transmitted or not.

<4> Clear this bit with an instruction to stop automatic transmission of SOF packets.

NKRTRY (bit 4): NAK automatic retry control

<1> If a transaction ends with a NAK when this bit is set to 1, TRSRUN is not automatically cleared and the USB host controller automatically retries the transaction repeatedly until the transaction ends with a response other than NAK (ACK response, STALL response, or error).

<2> The interval at which NAK automatic retries are performed is determined by bit 5 (NKRI1) and bit 4 (NKRI0) of UHCCTR (see Table 3.21.5, “NAK automatic interval control).”

<3> If a transaction ends with a NAK when this bit is set to 0, TRSRUN is automatically cleared and no automatic transaction retry is performed.

ERRTRY (bit 3): Error automatic retry control

<1> If a transaction error occurs when this bit is set to 1, TRSRUN is not automatically cleared and the USB host controller automatically retries the transaction. If three transaction errors occur in succession, however, TRSRUN is automatically cleared and ERRIF is set.

<2> The error automatic retry interval cannot be altered. Retry transaction processing is started immediately whenever a transaction ends in error.

<3> When this bit is set to 0, no automatic retry is performed when a transaction error occurs.

ALS (bit 2): A port LS flag

<1> When this bit is set to 1, USB communication with the USB downstream A port is performed in LS mode.

<2> When this bit is set to 0, USB communication with the USB downstream A port is performed in FS mode.

CBS1 to CBS0 (bits 1, 0): A port bus state setting register

<1> These bits define the current bus state of the USB downstream A port.

Table 3.21.7 A Port Bus State Settings

A port Bus State	CBS [1: 0]
SE0	00
J state	10
K state	01

<2> See 3.21.4.15, "Bus active interrupt control register (BACINT)."

3.21.4.11 USB host status register (UHCSTA)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEBF	0000 0000	R/W	UHCSTA	TRSRUN	TRSTGL	TRSOVR	TGLERR	TTKN1	TTKN0	TRSISO	TGLMON

TRSRUN (bit 7): Transaction control

<1> Setting this bit to 1 starts transaction processing on the port that is selected by bit 5 (UHPSEL0) of UHCPRT.

<2> This bit is automatically cleared when the transaction ends (with ACK, NAK, error, or STALL).

<3> If a data toggle mismatch is found during an IN transaction, this flag is not cleared but the transaction is restarted even when an IN transaction ends with an ACK. See the description on TGLMON (bit 0).

<4> If NAK automatic retry is enabled (NKRTRY=1), this bit is not automatically cleared even when the transaction ends with a NAK response.

<5> If error automatic retry is enabled (ERRTRY=1), this bit is not automatically cleared unless the transaction ends in error two times in succession. This bit is cleared automatically when three transaction errors occur in succession.

TRSTGL (bit 6): Transaction data toggle

<1> A data packet with a packet ID of DATA0 is transmitted during a SETUP transaction irrespective of the setting of this bit. TRSTGL is automatically set to 1 when the transaction ends normally with an ACK.

<2> In an OUT transaction, the data packet whose packet ID matches TRSTGL is transmitted. TRSTGL is automatically inverted if the transaction ends with an ACK. However, the bit is not inverted in isochronous transfer mode.

<3> In an IN transaction, the receive data is transferred to RAM only when the packet ID of the data packet from the device matches TRSTGL. In isochronous transfer mode or when R2RAM (UHTST0, bit 3) is set to 1, however, the receive data is transferred to RAM regardless of whether a data toggle match or mismatch occurs. TRSTGL is automatically inverted if the transaction ends normally with an ACK on a data toggle match. However, the bit is not inverted in isochronous transfer mode.

TRSOVR (bit 5): Payload over flag

<1> This bit is automatically set to 1 when a volume of data exceeding the byte count specified in UHCCNT is received.

<2> The excess data is not transferred to RAM and the extra data of the receive data count register (UHCRX) is not counted.

<3> This flag must be cleared with an instruction.

TGLERR (bit 4): Data toggle error flag

<1> This bit is automatically set to 1 if an IN transaction ends with an ACK on a data toggle mismatch when TGLMON is set to 1.

<2> The state of this flag remains unchanged if an IN transaction ends with an ACK on a data toggle mismatch with TGLMON set to 0.

<3> This bit is automatically cleared to 0 regardless of the TGLMON setting if an IN transaction ends with an ACK on a data toggle match.

<4> This bit is automatically cleared to 0 when a SETUP or OUT transaction, or isochronous transfer ends with an ACK.

<5> This flag can also be cleared with an instruction.

TTKN1 to TTKN0 (bits 3, 2): Token setting

<1> These bits specify the type of the token to be transmitted.

Table 3.21.8 Token Settings

TTKN [1:0]	Token
00	OUT
01	Inhibited
10	IN
11	SETUP

TRSISO (bit 1): Isochronous transfer setting

<1> This bit must be set to 1 to perform isochronous transfer.

<2> This bit must always be set to 0 to perform SETUP transaction.

TGLMON (bit 0): Transaction toggle monitor

<1> If an IN transaction ends with an ACK when this bit is set to 1, ACKIF is automatically set to 1 and TRSRUN is automatically cleared to 0 regardless of whether a data toggle match or mismatch occurs.

<2> If an IN transaction ends with an ACK when this bit is set to 0, ACKIF is automatically set to 1 and TRSRUN automatically cleared to 0 only when a data toggle match occurs.

3.21.4.12 USB host count register (UHCCNT)

1) A total of 10 bits consisting of the bits of this register and bits 5 and 4 (UHCCN9 and UHCCN8) of BACINT (FEC4H) specify the number of data bytes to be transmitted (SETUP and OUT transactions) or the maximum payload size in receive mode (IN transaction).
2) The legitimate value range is from 000[H] to 3FF[H] (0 to 1023 bytes).

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC0	0000 0000	R/W	UHCCNT	UHCCN7	UHCCN6	UHCCN5	UHCCN4	UHCCN3	UHCCN2	UHCCN1	UHCCN0

3.21.4.13 USB host receive data count register (UHCRX)

1) The number of data bytes received during an IN transaction can be read from a total of 10 bits consisting of the bits of this register and bits 7 and 6 (UHCRX9 and UHCRX8) of BACINT (FEC4H).
2) A total of 10 bits consisting of the bits of this register and bits 7 and 6 (UHCRX9 and UHCRX8) of BACINT are updated when the IN transaction ends normally with an ACK.
3) If a data toggle mismatch occurs when TGLMON is set to 0, a total of 10 bits consisting of the bits of this register and bits 7 and 6 (UHCRX9 and UHCRX8) of BACINT are not updated. If TGLMON is set to 1, the bits are updated regardless of whether a data toggle match or mismatch occurs.
4) When more data bytes than specified in UHCCNT are received, the extra data bytes are not included in the count up.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC1	0000 0000	R/W	UHCRX	UHCRX7	UHCRX6	UHCRX5	UHCRX4	UHCRX3	UHCRX2	UHCRX1	UHCRX0

3.21.4.14 USB host address register (UHCADL, UHCADH)

1) This register defines the starting address of the RAM area to be used as a buffer area for storing transmit or receive data.
2) The legitimate value range is from 0000[H] to 1FFC[H].

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC2	0000 00HH	R/W	UHCADL	UHAD7	HUAD6	UHAD5	UHAD4	UHAD3	UHAD2	-	-
FEC3	HHH0 0000	R/W	UHCADH	-	-	-	UHAD12	UHAD11	UHAD10	UHAD9	UHAD8

3.21.4.15 Bus active interrupt control register (BACINT)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC4	0000 0000	R/W	BACINT	UHCRX9	UHCRX8	UHCCN9	UHCCN8	BACBIF	BACBIE	BACIF	BACIE

UHCRX9 to UHCRX8 (bits 7, 6): Receive data byte count

<1> See 3.21.4.13, "USB host receive data count register (UHCRX)."

UHCCN9 to UHCCN8 (bits 5, 4): Transmit data byte count/maximum receive mode payload size

<1> See 3.21.4.12, "USB host count register (UHCCNT)."

BACBIF (bit 3): B port bus active interrupt flag

<1> This bit is set to 1 when a bus state that is different from the one specified by BCBS[1:0] (BPTINT, bits 1 and 0) is found at the USB downstream B port.

<2> This bit must always be cleared with an instruction when the BCBS[1:0] setting is altered.

BACBIE (bit 2): B port bus active interrupt request enable control

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

BACIF (bit 1): A port bus active interrupt flag

<1> This bit is set to 1 when a bus state that is different from the one specified by CBS[1:0] (BUSCTR, bits 1 and 0) is found at the USB downstream A port.

<2> This bit must always be cleared with an instruction when the CBS[1:0] setting is altered.

BACIE (bit 0): A port bus active interrupt request enable control

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

3.21.4.16 USB host test register 0 (UHTST0)

1) This register is a test register.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC5	0000 0000	R/W	UHTST0	UHTST	UHCMPT	UHCMPK	VD3KIL	R2RAM	UHTAD2	UHTAD1	UHTAD0

UHTST (bit 7): Test bit

This bit must always be set to 0.

UHCMPT (bit 6): Test bit

This bit must always be set to 0.

UHCMPK (bit 5): Test bit

This bit must always be set to 0.

VD3KIL (bit 4): Test bit

This bit must always be set to 0.

R2RAM (bit 3): RAM transfer control

<1> When this bit is set to 1, the receive data is transferred to RAM regardless of whether a data toggle match or mismatch occurs during an IN transaction.

<2> When this bit is set to 0, the receive data is transferred to RAM only when a data toggle match occurs during an IN transaction. In isochronous transfer mode, however, the data is transferred to RAM regardless of whether a data toggle match or mismatch occurs.

UHTAD2 to UHTAD0 (bits 2 to 0): Test bits

3.21.4.17 USB host test register 1 (UHTST1)

1) This register is a test register.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC6	0000 0000	R/W	UHTST1	UHTDT7	UHTDT6	UHTDT5	UHTDT4	UHTDT3	UHTDT2	UHTDT1	UHTDT0

3.21.4.18 B port bus control register (BPTCTR)

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC7	HHHH 0000	R/W	BPTCTR	-	-	-	-	BLS	BPTRST	BPTRSM	BPTSOF

BLS (bit 3): B port LS flag

<1> When this bit is set to 1, USB communication with the USB downstream B port is performed in LS mode.

<2> When this bit is set to 0, USB communication with the USB downstream B port is performed in FS mode.

BPTRST (bit 2): B port bus reset transmission control

<1> Setting this bit to 1 transmits a bus reset signal (SE0) to the USB downstream B port.

<2> The bus reset signal continues to be transmitted until this bit is cleared with an instruction.

BPTRSM (bit 1): B port resume transmission control

<1> Setting this bit to 1 transmits a resume signal (K state) to the USB downstream B port.

<2> The resume signal continues to be transmitted until this bit is cleared with an instruction.

<3> When this bit is cleared, the USB host stops the transmission of the K state signal and transmits a low-speed EOP at the end.

BPTSOF (bit 0): B port SOF transmission control

<1> Setting this bit to 1 starts the frame timer, which automatically transmits an SOF packet to the USB downstream B port every 1 ms.

<2> No SOF packet is transmitted to B port even when this bit is set to 1 if a bus reset or resume signal is transmitted to B port at the same timing when an SOF packet is transmitted.

<3> When this bit is set to 1, the contents of frame number registers (UHFRML and UHFRMH) are counted up every 1 ms regardless of whether SOF packets are transmitted or not.

<4> Clear this bit with an instruction to stop automatic transmission of SOF packets.

3.21.4.19 B port interrupt control register 1 (BPTINT)

1) This register controls the B port interrupts.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEC8	0000 0000	R/W	BPTINT	BDTCIF	BDTCIE	BRSMIF	BRSMIE	BATCIF	BATCIE	BCBS1	BCBS0

BDTCIF (bit 7): B port device disconnection detection flag

<1> This flag is set when the SE0 state (UHBD+ = L, UHBD- = L) continues for 2.5 µs or more at the USB downstream B port.

<2> This flag remains set until the above-mentioned bus state is reset.

<3> This flag must be cleared with an instruction.

BDTCIE (bit 6): B port device disconnection detection interrupt request enable control

<1> When this bit and BDTCIF are set to 1, an interrupt request to vector address 002BH is generated.

BRSMIF (bit 5): B port FS resume/LS device connection detection flag

<1> This flag is set when the bus state (UHBD+ = L, UHBD- = H) continues for 2.5 µs or more at the USB downstream B port.

<2> This flag remains set until the above-mentioned bus state is reset.

<3> This flag must be cleared with an instruction.

BRSMIE (bit 4): B port FS resume/LS device connection detection interrupt request enable control

<1> When this bit and BRSMIF are set to 1, an interrupt request to vector address 002BH is generated.

BATCIF (bit 3): B port FS device connection/LS resume detection flag

<1> This flag is set when the bus state (UHBD+= H, UHBD- =L) continues for 2.5 µs or more at the USB downstream B port.

<2> This flag remains set until the above-mentioned bus state is reset.

<3> This flag must be cleared with an instruction.

BATCIE (bit 2): B port FS device connection/LS resume detection interrupt request enable control

<1> When this bit and BATCIF are set to 1, an interrupt request to vector address 002BH is generated.

BCBS1 to BCBS0 (bits 1, 0): B port bus state setting register

<1> These bits define the current bus state of the USB downstream B port.

Table 3.21.9 B Port Bus State Settings

B Port Bus State	BCBS[1:0]
SE0	00
J state	10
K state	01

2) See 3.21.4.15, "Bus active interrupt control register (BACINT)."

3.21.4.20 Port 2 data latch (P2)

1) Bits 6 and 7 of the port 2 data latch are used to control the output data on UHBD- and UHBD+.
2) Reading bits 6 and 7 with an instruction causes the data on the UHBD- and UHBD+ pins to be read in. If P2 (FE48) is manipulated using a NOT1, CLR1, SET1, DBZ, DBNZ, INC, or DEC instruction, however, the contents of the register are referenced instead of the data on the pins.
3) It is always possible to read the data on the UHBD- and UHBD+ pins 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

3.21.4.21 Port 2 data direction register (P2DDR)

1) Bits 6 and 7 of the port 2 data direction register are used to control the general-purpose port output mode of the UHBD- and UHBD+ pins, respectively.

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		UHBD-/UHBD+ Port State (n=6 or 7)	
P2n	P2nDDR	Input	Output
0	0	Enabled	USB communication mode
1	0	Enabled	
0	1	Enabled	CMOS low output
1	1	Enabled	CMOS high output

3.21.5 USB Host Communication Example

3.21.5.1 Setting up the clock

1) Connect a 12 MHz resonator across the CF1 and CF2 pins for the main clock. Set PLLCNT to 00[H] or 80[H]. (See 3.21.4.2, "USB PLL control register (PLLCNT).)"
2) Set bit 7 (CLKSGL) and bit 4 (CLKCB4) of the OCR register to 1 to use the main clock as the system clock.

* To perform USB communication, it is necessary to set the system clock frequency to 8 to 12 MHz.

Table 3.21.10 Sample Clock Settings

Main Clock Oscillation Frequency	PLLCNT	OCR
12 MHz	00[H] or 80[H]	92[H]

3) Load the UHCCTR register with C0[H] to enable the USB host.

** Below are examples of USB communication through the USB downstream A port (FS device). For an example of USB communication through B port, substitute the control registers for A port for those for B port. When performing USB communication through B port, set bit 6 (P26DDR) and bit 7 (P27DDR) of the P2DDR to 0.*

3.21.5.2 Detecting a device connection

1) Load the current bus state in bits 1 and 0 (CBS[1:0]) of the BUSCTR register. If no device is connected, set CBS[1:0] to 00 because the bus state is SE0.
2) Clear bit 1 (BACIF) of the BACINT register to 0.
3) If a device is connected, the bus state changes from SE0 to the J state and BACIF is set to 1. In this case, if bit 0 (BACIE) of the BACINT register is set to 1, a HOLD mode release signal and an interrupt request are generated.
4) If the J state continues for 2.5 μs or more after the device is connected, bit 1 (ATCIF) of the UHCINT register is set to 1. At this moment, an interrupt request is generated if bit 0 (ATCIE) of the UHCINT register is set to 1.

3.21.5.3 Detecting a device disconnection

1) Load the current bus state in bits 1 and 0 (CBS[1:0]) of the BUSCTR register. If a device is connected, set CBS[1:0] to 10 because the bus state is the J state.
2) Clear bit 1 (BACIF) of the BACINT register to 0.
3) When the device is disconnected, the bus state changes from the J state to the SE0 state and BACIF is set to 1. In this case, if bit 0 (BACIE) of the BACINT register is set to 1, a HOLD mode release signal and an interrupt request are generated.
4) If the SE0 state continues for 2.5 μs or more after the device is disconnected, bit 7 (DTCIF) of the UHCINT register is set to 1. At this moment, an interrupt is generated if bit 6 (DTCIE) of the UHCINT register is set to 1.

3.21.5.4 Transmitting a bus reset

1) Set bit 7 (RSTRUN) of the BUSCTR register to 1, and transmission of a bus reset will start.
2) Clear RSTRUN to 0 to stop the bus reset transmission.

3.21.5.5 Transmitting a resume signal

1) Set bit 6 (RSMRUN) of the BUSCTR register to 1, and resume transmission will start.
2) Clear RSMRUN to 0 to stop resume transmission .

3.21.5.6 Transmitting an SOF packet

1) Load the frame number registers (UHFRML and UHFRMH) with the initial frame number (0000[H]).
2) Set bit 5 (SOFRUN) of the BUSCTR register to 1, and an SOF packet is transmitted automatically every 1 ms. The frame number register value is incremented automatically every 1 ms.
3) Clear SOFRUN to 0 to stop transmitting SOF packets.

3.21.5.7 Setting up a SETUP transaction

1) Load the USB device address and endpoint number into the DEVADR and DEVEPN registers, respectively.
2) Store the 8-byte data to be transmitted as the data packet for the SETUP transaction in RAM, and load the starting address of RAM into the UHCADL and UHCADH registers.
3) Load the byte size (8 bytes for the SETUP transaction) of the data packet to be transmitted into the UHCCNT register.
4) Set bits 3 and 2 (TTKN[1:0]) of the UHCSTA register for the SETUP token (TTKN [1:0] = 11) and set bit 7 (TRSRUN) to 1.
5) If the transaction ends normally, TRSRUN is cleared to 0 and bit 7 (ACKIF) of the TRSINT register is set to 1. At this moment, an interrupt is generated if bit 6 (ACKIE) of the TRSINT register is set to 1.
6) If the transaction ends with an error, TRSRUN is cleared to 0 and bit 3 (ERRIF) of the TRSINT register is set to 1. At this moment, an interrupt is generated if bit 2 (ERRIE) of the TRSINT register is set to 1.

3.21.5.8 Setting up an OUT transaction

1) Set up the DEVADR and DEVEPN registers (this step may be skipped if no change needs to be made in the current settings).
2) Store the data to be transmitted as the data packet of the OUT transaction in RAM, and load the starting address of RAM into the UHCADL and UHCADH registers.
3) Load the byte size of the data packet to be transmitted into the UHCCNT register.
4) Place the toggle information of the data packet to be transmitted in bit 6 of the UHCSTA register, then set bits 3 and 2 (TTKN[1:0]) for the OUT token (TTKN[1:0]=00) and set bit 7 (TRSRUN) to 1.
5) If the transaction ends normally, TRSRUN is cleared to 0 and bit 7 (ACKIF) of the TRSINT register is set to 1. At this moment, an interrupt is generated if bit 6 (ACKIE) of the TRSINT register is set to 1.
6) If the transaction ends with an NAK, with an error, or with a STALL, TRSRUN is cleared to 0 and bit 5 (NAKIF), bit 3 (ERRIF), or bit 1 (STLIF) of the TRSINT register is set to 1, respectively. In this case, an interrupt is generated if bit 4 (NAKIE), bit 2 (ERRIE), or bit 0 (STLIE) of the TRSINT register is set to 1, respectively.

3.21.5.9 Setting up an IN transaction

1) Set up the DEVADR and DEVEPN registers (this step may be skipped if no change needs to be made to the current settings).
2) Load the starting address of RAM for storing receive data into the UHCADL and UHCADH registers.
3) Load the maximum payload size or the byte size of the data to receive into the UHCCNT register.
4) Place the toggle information of the receive data packet in bit 6 of the UHCSTA register, then set bits 3 and 2 (TTKN [1:0]) for the IN token (TTKN [1:0] = 10) and set bit 7 (TRSRUN) to 1.
5) If the transaction ends normally, TRSRUN is cleared to 0 and bit 7 (ACKIF) of the TRSINT register is set to 1. At this moment, an interrupt is generated if bit 6 (ACKIE) of the TRSINT register is set to 1.
6) The number of received data bytes can be read from the UHCRX register. The received data is stored in the RAM area that is specified in step 2). If a data packet that does not match the toggle information that is set up in step 4) is received, however, the received data is not stored in RAM at all.
7) If the transaction ends with an NAK, with an error, or with a STALL, TRSRUN is cleared to 0 and bit 5 (NAKIF), bit 3 (ERRIF), or bit 1 (STLIF) of the TRSINT register is set to 1, respectively. In this case, an interrupt is generated if bit 4 (NAKIE), bit 2 (ERRIE), or bit 0 (STLIE) of the TRSINT register is set to 1, respectively.

3.21.6 USB Host HALT Mode Operation

1) No USB host communication involving data transmission or reception can be carried out normally in HALT mode since automatic data transfer between RAM and the transmit/receive buffer is interrupted when the CPU enters HALT mode.
2) HALT mode can be released by a USB host interrupt that involves neither data transmission nor reception.

3.22 Infrared Remote Control Receiver Circuit 2 (REMOREC2)

3.22.1 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 filter function
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 (can support 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

3.22.2 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 the subclock oscillation source (the RM2CK reference clock is selected from 8 sources) to identify the data as 0, 1, or error. Data that is found normal is loaded in the remote control receive shift register (RM2SFT). Every time 8 bits of data are loaded in the register, they are transferred to the remote control receive data register (RM2RDT). At this moment, the data transfer flag is set. The receive end 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> RM2SFT-to-RM2RDT data transfer

<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 in X'tal HOLD mode 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.

The 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

3.22.3 Circuit Configuration

3.22.3.1 Remote control receive control register (RM2CNT) (8-bit register)

1) This register controls the remote control receive operation.

3.22.3.2 Remote control receive interrupt control register (RM2INT) (8-bit register)

1) This register controls the remote control receive interrupts.
2) When the REMOREC2 starts a receive operation with RM2CK selected as the subclock oscillation source, X'tal HOLD mode can be released using the interrupt occurring in the REMOREC2 circuit.

3.22.3.3 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 data loading direction (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) The 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.

3.22.3.4 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.

3.22.3.5 Remote control receive bit counter and prescaler setting 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 when a receive operation is completed, and the bits (RM2GPR1,0/RM2DPR1,0) that define the count value 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 set to 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.
3) The value of RM2GPR1 and RM2GPR0 exert no influence on the receive operation if RM2FMT2 to RM2FMT0 are set to 2 or 3.

3.22.3.6 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 measurement 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 setting registers when receiving the guide pulse and the data pulse. The count is set up by RM2GPR1 and RM2GPR0 (RM2CTPR, bits 7 and 6), and RM2DPR1 and RM2DPR0 (RM2CTPR, bits 5 and 4).

   A count clock to RM2MJCT is generated for 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

3.22.3.7 Remote control receive guide pulse width setting 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 2 or 3.

3.22.3.8 Remote control receive data 0 pulse width setting 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.

3.22.3.9 Remote control receive data 1 pulse width setting 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.

3.22.3.10 Remote control receive guide pulse and data pulse width high byte setting 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 the RM2SFT.

3.22.3.11 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 3.22.5 "Remote Control Receiver Circuit Operation" for the operation of the REMOREC2 in various receive formats.

3.22.3.12 Remote control receive noise filter (RM2NFLT)

1) This noise filter rejects occurrences of the 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.

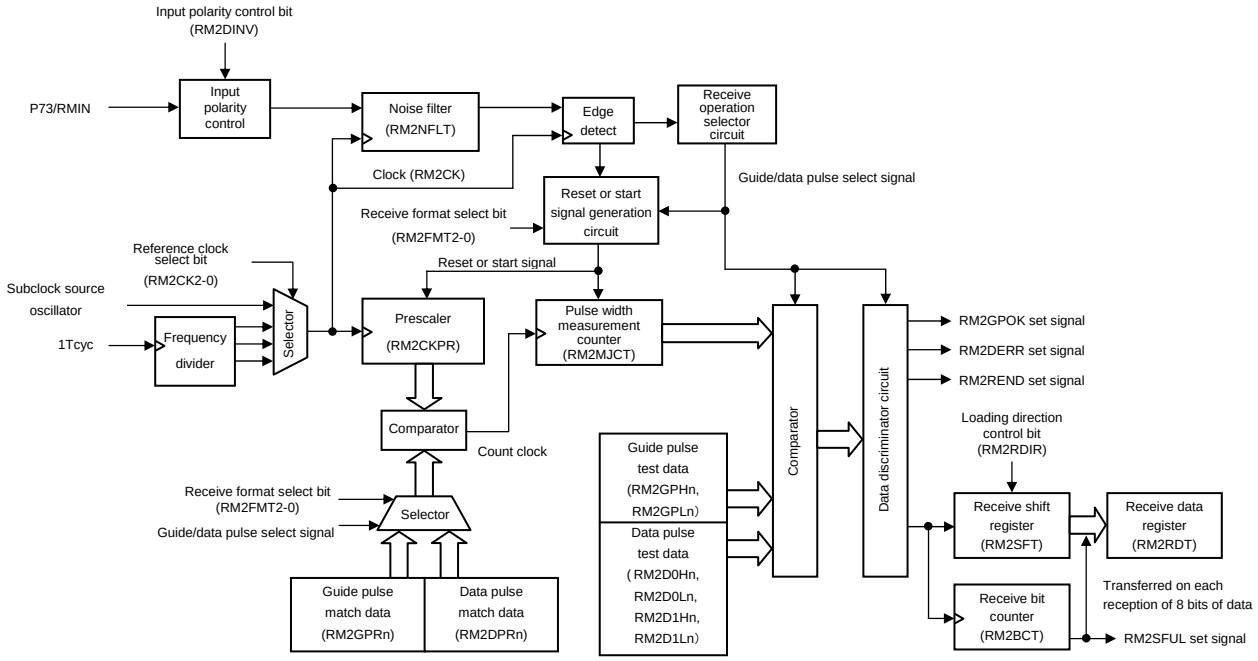


Figure 3.22.1 Infrared Remote Control Receiver Circuit 2 Block Diagram (RM2FMT2 to 0 = 0 to 2 setting)

3.22.4 Related Registers

3.22.4.1 Remote control receive control register (RM2CNT)

1) This register is an 8-bit register that controls the remote control receive operation.

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): Remote control receive operation control

When this bit is set to 0, the remote control receiver circuit stops operation.

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):
RM2FMT0 (bit 4):
Remote control receive format select

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 values other than those listed above are inhibited.
* See the Subsection 3.22.5 "Remote Control Receiver Circuit Operation" for the operation of the REMOREC2 in various receive formats.

RM2DINV (bit 3): Remote control 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.

REMOREC2

RM2CK2 (bit 2):
RM2CK1 (bit 1): } **Remote control 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

Note:

- *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.*

3.22.4.2 Remote control receive interrupt control register (RM2INT)

1) This register is an 8-bit register that controls the remote control receive interrupts.
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 receive operation is started 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 when RM2FMT2 to RM2FMT0 are set 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, an 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 receive 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, an 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 the 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, an X'tal HOLD mode release signal and an interrupt request to vector address 0013H are generated.

RM2REND (bit 1): Receive end flag

This bit is set when the end of the receive format conditions is detected.

This flag must be cleared with an instruction.

RM2ENIE (bit 0): Receive end interrupt request enable control

When this bit and RM2REND are set to 1, an X'tal HOLD mode release signal and an interrupt request to vector address 0013H are generated.

Note:

- *RM2GPOK is not set when RM2FMT2 to RM2FMT0 are set to 2 or 3.*

3.22.4.3 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) The RM2SFT is reset when one of the following conditions occurs:
   <1> The remote control 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> RM2SFT-to-RM2RDT data transfer 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).*

3.22.4.4 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).*

3.22.4.5 Remote control receive bit counter and prescaler setting 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 at the end of reception, and the bits (RM2GPR1, 0/RM2DPR1, 0) that define the count value of RM2CKPR in 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 the RM2BCT.

   The RM2BCT is reset when one of the following conditions occurs:

   <1> The remote control 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) Bits 3 to 0 of this register are 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):
RM2GPR0 (bit 6): } **Guide pulse receive mode RM2CKPR count select**

RM2DPR1 (bit 5):
RM2DPR0 (bit 4): } **Data pulse receive mode RM2CKPR count select**

* 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. Then, 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 suspended (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 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 set in RM2GPR1 and RM2GPR0 exerts no influence on the receive operation when RM2FMT2 to RM2FMT0 are set to 2 or 3.*

3.22.4.6 Remote control receive guide pulse width setting 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 set in this register exerts no influence on the receive operation when RM2FMT2 to RM2FMT0 are set to 2 or 3.*

3.22.4.7 Remote control receive data 0 pulse width setting 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

3.22.4.8 Remote control receive data 1 pulse width setting 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

3.22.4.9 Remote control receive guide pulse and data pulse width high byte setting 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 the 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 from the remote control is loaded into the RM2SFT register LSB first.

When this bit is set to 1, the data received from the remote control is loaded into the RM2SFT register MSB first.

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.

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 3.22.5 "Remote Control Receiver Circuit Operation" for the operation of the REMOREC2 in various receive formats.*

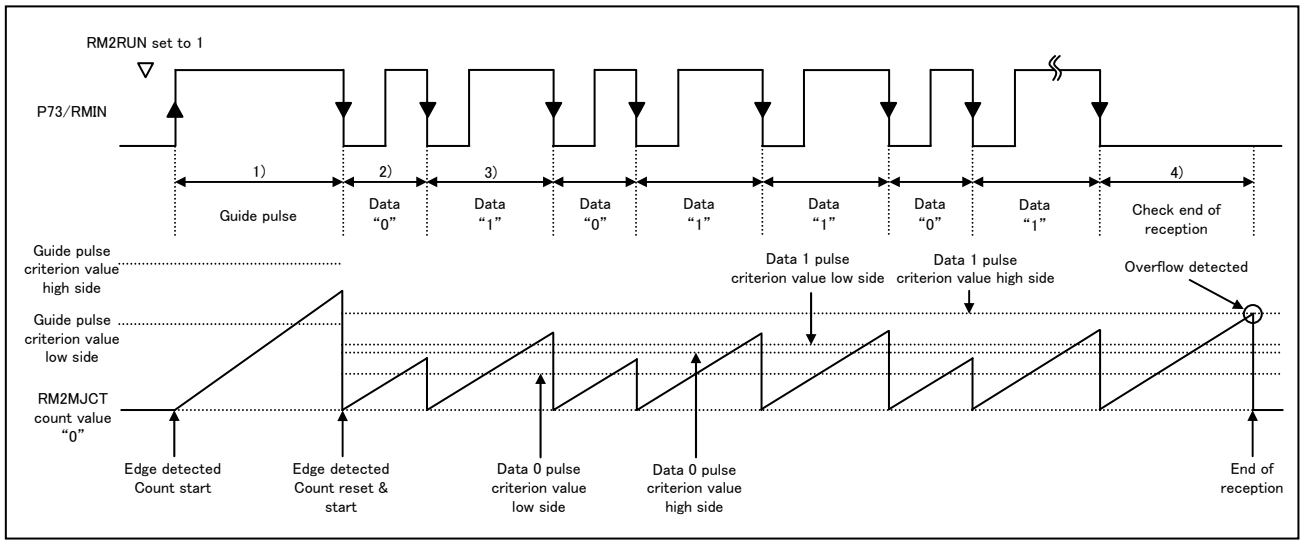
3.22.5 Remote Control Receiver Circuit Operation

3.22.5.1 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 a 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 the value given by RM2GPL4 to RM2GPL0 < the 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 given 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 the value given by RM2D0L4 to RM2D0L0 < the value given by RM2D0H4 to RM2D0H0 ≦ the value given by RM2D1L4 to RM2D1L0 < the 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.

*** 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, the 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 falls outside the valid criterion value range, the REMOREC2 sets the RM2DERR flag and returns to the idle state, waiting for a guide pulse.

(4) The number of received data bits is counted by the RM2BCT. When receiving a 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.

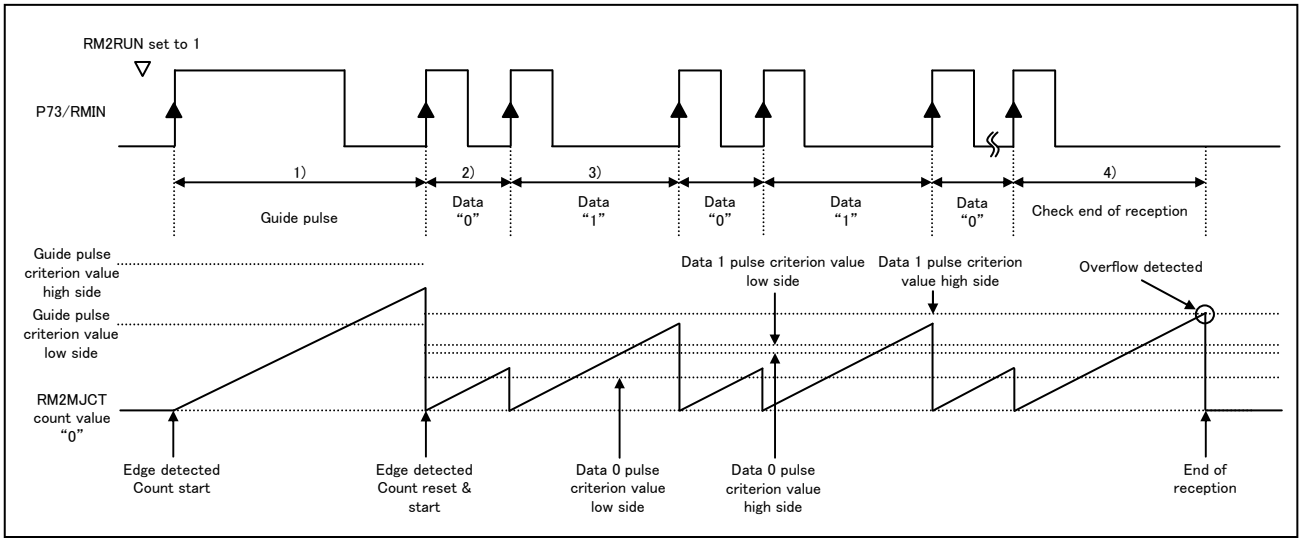
(5) When the REMOREC2 detects the end of reception condition, it sets the RM2REND and RM2HOLD flags and suspends operation. Subsequently, when the RM2SFT is read, the REMOREC2 clears the RM2HOLD flag and enters the idle state, waiting for a guide pulse (resuming the receive operation).

3.22.5.2 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 receive format A. Refer to "Setting up the receive format A criterion values."

*** Receive format B receive operation**

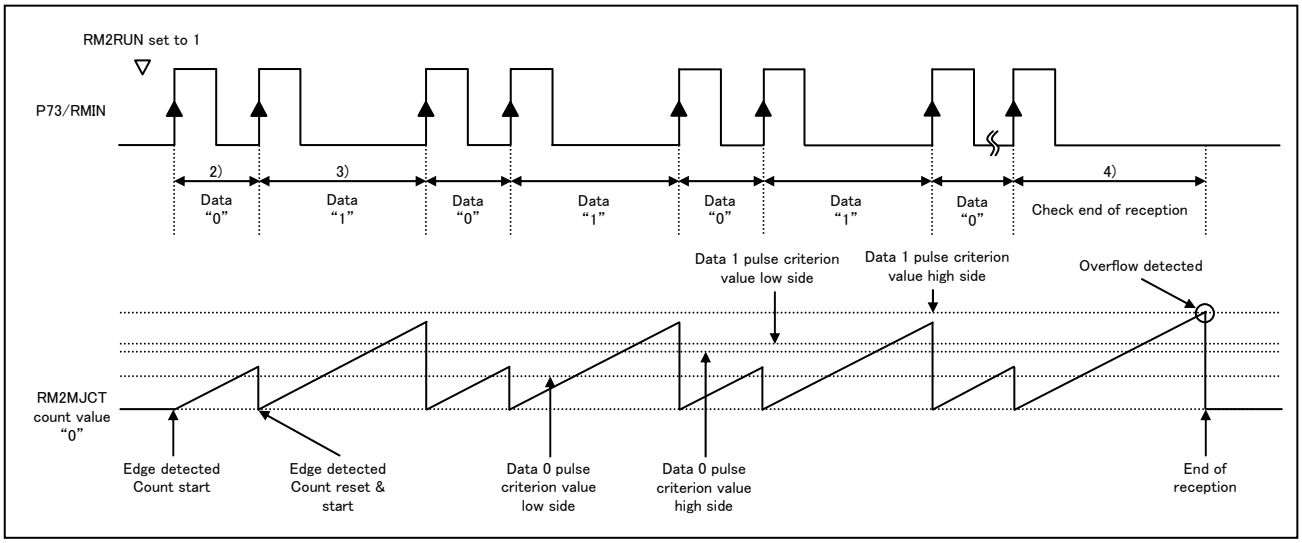
The REMOREC2 takes the same actions for receive format B as for receive format A. Refer to "Receive format A receive operation."

3.22.5.3 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 receive format A. Refer to "Setting up the receive format A criterion values."

*** Receive format C receive operation**

(1) When the REMOREC2 detects the first rising edge 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 falls outside the valid criterion value range, the REMOREC2 sets the RM2DERR flag and returns to the idle state, waiting for a next rising edge.

(4) The number of received data bits is counted by the RM2BCT. When receiving a 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.

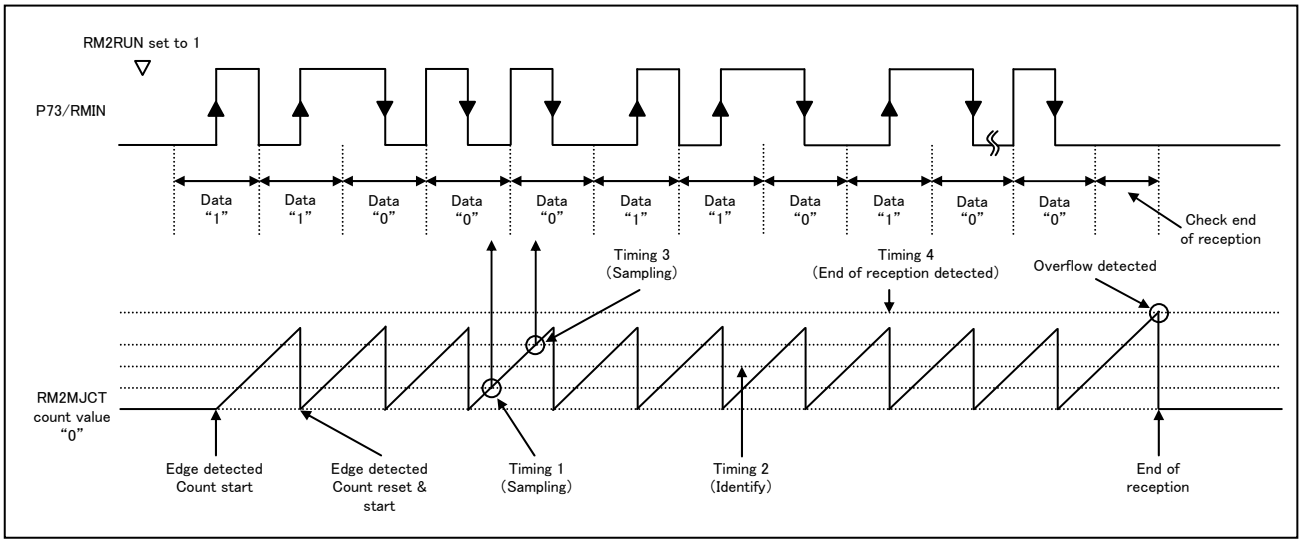
(5) When the REMOREC2 detects the end of reception condition, it sets the RM2REND and RM2HOLD flags and suspends operation. Subsequently, when the RM2SFT is read, the REMOREC2 clears the RM2HOLD flag and enters the idle state, waiting for the next rising edge (resuming the receive operation).

3.22.5.4 Receive operation when "receive format D" is specified

• Receive format D outline

Guide pulse	: No
Data encoding system	: Manchester coding
Stop bits	: No

*** 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 the value given by RM2D0L4 to RM2D0L0 < the value given by RM2D0H4 to RM2D0H0 < the value given by RM2D1L4 to RM2D1L0 < the value given by RM2D1H4 to RM2D1H0.

Note: The minimum criterion value is RM2CK × 4. The 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 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 identified as 0 or 1, the data (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 an error, the REMOREC2 sets the RM2DERR flag and returns to the idle state, waiting for the 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).

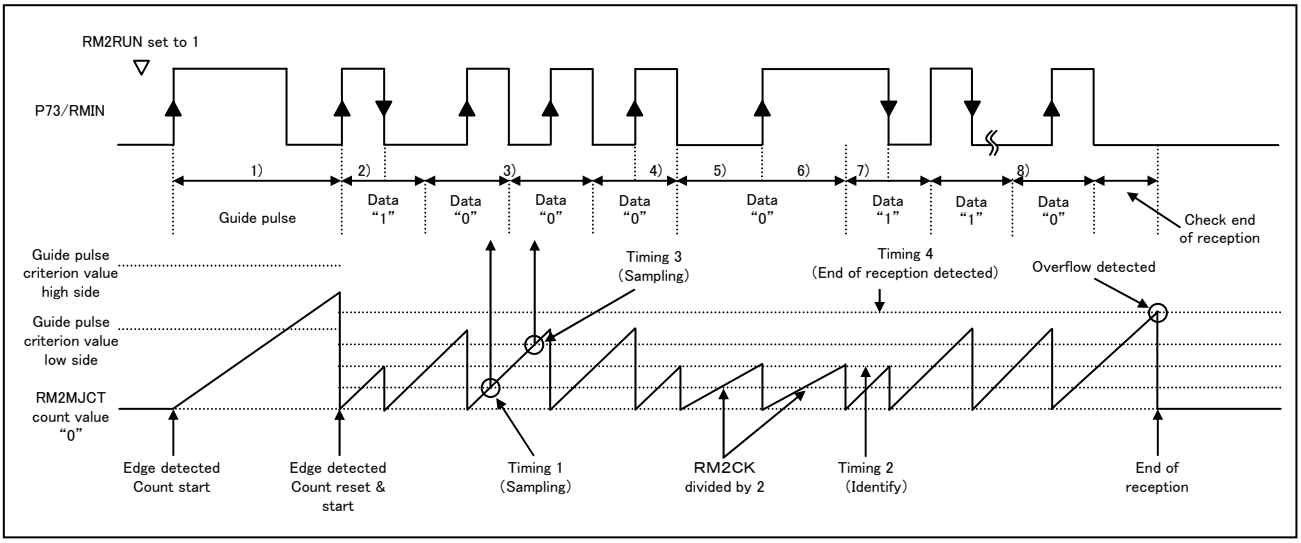
(7) When the REMOREC2 detects the end of reception condition, it sets the RM2REND and RM2HOLD flags and suspends operation. Subsequently, when the RM2SFT is read, the REMOREC2 clears the RM2HOLD flag and enters the idle state, waiting for the next rising edge (resuming the receive operation).

3.22.5.5 Receive operation when "receive format E" is specified

• Receive format E outline

Guide pulse	: Yes
Data encoding system	: Manchester coding
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. Refer to "Setting up the receive format B criterion values."

The procedure for setting up the data pulse receive timings for receive format E is identical to that for receive format D. Refer to "Setting up the receive format D timings."

Note: The minimum criterion value is RM2CK × 4. The interval between the low and high sides of the 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), (7) or (3).

(5) If identified as 0 or 1, the data (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 to 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).

(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 processing as in step (5). It also resets the RM2MJCT and divides the frequency of RM2CK by 2. If the data is identified as an error, the REMOREC2 performs the same processing as in 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 processing as in step (5). It also resets the RM2MJCT and resets RM2CK to the 1/1 frequency. If the data is identified as an error, the REMOREC2 performs the same processing as in 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) processing when it detects the end of reception condition.

4. Control Functions

4.1 Interrupt Function

4.1.1 Overview

This series of microcontrollers has the capability to control three levels of multiple interrupts, i.e., low level (L), high level (H), and highest level (X). The master interrupt enable register and interrupt priority control register 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. 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 previous state of the PC and interrupt level.

2) Multilevel interrupt control
   - The interrupt function supports three levels of interrupts, i.e., 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 the level 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. When interrupt requests of the same level occur at the same time, the one whose vector address is the lowest has 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 operation is performed to the IE (FE08H) or 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	Level	Interrupt Source
1	00003H	X or L	INT0
2	0000BH	X or L	INT1
3	00013H	H or L	INT2/T0L/INT4/UHC-A bus active/UHC-B bus active/ remote control receive
4	0001BH	H or L	INT3/INT5/base timer
5	00023H	H or L	T0H/INT6/UHC-A device connection/UHC-A device disconnection/UHC-A resume
6	0002BH	H or L	TIL/TIH/INT7/AIF start/SMIIC0/UHC-B device connection/ UHC-B device disconnection/UHC-B resume
7	00033H	H or L	SIO0/UART1 receive end
8	0003BH	H or L	SIO1/SIO4/UART1 buffer empty/UART1 transmit end/ AIF end
9	00043H	H or L	ADC/T6/T7/UHC-ACK/UHC-NAK/UHC error/ UHC-STALL
10	0004BH	H or L	Port 0/PWM0/PWM1/T4/T5/UHC-SOF

   - Priority level: X > H > L
   - When interrupts of the same level occur at the same time, the interrupt with the lowest vector address is processed first.

7) It is necessary to manipulate the following special function registers to enable interrupts and to specify their priority.
   - 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 state of the interrupt level flag 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 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 five systems of oscillator circuits, i.e., a main clock oscillator, a subclock oscillator, a low-speed RC oscillator, a medium-speed RC oscillator, and a USB PLL oscillator as system clock generator circuits. The low- and medium-speed RC oscillator circuits have internal resistors and capacitors, so that no external circuit is required.

The system clock can be selected from these five 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 five types of clocks generated by the main clock oscillator, subclock oscillator, low-speed RC oscillator, medium-speed RC oscillator, and USB 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 has 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
   - The above mentioned five systems of oscillators are stopped/started by instructions.

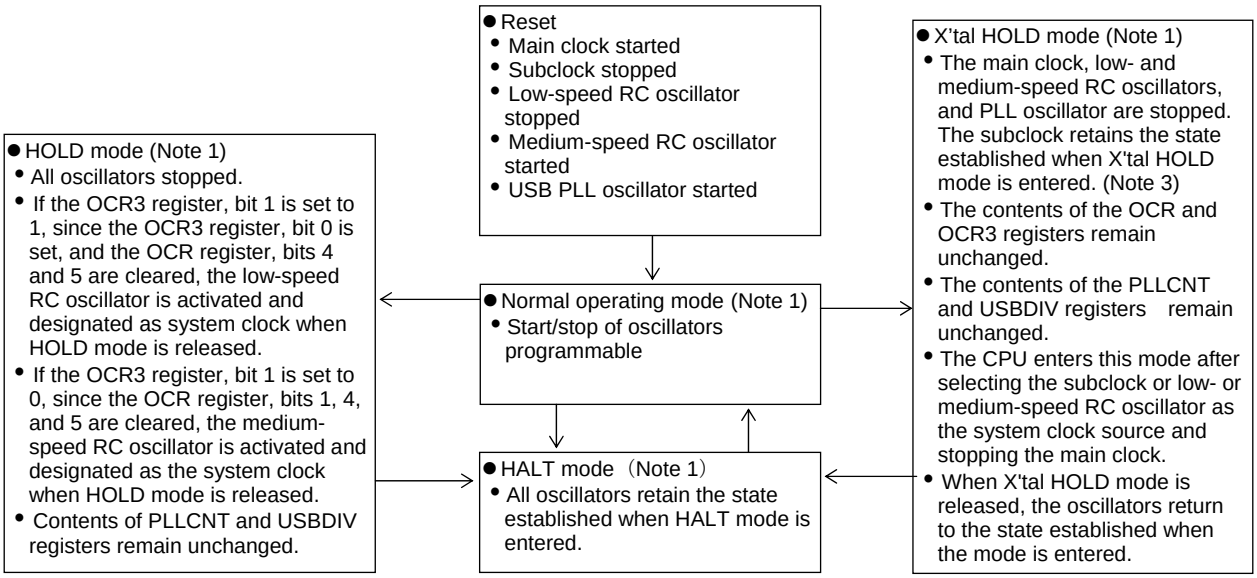
4) Multiplexed I/O pin function
   - The crystal oscillator pin XT1 can also be used as an input port, and XT2 as an input/output port.

5) Oscillator circuit states and operating modes

Mode/Clock	Main Clock	Subclock	Low-speed RC Oscillator (Note 1)	Medium-speed RC Oscillator	USB PLL Oscillator	System Clock
Reset	Running	Stopped	Stopped	Running	Running	Medium-speed RC oscillator
Normal mode	Programmable	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	State established at entry time
HOLD	Stopped	Stopped	Stopped	Stopped	Stopped	Stopped
Immediately after exit from HOLD mode	State established at entry time	State established at entry time	Running (Note 2)	Running (Note 2)	State established at entry time	Medium-/low-speed RC oscillator
X'tal HOLD	Stopped	State established at entry time	Stopped (Note 3)	Stopped	Stopped	Stopped
Immediately after exit from 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	State established at entry time

Note: See Section 4.3 "Standby Function" for the procedures to enter and exit the microcontroller operating modes.

System Clock



Note 1: The low-speed RC oscillation is also controlled directly by the watchdog timer. Its oscillation in the standby mode is also controlled by the watchdog timer. For more information, see Section 4.6, "Watchdog Timer (Internal Timer Type)."

Note 2: Either the medium- or low-speed RC oscillator circuit is automatically enabled and designated as the system clock after HOLD mode is exited according to the value of bit 1 of the oscillation control register 3 (OCR3) that is established on entry into HOLD mode.

Note 3: The low-speed RC oscillator retains the state that is established on entry into X'tal HOLD mode if the CPU enters X'tal HOLD mode when the base timer has been activated with the low-speed RC oscillator selected as the base timer input clock source.

6) It is necessary to manipulate the following special function registers to control the system clock.
   • USBDIV, PCON, CLKDIV, PLLCNT, OCR, OCR3

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE04	0000 0000	R/W	USBDIV	USBDIV7	DVCKON	DVCKDR	USBDIV4	USBDIV3	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
FE7C	0000 0000	R/W	OCR3	OCR3B7	OCR3B6	OCR3B5	OCR3B4	OCR3B3	OCR3B2	SRCSEL	SRCSTART

4.2.3 Circuit Configuration

4.2.3.1 Main clock oscillator circuit

1) This circuit is prepared for oscillation by connecting a ceramic resonator, a capacitor, and a damping resistor to the CF1 and CF2 pins.
2) When the main clock is not to be used, connect CF1 to VDD and keep the CF2 pin open.

4.2.3.2 Subclock oscillator circuit

1) This circuit is prepared for oscillation by connecting a crystal resonator (32.768 kHz typ), a capacitor, and a damping resistor to the XT1 and XT2 pins and controlling the OCR register.
2) The data at the XT1 and XT2 pins can be read as bits 2 and 3 of the OCR register.
3) If the XT1 and XT2 pins are not to be used as a subclock or general-purpose I/O ports, pull-down the XT1 pin with a resistor of 100 kΩ or less and configure the XT2 pin for low output.

4.2.3.3 Internal low-speed RC oscillator circuit

1) This circuit oscillates according to the internal resistors and capacitors (30 kHz typ).
2) The circuit serves as the system clock to be used for low-power, low-speed operation or as the count clock source for the internal timer-type watchdog timer.

4.2.3.4 Internal medium-speed RC oscillator circuit

1) This circuit oscillates according to the internal resistors and capacitors (1 MHz typ).
2) The clock from the medium-speed RC oscillator is designated as the system clock after the reset state is released. After HOLD mode is released, the clock from either the medium- or low-speed RC oscillator is selected as the system clock according to the value of bit 1 of the oscillation control register 3 (OCR3) that is established when HOLD mode is entered.

4.2.3.5 Internal PLL oscillator circuit for the USB

1) The main clock is oscillated by connecting a ceramic resonator, capacitor, and damping resistor to the CF1 and CF2 pins.
2) An external circuit (see the Data Sheet) is connected to the UFILT/P34 pin.
3) The internal multiplier circuit generates 48 MHz clock for the USB using the main clock as the reference clock.
4) The 4 to 12 MHz clock which is derived by frequency-dividing the 48 MHz clock for the USB can be supplied to the system as the system clock. When driving the USB interface control circuit, however, it is necessary to set the PLL oscillator circuit to generate an 8 to 12 MHz clock.

4.2.3.6 Power control register (PCON) (3-bit register)

1) This register specifies the operating mode (normal/HALT/HOLD/X'tal HOLD).

4.2.3.7 Oscillation control register (OCR) (8-bit register)

1) This register controls the start/stop operation of the main clock, subclock, and medium-speed RC oscillator circuits.
2) This register selects the system clock.
3) The register sets the frequency division ratio of the oscillator 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.

4.2.3.8 XT2 general-purpose port output control register (XT2PC) (8-bit register)

1) This register controls the general-purpose output (N-channel open drain) at the XT2 pin.

4.2.3.9 Oscillation control register 3 (OCR3) (8-bit register)

1) This register controls the start/stop operation of the low-speed oscillator circuit.
2) The register controls the RC clock (medium- or low-speed RC oscillator) selector.

4.2.3.10 USB PLL control register (PLLCNT) (8-bit register)

1) This register controls the start/stop operation of the PLL oscillator circuit.
2) The register selects the frequency of the PLL reference clock (resonator connected to the CF pin).

4.2.3.11 USB frequency-divided clock control register (USBDIV) (8-bit register)

1) This register determines the frequency-divided clock (selected from 4, 4.8, 6, 8, and 12 MHz) of the 48 MHz clock for the USB.
2) This register controls the high-speed clock (main clock/USB frequency-divided clock) selector.

4.2.3.12 System clock frequency 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 supported.

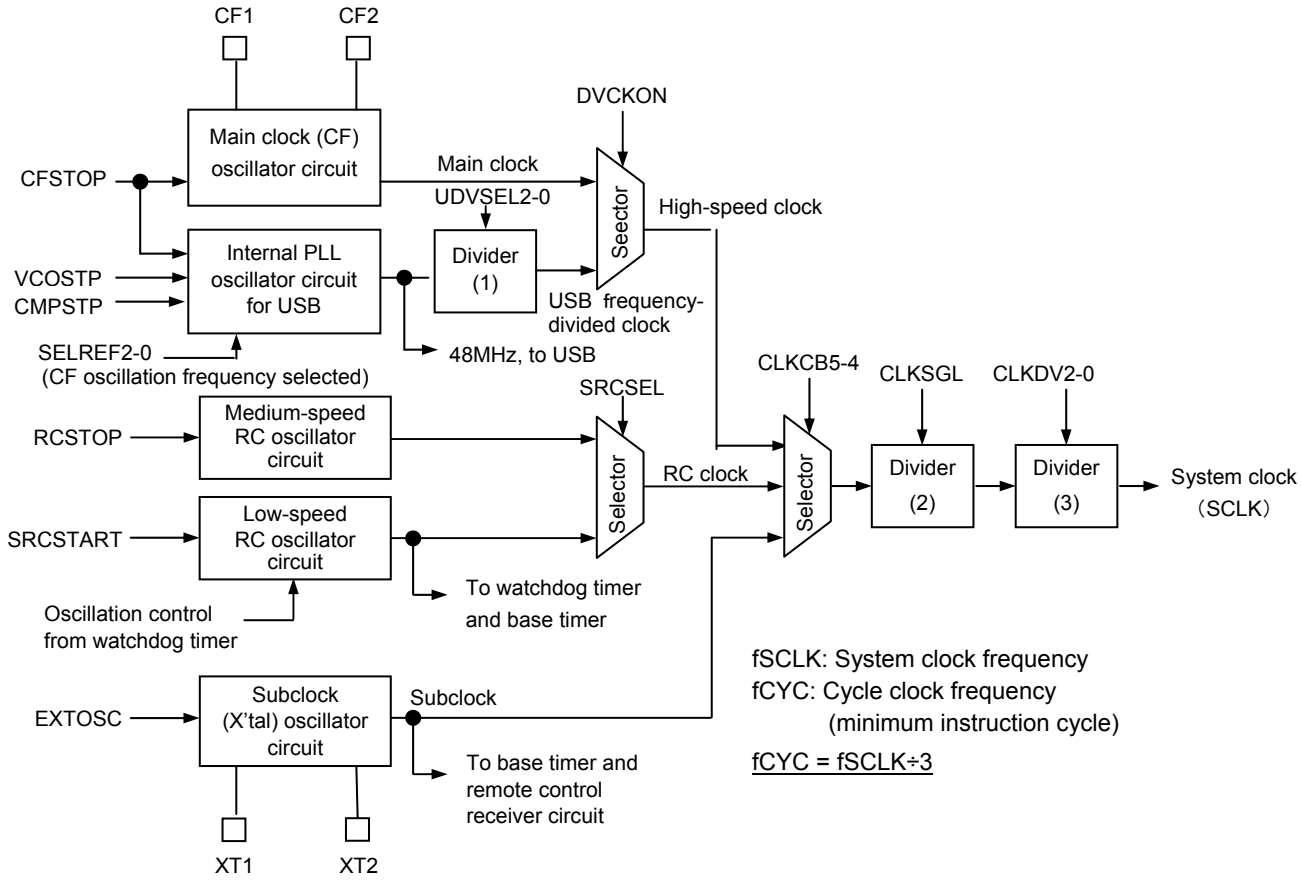


Fig. 4.2.1 System Clock Generator Block Diagram

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.
- When the microcontroller enters HOLD mode, all oscillators (main clock, subclock, low-/medium-speed RC, and PLL) are suspended and the related registers are placed in the states described below.

  If the OCR3 register, bit 1 is set to 1, the OCR3 register, bit 0 is set and the OCR register, bits 4 and 5 are cleared.

  If the OCR3 register, bit 1 is set to 0, the OCR register, bits 1, 4, and 5 are cleared.
- When the CPU exits HOLD mode, the low- or medium-speed RC oscillator starts oscillation and is designated as the system clock source.

  The main clock, subclock, and PLL oscillators return to their state that was established before the CPU entered HOLD mode.
- When the CPU enters X'tal HOLD mode, all oscillators except the subclock (i.e., main clock, low-/medium-speed RC, PLL) are suspended but the state of the OCR and OCR3 registers remains unchanged. If, however, the CPU enters X'tal HOLD mode with the low-speed RC oscillator selected as the clock source to the base timer, the low-speed RC oscillator retains the state that is established when X'tal HOLD is entered.
- Since it is impossible to secure the oscillation stabilization time for the main clock after the CPU exits X'tal HOLD mode, it is necessary to assign the system clock to either the subclock or low- or medium-speed RC oscillator clock when X'tal HOLD mode is entered.
- Since X'tal HOLD mode is used usually for low-current clock counting or when the infrared remote control receiver is in standby state, less current will be consumed if the system clock is switched to the subclock or low-speed RC, and the main clock, medium-speed RC, and PLL oscillators are suspended before the CPU enters X'tal HOLD mode.

<2> XTIDLE must be cleared with an instruction.

<3> PDN is cleared when a HOLD mode release signal (INT0, INT1, INT2, INT4, INT5, port 0, base timer, infrared remote control receive, or UHC bus active interrupt) 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.

Note: The low-speed RC oscillation is also controlled directly by the watchdog timer. Its oscillation in the standby mode is also controlled by the watchdog timer. For more information, see Section 4.6, "Watchdog Timer (Internal Timer Type)."

4.2.4.2 Oscillation control register (OCR) (8-bit register)

1) This register is an 8-bit register that is used to select the system clock frequency division ratio, to control the operation of the oscillator circuits, to select the system clock, and to read 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 frequency 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 and XT2 function control

<1> When this bit is set to 1, the XT1 and XT2 pins serve as the pins for subclock oscillation and prepare for oscillation when a crystal resonator (32.768 kHz typ), capacitors, and damping resistors are connected. When the OCR register is read in this case, bit 3 reads the data on 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 on the XT2 pin and bit 2 reads the data on the XT1 pin.

<3> This bit is cleared on reset and the XT1 and XT2 pins are configured for input.

Note: The XT2 general-purpose port output function is disabled when this bit is set to 1.

CLKCB5 to CLKCB4 (bits 5, 4): System clock select

<1> CLKCB5 and CLKCB4 are used to select the system clock.

<2> CLKCB5 and CLKCB4 are cleared on reset or when HOLD mode is entered, and the system clock is set to the RC clock.

CLKCB5	CLKCB4	System Clock
0	0	RC clock
0	1	High-speed clock
1	0	Subclock
1	1	High-speed clock

* See Figure 4.2.1 for details.

XT2IN (bit 3): XT2 data (read only)

XT1IN (bit 2): XT1 data (read only)

<1> Data that can be read via XT1IN varies as shown 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): Medium-speed RC oscillator circuit control

<1> Setting this bit to 1 stops the medium-speed RC oscillator circuit.

<2> Setting this bit to 0 starts the medium-speed RC oscillator circuit.

<3> This bit is cleared on reset and oscillation becomes possible.

<4> When the microcontroller enters HOLD mode, this bit is set as described below according to the value of bit 1 of the OCR3 register.

- If OCR3 register, bit 1 is set to 1, the state of this bit remains unchanged.
- If OCR3 register, bit 1 is set to 0, this bit is cleared and when the CPU exits HOLD mode, the oscillator starts oscillation and the medium-speed RC oscillator is set to the system clock source.

CFSTOP (bit 0): CF oscillator circuit control

<1> Setting this bit to 1 stops the CF oscillator circuit.

<2> Setting this bit to 0 starts the CF oscillator circuit.

<3> This bit is cleared on reset and oscillation becomes possible.

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) 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 to XT2PCB2 (bits 7 to 2): General-purpose flags

These bits can be used as general-purpose flags.

Manipulating 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 the general-purpose output port function, set EXTOSC to 0.

4.2.4.4 Oscillation Control Register 3 (OCR3) (8-bit register)

1) This register is an 8-bit registert used to control the operation of the low-speed RC oscillator circuit and to select the RC clock source.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE7C	0000 0000	R/W	OCR3	OCR3B7	OCR3B6	OCR3B5	OCR3B4	OCR3B3	OCR3B2	SRCSEL	SRCSTART

OCR3B7 to OCR3B2 (bits 7 to 2): General-purpose flags

These bits can be used as general-purpose flags.

Manipulating these bits exerts no influence on the operation of this function block.

SRCSEL (bit 1): RC clock select

<1> A 1 in this bit selects the low-speed RC oscillator clock as the RC clock.

<2> A 0 in this bit selects the medium-speed RC oscillator clock as the RC clock.

<3> This bit is cleared on reset.

* See Figure 4.2.1 for more information.

SRCSTART (bit 0): Low-speed RC oscillator circuit control

<1> A 1 in this bit starts the low-speed RC oscillator circuit.

<2> A 0 in this bit stops the low-speed RC oscillator circuit.

<3> This bit is cleared and the oscillator circuit is disabled for oscillation on reset.

<4> This bit is set as described below according to the value of SRCSEL when the CPU enters HOLD mode,.

- If SRCSEL is set to 1, this bit is set and, after HOLD mode is released, the low-speed RC oscillator starts oscillation and the system clock is set to the low-speed RC oscillator clock.
- If SRCSEL is set to 0, the state of this bit remains unchanged.

4.2.4.5 USB PLL control register (PLLCNT) (8-bit register)

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 to SELREF0 (bits 7 to 5): PLL reference clock frequency select

<1> Theses bits select the oscillation frequency of the main clock connected to the CF pins (CF1 and CF2).

<2> Either 8 MHz or 12 MHz can be selected if “Enable” is selected in the user option “Main clock 8MHz selection.” If “Disable” is selected, only 12 MHz is selected.

Main clock oscillation frequency selection

Main Clock 8MHz Selection	SELREF[2:0]	Main Clock Oscillation Frequency
Enable	000	8 MHz
	100	12 MHz
Disable	000	12 MHz
	100	12 MHz

* Any other settings than above are inhibited.

PLLTEST (bit 4): PLL test bit

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> Set both VCOSTP and CMPSTP to 0 when generating the 48 MHz clock for USB using the internal PLL. In this case, set bit 4 (P34DDR) of P3DDR (FE4DH) and bit 4 (P34) of P3 (FE4CH) to 0. In addition, it is necessary to connect the external filter circuit shown in Figure 3.21.3 to the P34/UFILT pin.

<2> Set these bits to 1 when the PLL is not to be used.

PLLCNT1 to PLLCNT0 (bits 1 to 0): Reserved bits

These bits must always be set to 0.

4.2.4.6 USB frequency divided clock control register (USBDIV) (8-bit register)

1) This register is an 8-bit register that is used to select the frequency of the USB frequency-divided clock, to control the external output of the USB frequency-divided clock, and to select the high-speed clock.

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE04	0000 0000	R/W	USBDIV	USBDIV7	DVCKON	DVCKDR	USBDIV4	USBDIV3	UDVSEL2	UDVSEL1	UDVSEL0

USBDIV7 (bit 7): Reserved bit

This bit must always be set to 0.

DVCKON (bit 6): High-speed clock select

<1> A 1 in this bit selects the USB frequency divided clock that is generated by frequency-dividing the 48 MHz clock for USB as the high-speed clock

<2> A 0 in this bit selects the main clock (CF) as the high-speed clock.

DVCKDR (bit 5): USB frequency-divided clock external output control

<1> This bit must be set to 1 to transmit the USB frequency-divided clock that is generated by frequency-dividing 48 MHz clock for USB from 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 configuration, the USB frequency-divided clock is transmitted from P73.

<2> Set this bit to 0 if the USB frequency-divided clock is not transmitted to any external device.

USBDIV4 to USBDIV3 (bits 4 to 3): Reserved bits

These bits must always be set to 0.

UDVSEL2 to UDVSEL0 (bits 2 to 0): USB frequency-divided clock frequency select

<1> These bits select the frequency of the divided clock of the 48 MHz clock for USB.

<2> Set the frequency divided clock to 8 to 12 MHz to run the USB interface controller circuit by supplying the USB frequency-divided clock as the system clock.

<3> To change the USB frequency-divided clock frequency from any value other than its initial value (UDVSEL＝000), temporarily set it to the "frequency-divided clock stopped" 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

USB Frequency-divided Clock Frequency Select

UDVSEL[2:0]	Frequency-divided Clock Frequency (MHz)
000	Frequency divided clock stopped
001	4.8
010	6
011	8
100	12
101	24 (inhibited)
110	4
111	16 (inhibited)

4.2.4.7 System clock divider control register (CLKDIV) (3-bit register)

1) This register is a 3-bit register that controls the frequency division operation 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 to CLKDV0 (bits 2 to 0):

These bits define the frequency division ratio of the system clock.

CLKDV2	CLKDV1	CLKDV0	Division Ratio
0	0	0	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

4.3 Standby Function

4.3.1 Overview

This series of microcontrollers supports three standby modes, i.e., HALT, HOLD, and X'tal HOLD modes, that are used to reduce current consumption at power-failure time or in program standby mode. In 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 (Some serial transfer functions are stopped.) (Note 1)
   - HALT mode is entered by setting bit 0 of the PCON register.
   - 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 oscillators are suspended. The microcontroller suspends the execution of instructions and its peripheral circuits stop processing. (Notes 1 and 2)
   - 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, port 0, or UHC bus active interrupt) occurs, bit 1 of the PCON register is cleared and the microcontroller switches into HALT mode.

3) X'tal HOLD mode
   - All oscillators except the subclock oscillation are suspended. (If, however, the base timer has been started with low-speed RC oscillator selected as the base timer input clock source, the low-speed RC oscillator circuit retains the state that is established when X'tal HOLD mode is entered.) The microcontroller suspends the execution of instructions and all the peripheral circuits except the base timer and the infrared remote control receiver circuit are suspended. (Notes 1 and 2)
   - 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 an X'tal HOLD mode release signal (INT0, INT1, INT2, INT4, INT5, port 0, base timer, infrared remote control receive, or UHC bus active interrupt) occurs, bit 1 of the PCON register is cleared and the microcontroller switches into HALT mode.

Note 1: The low-speed RC oscillation is also controlled directly by the watchdog timer. Its oscillation in the standby mode is also controlled by the watchdog timer. For more information, see Section 4.6, "Watchdog Timer (Internal Timer Type)."

Note 2: Do not allow the microcontroller to enter HOLD, or X'tal HOLD mode while AD conversion is in progress. Make sure that ADSTART (ADCRC register, bit 2) 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 used to specify 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.

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.

- When the microcontroller enters HOLD mode, all oscillators (main clock, subclock, low-/ medium-speed RC, and PLL) are suspended and the related registers are placed in the states described below.

  If the OCR3 register, bit 1 is se t to 1, the OCR3 register, bit 0 is set and the OCR register, bits 4 and 5 are cleared.

  If the OCR3 register, bit 1 is set to 0, the OCR register, bits 1, 4, and 5 are cleared.

  If the OCR3 register, bit 1 is set to 0, the OCR register, bits 1, 4, and 5 are cleared.
- When the microcontroller exits HOLD mode, the low- or medium-speed RC oscillator starts operation and is designated as the system clock source. The main clock, subclock, and PLL oscillators return to their state that was established before the microcontroller entered HOLD mode.
- When the microcontroller enters X'tal HOLD mode, all oscillators except the subclock (i.e., main clock, low-/medium-speed RC, and PLL) are suspended but the state of the OCR and OCR3 registers remains unchanged. If, however, the microcontroller enters X'tal HOLD mode with the low-speed RC oscillator selected as the clock source to the base timer, the low-speed RC oscillator retains the state that is established when X'tal HOLD is entered.
- Since it is impossible to secure the oscillation stabilization time for the main clock after the microcontroller exits X'tal HOLD mode, it is necessary to assign the system clock to either the subclock or low- or medium-speed RC oscillator clock when X'tal HOLD mode is entered.
- Since X'tal HOLD mode is used usually for low-current clock counting or when the infrared remote control receiver is in a standby state, less current will be consumed if the system clock is switched to the subclock or low-speed RC, and the main clock, medium-speed RC, and PLL oscillators are suspended before the microcontroller enters X'tal HOLD mode.

<2> XTIDLE must be cleared with an instruction.

<3> PDN is cleared when a HOLD mode release signal (INT0, INT1, INT2, INT4, INT5, port 0, base timer, infrared remote control receive, or UHC bus active interrupt) 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 also 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.

Note: The low-speed RC oscillation is also controlled directly by the watchdog timer. Its oscillation in the standby mode is also controlled by the watchdog timer. For more information, see Section 4.6, "Watchdog Timer (Internal Timer Type)."

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 POR/ LVD • 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 (APPENDIX-I).	• 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 4/3 are set to 0/1.	• WDT, bits 2 to 0 are cleared if WDT register (FE0F), bit 4 is set. • WDTCNT, bit 5 is cleared if WDTCNT register (FE79), bit 4/3 are set to 0/1. • PCON register, bit 0 is set. • OCR register, bits 5, 4, and 1 are cleared if OCR3 register, bit 1 is set to 0. • OCR3 register, bit 0 is set and OCR register, bits 5 and 4 are cleared if OCR3 register, bit 1 is set to 1.	• WDT, bits 2 to 0 are cleared if WDT register (FE0F), bit 4 is set. • WDTCNT, bit 5 is cleared if WDTCNT register (FE79), bit 4/3 are set to 0/1. • PCON register, bit 0 is set.
Main clock oscillation	Running	State established at entry time	Stopped	Stopped
Low-speed RC oscillation	Stopped	State established at entry time (Note 1)	Stopped (Note 1)	Stopped (Notes 1 and 2)
Medium-speed 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 • LVD: Undefined or data retained (depends on supply voltage) • Watchdog timer: Data retained	Data retained	Data retained	Data retained
Base timer	Stopped	State established at entry time	Stopped	State established at entry time
Infrared remote control receiver circuit	Stopped	State established at entry time	Stopped	State established at entry time
Peripheral modules except base timer and infrared remote control receiver circuit	Stopped	State established at entry time (Note 3)	Stopped	Stopped
Exit conditions	Entry conditions cancelled	• Interrupt request accepted • Reset/entry conditions established	• Interrupt request from INT0 to INT2, INT4, INT5, port 0, or UHC bus active generated • Reset/entry conditions established	• Interrupt request from INT0 to INT2, INT4, INT5, port 0, base timer, infrared remote control receiver circuit, or UHC bus active generated • Reset/entry conditions established
Returned mode	Normal mode	Normal mode (Note 4)	HALT mode (Note 4)	HALT mode(Note 4)
Data changed on exit	None	PCON register, bit 0 is cleared	PCON register, bit 1 is cleared	PCON register, bit 1 is cleared

Standby

Note 1: The low-speed RC oscillation is also controlled directly by the watchdog timer. Its oscillation in the standby mode is also controlled by the watchdog timer. For more information, see Section 4.6, "Watchdog Timer (Internal Timer Type)."

Note 2: If X'tal HOLD mode is entered with the low-speed RC oscillator selected as the base timer input clock source, and the base timer has been started, the low-speed RC oscillator circuit retains the state that is established when X'tal HOLD mode is entered.

Note 3: Some functions of serial transfer and USB interface control circuit are stopped.

Note 4: The microcontroller switches into the reset state if it exits the current mode on the establishment of reset/entry conditions.

Table 4.3.2 Pin States and Operating Modes (This Series)

Pin Name	Reset Time	Normal Mode	HALT Mode	HOLD Mode	On Exit from HOLD
$\overline{RES}$	• Input *Low level is generated when internal reset circuit is active.	• 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 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.	• State established on entry into HOLD mode
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 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.	• State established on entry into HOLD mode
CF1	• CF oscillator inverter input • Oscillation enabled • Feedback resistor present between CF1 and CF2.	• CF oscillator inverter input • Enabled/disabled by register OCR (FE0EH) • Feedback resistor present between CF1 and CF2.	←	• CF oscillator inverter input • Oscillation suspended • Feedback resistor present between CF1 and CF2.	• State established on entry into HOLD mode
CF2	• CF oscillator inverter output • Oscillation enabled	• CF oscillator inverter output • Enabled/disabled by 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	• State established on entry into HOLD mode

Continued on next page.

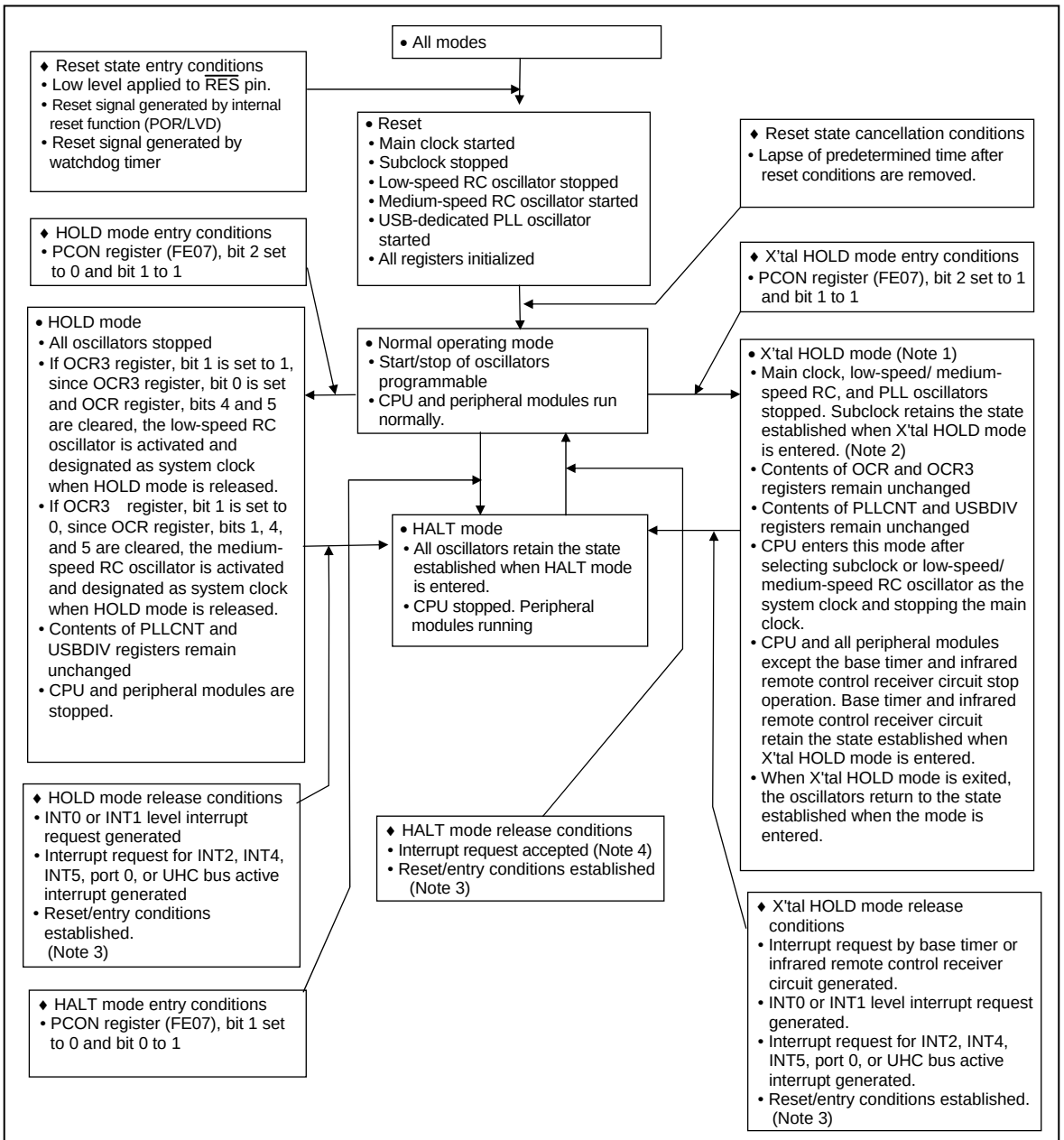
Pin States and Operating Modes (continued)

Pin Name	Reset Time	Normal Mode	HALT Mode	HOLD Mode	On Exit from HOLD
P00 to P01 P03 to P07	• Input mode • Pull-up resistor off	• Input/output/pull-up resistor controlled by a program	←	←	←
P02	• Low level output • Pull-up resistor off	• Input/output/pull-up resistor controlled by a program	←	←	←
P10 to P17	• Input mode • Pull-up resistor off	• Input/output/pull-up resistor controlled by a program	←	←	←
P20 to P25	• Input mode • Pull-up resistor off	• Input/output/pull-up resistor controlled by a program	←	←	←
P30 to P32	• Input mode • Pull-up resistor off	• Input/output/pull-up resistor controlled by a program	←	←	←
P33	• Input mode • Pull-up resistor off •Audio-dedicated PLL oscillation disabled	• Input/output/pull-up resistor controlled by a program • Audio-dedicated PLL oscillation enabled/ disabled controlled by a program • (Note 1)	← (Note 1)	• nput/output/pull-up resistor controlled by a program • Audio-dedicated PLL oscillation disabled	• State established on entry into HOLD mode (Note 1)
P34	• Pull-up resistor off • USB-dedicated PLL oscillation enabled (Note 2)	• Input/output/pull-up resistor controlled by a program • USB-dedicated PLL oscillation enabled/ disabled controlled by a program (Note 2)	← (Note 2)	• Input/output/pull-up resistor controlled by a program • USB-dedicated PLL oscillation disabled	• State established on entry into HOLD mode (Note 2)
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 (since on-time is automatically extended, it takes 1920 to 2048 Tcyc for the transistor to go off).	• 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	←	←	←
PWM0 PWM1	• Input mode	• Input/output controlled by a program	←	←	←
UHAD- UHAD+	• Input mode	• Input/output controlled by a program when used as a general- purpose port.	←	←	←
UHBD- UHBD+	• Input mode	• Input/output controlled by a program when used as a general- purpose port.	←	←	←

Note 1: PLL frequency adjustment pulses are output through the P33/AFILT pin in audio-dedicated PLL oscillation mode.

Note 2: PLL frequency adjustment pulses are output through the P34/UFILT pin in USB-dedicated PLL oscillation mode

Standby



Note 1: The low-speed RC oscillation is also controlled directly by the watchdog timer. Its oscillation in the standby mode is also controlled by the watchdog timer. For more information, see Section 4.6, "Watchdog Timer (Internal Timer Type)."

Note 2: If X'tal HOLD mode is entered with the low-speed RC oscillator selected as the base timer input clock source and the base timer has been started, the low-speed RC oscillator circuit retains the state that is established when X'tal HOLD mode is entered.

Note 3: The CPU enters the reset state when the reset/entry conditions are established.

Note 4: 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)

Figure 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 four types of reset functions:

1) External reset via the $\overline{\text{RES}}$ pin
   - The microcontroller is reset without fail by applying 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 external time constant element.

2) Internal reset
   - The internal reset function is available in two types: the power-on reset (POR) that triggers a reset when power is turned on, and the low-voltage detection reset (LVD) that triggers a reset when the power voltage falls below a certain level.
   - Options are available to set the power-on reset release level, to enable (use) and disable (non-use) the low-voltage detection reset function, and to set its threshold level.

3) Runaway detection reset function by the external RC time constant type watchdog timer
   - The runaway detection reset function can be implemented by connecting external resistors and capactors to the external interrupt pin (P70/INT0/T0LCP) and creating an appropriate time constant value.

4) Runaway detection reset function by the internal timer type watchdog timer
   - Reset signals can be generated at constant intervals using the internal low-speed RC oscillator clock or subclock.

An example of a reset circuit is shown in Figure 4.4.1. The external circuit connected to the reset pin shows an example that LVD reset function is disabled and an external power-on reset circuit is configured.

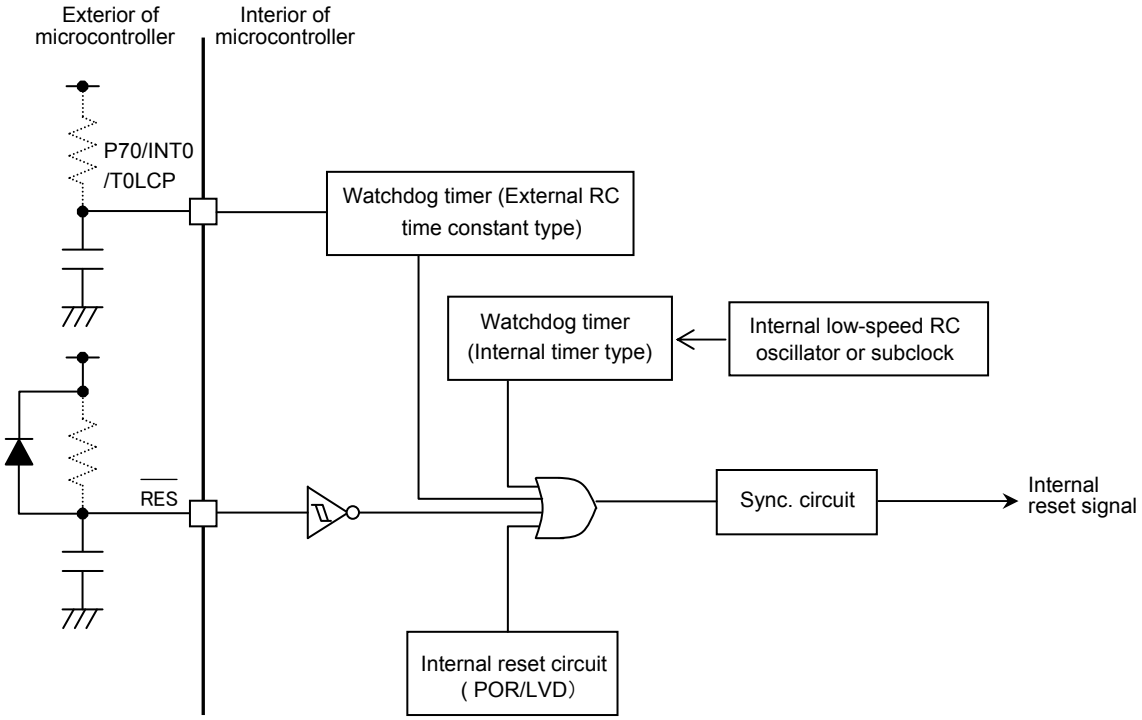


Fig. 4.4.1 Reset Circuit Block Diagram

4.4.3 Reset State

When a reset is generated by the $\overline{\text{RES}}$ pin, internal reset circuit, 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 medium-speed RC oscillator when a reset occurs, hardware initialization is also carried out immediately even at power-on time. To use the main clock subsequently, 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 by a user option. The special function registers (SFRs) are also initialized to the values that are listed in Appendix (A-I), Special Function Register (SFR) Map.

<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.*
- *When using the internal reset function, it is necessary to implement and connect an external circuit to the reset pin according to the user's operating environment. Be sure to review and observe the operating specifications, circuit configuration, precautions, and considerations discussed in Section 4.7, "Internal Reset Function."*

4.5 Watchdog Timer (External RC Time Constant Type)

4.5.1 Overview

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 a high level, it triggers a reset or interrupt, regarding that a program runaway condition occurred.

4.5.2 Functions

1) Detection of a runaway condition

   A program for discharging the RC circuit periodically needs to be prepared. If a program runaway occurs, 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 re-execution)
   - External interrupt INT0 generation (continue program execution)

     The priority level of the external interrupt INT0 can be changed using the master interrupt enable control register (IE).

4.5.3 Circuit Configuration

The watchdog timer consists 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.

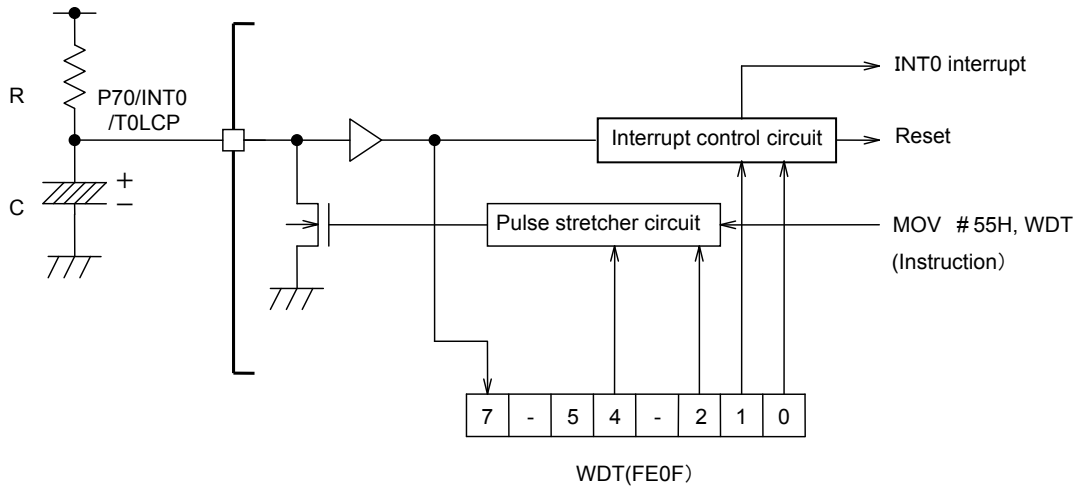


Fig. 4.5.1 Watchdog Timer Circuit (External RC Time Constant Type)

4.5.4 Related Registers

4.5.4.1 Watchdog timer control register (WDT)

1) This register controls the operation of the watchdog timer.

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 the watchdog timer. 1: Enables clearing the watchdog timer.
WDTRST (bit 1)	Runaway-time reset control
	0: Disables reset when runaway is detected. 1: Triggers reset when runaway is detected.
WDTRUN (bit 0)	Watchdog timer operation control
	0: Maintains watchdog timer operating state. 1: Starts watchdog timer operation.

WDTFLG (bit 7): Runaway detection flag

This bit is set when a program runaway is detected by the watchdog timer. The application can identify the occurrence of a program runaway by monitoring this bit (provided that WDTRST is set to 1).

This bit is not reset automatically. It must be reset with an instruction.

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 HALT or HOLD state. When this bit is set to 1, WDTCLR, WDTRST, and WDTRUN are reset and the watchdog timer is stopped in 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 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 external capacitor to be discharged while the watchdog timer is running (WDTRUN=1). If the instruction for clearing the watchdog timer is executed when this bit is set to 1, the N-channel transistor of the P70/INT0/T0LCP pin turns on, the external capacitor is discharged, and the watchdog timer is cleared. The pulse stretcher circuit also functions during this process. Setting the bit to 0 disables turning on the N-channel transistor of the P70/INT0/T0LCP pin and clearing the watchdog timer.

Also, if this bit is set to 1 when the watchdog timer is stopped (WDTRUN=0), the N-channel transistor of the P70/INT0/T0LCP pin is turned on, the external capacitor is discharged, and the watchdog timer is cleared.

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. If this bit is set to 1, reset is triggered when a program runaway condition is detected, and the microcontroller re-executes the program starting at the program start address which is selected by a user option. If this bit is set to 0, no reset occurs when a program runaway is detected. 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 retains (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, the watchdog timer cannot by stopped by software (stopped by a reset).

Caution

If WDTRST is set to 1, a reset is triggered when pin P70/INT0/TOLCP goes to the high level even if the watchdog timer is not running.

The N-channel transistor of 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 stopped (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)

1) See Subsubsection 4.1.4.1 “Master interrupt t enable control register,” for details.

4.5.4.3 Port 7 control register (P7)

1) 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 a low level is applied to the external $\overline{\text{RES}}$ pin, when a reset is generated by the internal reset (POR/LVD) function, or when a reset is generated by the internal timer type watchdog timer.

   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 output N-channel transistor at the P70/INT0/T0LCP pin to start discharging the capacitor.

   • Checking the low level

     Check 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 WDTRUN is set to 1. Once the watchdog timer starts operation, the watchdog timer control register (WDT) is disabled for write; it is only possible 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.

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. Owing 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 by 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.

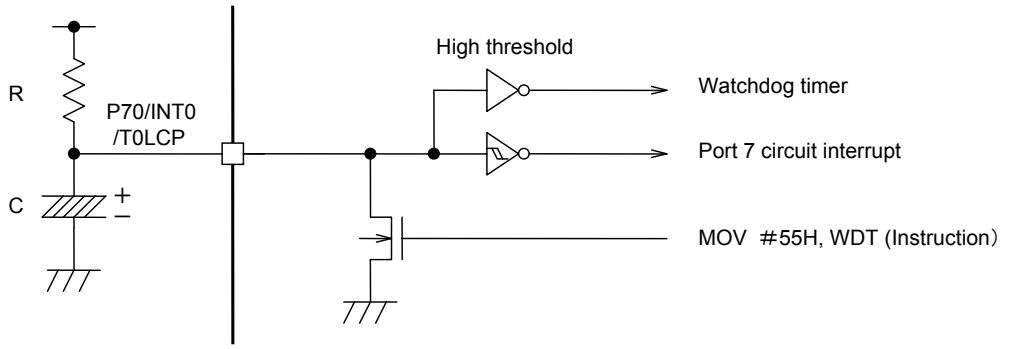


Fig. 4.5.2 P70/INT0/T0LCP Pin Configuration (Option: Pull-up Resistor OFF)

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 programmable pull-up resistor to the P70/INT0/T0LCP pin (see Figure 4.5.3).

The resistance of the pull-up resistor to be adopted in this case varies according to the power source voltage VDD. Check the pull-up resistance value by referring to the latest "SANYO Semiconductors Data Sheet" and calculate the time constant of the watchdog timer.

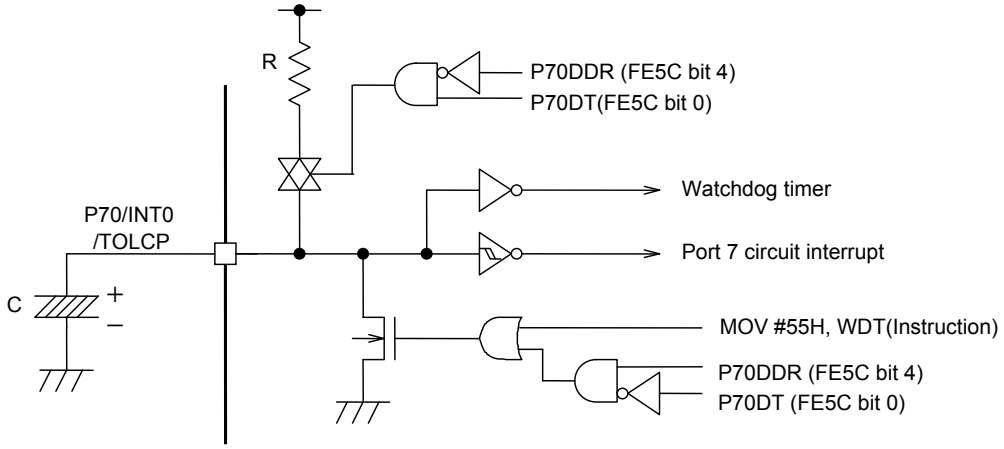


Fig. 4.5.3 Sample Application Circuit with a Pull-up Resistor

4) When the microcontroller enters HALT or HOLD mode with WDTHLT set to 1, bits WDTCLR, WDTRST, and WDTRUN are reset. To use the watchdog timer function when 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.

4.6 Watchdog Timer (Internal Timer Type)

4.6.1 Overview

This series of microcontrollers is provided with a watchdog timer (internal timer type) that has the following functions:

1) Capable of generating an internal reset on an overflow of a timer that runs on an internal low-speed RC oscillator clock or subclock.
2) Operation when the microcontroller enters standby mode can be selected from three modes (continue count operation, suspend operation, and suspend count operation while retaining the count value).

4.6.2 Functions

1) Watchdog timer function
   - A 17-bit up-counter (WDTCT) runs on the WDT clock (the clock source is selected from either the internal low-speed RC oscillator clock or subclock). A WDT reset (internal reset) signal is generated when the overflow time (selected from 8 time values) that is selected by the watchdog timer control register (WDTCNT) is reached. At this time, the WDT reset detection flag (WDTRSTF) is set. Since the WDTCT can be cleared by a program, it is necessary to code a program so that the WDTCT can be cleared at constant intervals.
   - If the WDT operation is started with the internal low-speed RC oscillator clock selected as the WDT clock source, since the internal low-speed RC oscillator circuit is controlled independently, even if the system clock happens to be suspended by a program runaway condition, the WDT continues operation, making it is possible to detect the runaway condition.
   - If the WDT operation is started when the subclock is selected as the WDT clock, a WDT reset is generated on detection of a subclock oscillation suspended by the XT function control bit (EXTOSC) of the oscillation control register (OCR) or on entry into HOLD mode. In this case, WDTRSTF is set.

2) Standby mode time operations
   - The action that the WDT takes in standby mode can be selected from three operating modes: continue count operation, suspend operation, and suspend count operation while retaining the count value. If the internal low-speed RC oscillator clock is selected as the WDT clock source when continue count operation is selected, an operating current of several μA is always flowing in the IC even when it is in standby mode because the internal low-speed RC oscillator circuit is continuing oscillation (For details, refer to the latest "SANYO Semiconductors Data Sheet.").

3) It is necessary to manipulate the following special function register to control the watchdog timer (internal timer type).
   - WDTCNT

Address	Initial Value	R/W	Name	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE79	0000 0000	R/W	WDTCNT	WDTRSTF	WDTCKSL	WDTRUN	IDLOP1	IDLOP0	WDTSL2	WDTSL1	WDTSL0

4.6.3 Circuit Configuration

4.6.3.1 WDT control register (WDTCNT) (8-bit register)

1) This register is used to manipulate the WDT reset detection flag, to select operation in standby mode, to select the overflow time, and to control the operation of WDT.

Note: WDTCNT is initialized to 00H when a low level is applied to the external $\overline{\text{RES}}$ pin, when a reset is generated by the internal reset (POR/LVD) function, or when a reset is generated by the external RC time constant type watchdog timer. Bit 6 and bits 4 to 0 of WDTCNT are not initialized when a reset is generated by the internal timer type watchdog timer.

*Note: The WDTCNT is disabled for writes once WDT operation is started (WDTRUN set to 1). If the instruction "**MOV #55H, WDTCNT**" is executed in this case, the WDTCT is cleared and count operation is restarted at a count value of 0 (the WDTCT is not cleared when it is loaded with 55H by any other instruction).*

Note: The internal low-speed RC oscillator circuit is started by setting WDTCKSL (WDTCNT, bit 6) to 0 and WDTRUN (WDTCNT, bit 5) to 1. Once the oscillator starts oscillation, an operating current of several μA flows. (For details, refer to the latest "SANYO Semiconductors Data Sheet"). Note that oscillation can also be started by setting SRCSTART (OCR3, bit 0) to 1.

4.6.3.2 WDT counter (WDTCT) (17-bit counter)

1) Operation start/stop: Start/stop is controlled by the 1/0 value of WDTRUN. When WDTRUN is set to 1 and IDLOP1 and IDLOP0 (WDTCNT, bits 4 and 3) are set to 2, the microcontroller enters standby mode.

2) Count clock: The WDT clock (selected from the internal low-speed RC oscillator clock or subclock).

3) Overflow: Generated when the WDTCT count value matches the count value designated by WDTSL2 to WDTSL0 (WDTCNT, bits 2 to 0)

   * Generates the WDT reset signal, WDTRUN clear signal, and set signal to WDTRSTF (WDTCNT, bit 7).

4) Reset: Setting WDTRUN to 0, or setting WDTRUN to 1 and executing the **MOV #55H, WDTCNT** instruction.

* See Figure 4.6.2 for details on WDT operation.

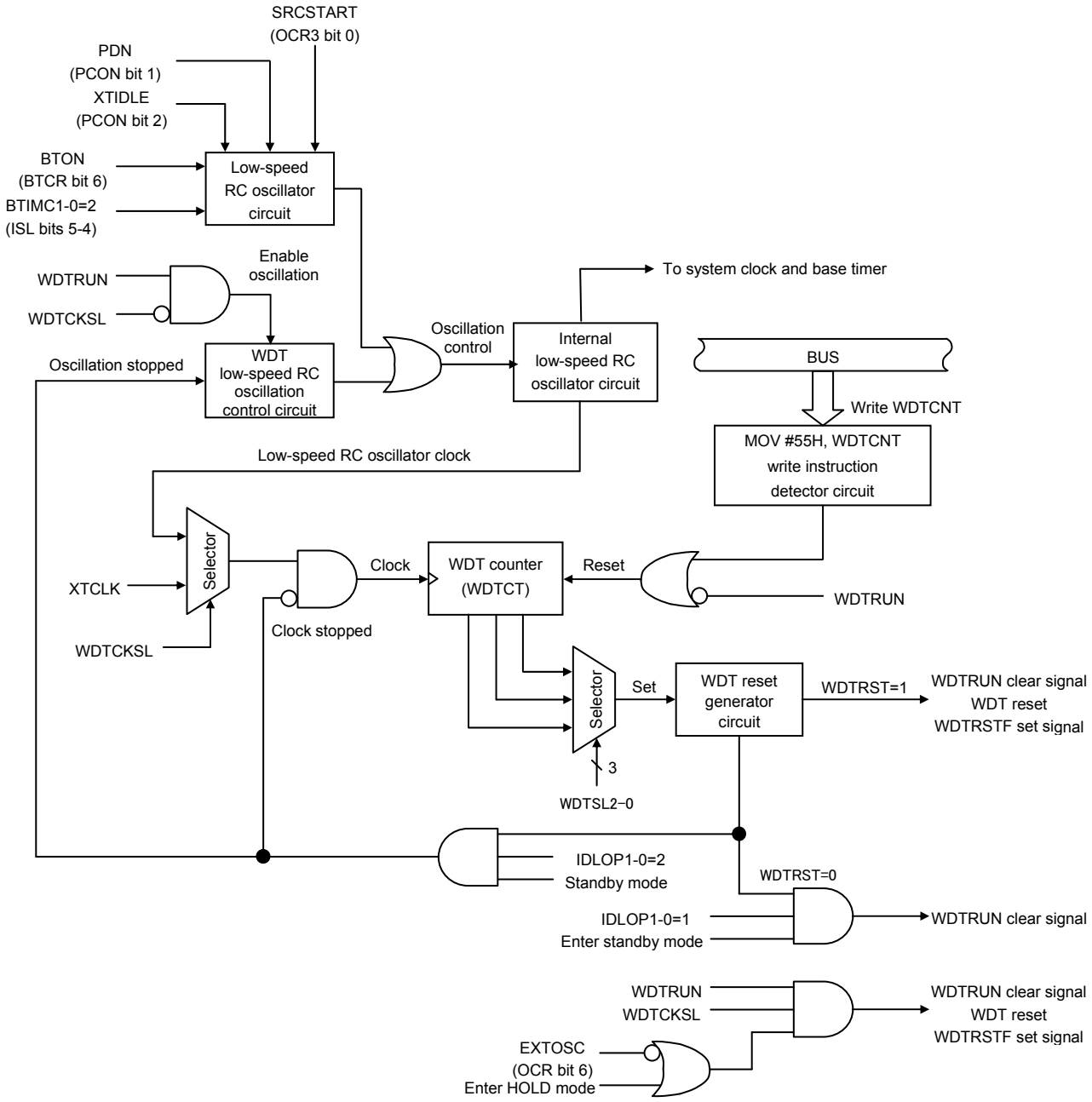


Figure 4.6.1 Watchdog Timer Block Diagram

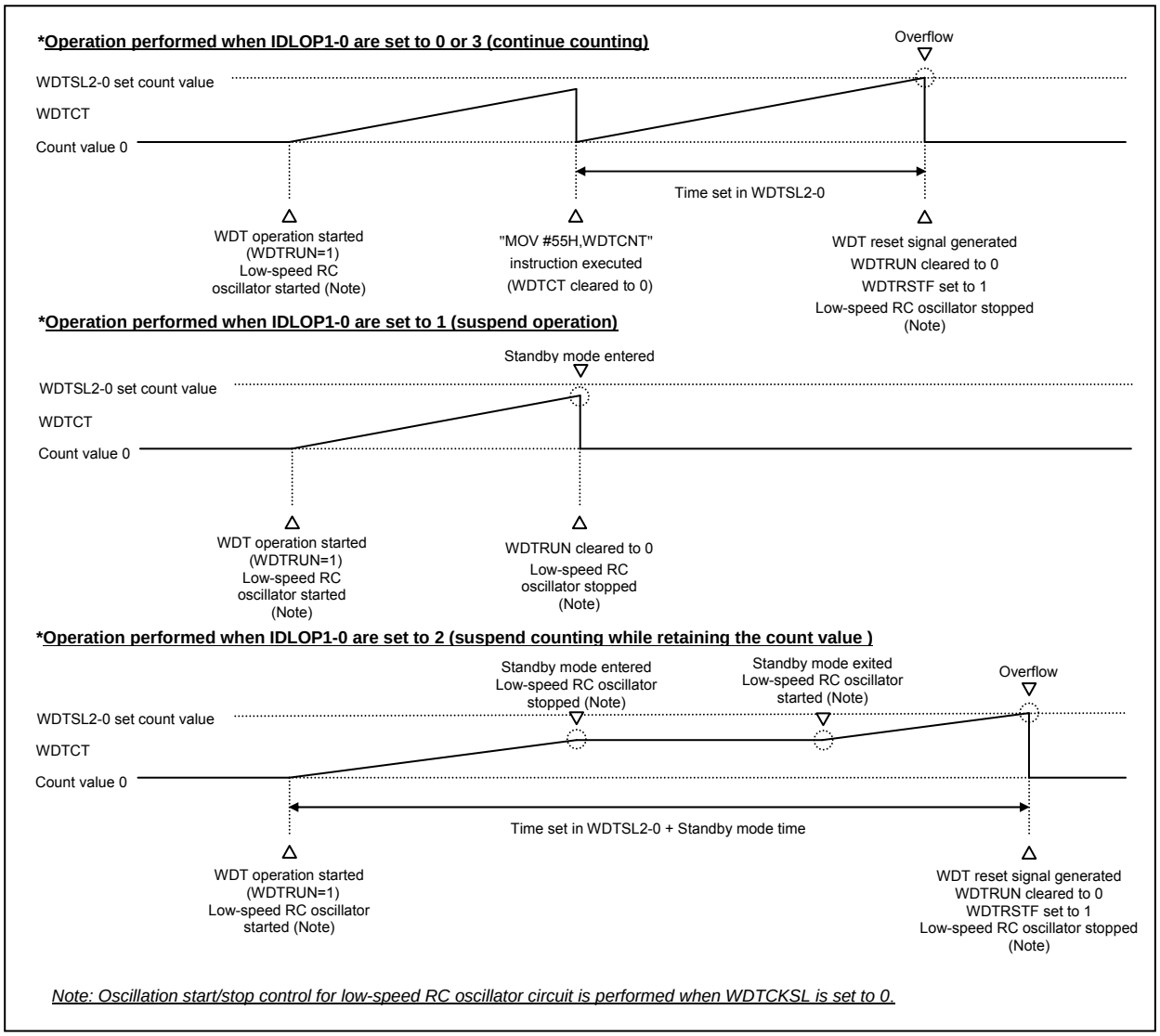


Figure 4.6.2 Sample Watchdog Timer Operation Waveforms

4.6.4 Related Register

4.6.4.1 WDT control register (WDTCNT)

1) This register is used to manipulate the WDT 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	WDTRSTF	WDTCKSL	WDTRUN	IDLOP1	IDLOP0	WDTSL2	WDTSL1	WDTSL0

WDTRSTF (bit 7): WDT reset detection flag

This bit is cleared when a reset is triggered by applying a low level signal to the external $\overline{\text{RES}}$ pin or by using the internal reset (POR/LVD) function.

This bit is set when a WDT-triggered reset occurs.

This flag can be rewritten with an instruction.

WDTCKSL (bit 6): WDTCT input clock select

WDTCKSL	WDTCT Input Clock
0	Internal low-speed RC oscillator clock
1	Subclock

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 select**

IDLOP1	IDLOP0	WDT Standby Mode Operation
0	0	Continue count operation
0	1	Suspend operation
1	0	Suspend count operation while retaining the count value
1	1	Continue count operation

* See “Figure 4.6.2” for details of the WDT operating modes.

WDTSL2 (bit 2):
WDTSL1 (bit 1):
WDTSL0 (bit 0): } **Overflow time select**

WDTSL2	WDTSL1	WDTSL0	WDTCT Set Count Value and Overflow Generation Time Example		
			Count Value	Low-speed RC Clock	Subclock
0	0	0	1024	34.1ms	31.25ms
0	0	1	2048	68.3ms	62.50ms
0	1	0	4096	137ms	125.0ms
0	1	1	8192	273ms	250.0ms
1	0	0	16384	546ms	500.0ms
1	0	1	32768	1.09s	1.000s
1	1	0	65536	2.18s	2.000s
1	1	1	131072	4.37s	4.000s

* Time values in the “Low-speed RC Clock” column of the table refer to the time for a WDTCT overflow to occur when the internal low-speed RC oscillation frequency is 30 kHz (typ). The low-speed RC oscillation frequency varies from IC to IC. For details, refer to the latest "SANYO Semiconductors Data Sheet."
* Time values in the “Subclock” column of the table refer to the time for a WDTCT overflow to occur when the 32.768 kHz X’tal oscillator is used.

Note: The WDTCNT is initialized to "00H" when a low-level signal is applied to the external $\overline{RES}$ pin or when a reset is triggered by the internal reset (POR/LVD) function or by the watchdog timer of external RC time constant type. Bit 6 and bits 4 to 0 of WDTCNT are not initialized, however, when a reset is triggered by the watchdog timer of internal timer type.

*Note: The WDTCNT is disabled for write once the WDT starts operation (WDTRUN set to 1). If the instruction **"MOV #55H, WDTCNT"** is executed in this case, the WDTCT is cleared and count operation is restarted at a count value of 0 (the WDTCT is not cleared when it is loaded with 55H with any other instruction).*

Note: The internal low-speed RC oscillator circuit is started by setting WDTCKSL to 0 and WDTRUN to 1. Once the oscillator starts oscillation, an operating current of several μA flows (For details, refer to the latest "SANYO Semiconductors Data Sheet"). Note that oscillation is also started by setting SRCSTART (OCR3, bit 0) to 1.

4.6.5 Using the Watchdog Timer

Code a program so that instructions for clearing the watchdog timer periodically are executed.

1) Starting the watchdog timer

<1> Set the time for a WDT reset to occur to WDTCKSL (WDTCNT, bit 6) and WDTSL2 to WDTSL0 (WDTCNT, bits 2 to 0).

<2> Set the WDT standby mode operation (HALT/HOLD/X'tal HOLD) to IDLOP1 to 0 (WDTCNT, bits 4 to 3).

<3> After <1> and <2>, set WDTRUN (WDTCNT, bit 5) to 1.

The watchdog timer starts functioning when WDTRUN is set to 1. Once the watchdog timer starts operation, WDTCNT is disabled for write; it is only possible to clear WDTCT and read WDTCNT. Consequently, the watchdog timer can never be stopped with an instruction. The function of the watchdog timer is stopped only when a low level is applied to the external $\overline{\text{RES}}$ pin, a reset by the internal reset function (POR/LVD) occurs, a reset by the watchdog timer of external RC time constant type occurs, or standby mode is entered when IDLOP1 to IDLOP0 are set to 1. In this case, WDTRUN is cleared.

2) Clearing the WDTCT

When the watchdog timer starts operation, WDTCT counts up. When this WDTCT overflows, a WDT reset occurs. To run the program in normal mode, it is necessary to periodically clear WDTCT before WDTCT overflows. Execute the following instruction to clear WDTCT while it is running:

MOV #55H, WDTCNT

3) Detecting a runaway condition

Unless the above-mentioned instruction is executed periodically, WDTCT overflows because the watchdog timer is not cleared. If an overflow occurs, the watchdog timer considers that a program runaway has occurred and triggers a WDT reset. In this case, WDTRSTF (WDTCNT, bit 7) is set. After a WDT reset occurs, the program execution restarts at address 0000H. (In the flash ROM version, the program execution restarts at the address selected as an user option.)

4.6.6 Notes on the Use of the Watchdog Timer

1) When the internal low-speed RC oscillator clock is selected as the WDT clock (WDTCKSL = 0)

- If the internal low-speed RC oscillator clock is not to be used as the system clock, set SRCSTART (OCR3, bit 0) to 0 (the start/stop of the internal low-speed RC oscillator circuit is also controlled from the watchdog timer side). If SRCSTART (OCR3, bit 0) is set to1, the internal low-speed RC oscillator circuit continues oscillation in HALT mode even though the watchdog timer is running with IDLOP1 and IDLOP0 set to 1 or 2.
- To realize ultra-low-power operation using HOLD mode, it is necessary to disable the watchdog timer from running in HOLD mode by setting IDLOP1 and IDLOP1 to 1 or 2. When setting IDLOP1 and IDLOP0 to 0 or 3, several μA of operating current flows at all times because the low-speed RC oscillator circuit continues oscillating even in HOLD mode.
- If standby mode is entered when the watchdog timer is running with IDLOP0 and IDLOP0 set to 2, the internal low-speed RC oscillator circuit stops oscillation and the watchdog timer stops count operation and retains the count value. When the CPU subsequently exits standby mode, the low-speed RC oscillator circuit resumes oscillation and the watchdog timer starts count operation. If the period from the release of standby mode to the next entry into standby mode is less than "low-speed RC oscillator clock × 4," however, the low-speed RC oscillator circuit may not stop oscillation when the CPU enters standby mode. In such a case (standby mode is on), several μA of operating current flows because the low-speed RC oscillator circuit is active though the count operation of the watchdog timer is stopped.

  To minimize the standby power requirement of the set, code the program so that an interval of "low-speed RC oscillator clock × 4" or longer is provided between the release from standby mode and entry into the next standby mode. Note that the oscillation frequency of the low-speed RC oscillator may fluctuate. See the latest "SANYO Semiconductors Data Sheet" for details.

2) When the subclock is selected as the WDT clock (WDTCKSL = 1)

- When the watchdog timer is used with WDTCKSL set to 1, set EXTOSC (OCR, bit 6) to 1 and start the watchdog timer operation with a program control allowing the subclock oscillator to be stabilized.
- On the detection of subclock oscillation suspended by EXTOSC (OCR, bit 6) being set to 0 or HOLD mode being entered when the watchdog timer is running, the watchdog timer considers that a program runaway has occurred and triggers a WDT reset. In this case, WDTRSTF is set.

* This mode is primarily used for applications using the real-time clock to realize low-power operation.

4.7 Internal Reset Function

4.7.1 Overview

This series of microcontroller incorporates internal reset functions called the power-on reset (POR) and low voltage detection reset (LVD). The use of these functions contributes to a reduction in the number of externally required reset circuit components (reset IC, etc.).

4.7.2 Functions

1) Power-on reset (POR) function

   POR is a hardware feature that generates a reset to the microcontroller when the power is turned on. This function allows the user to select the POR release level by option only when the low voltage detection reset function is set to disable. It is necessary to use the below-mentioned low voltage detection reset function together with this function, or configure an external reset circuit if there are possibilities that chatter may occur or momentary power loss may occur when the power is turned on.

2) Low voltage detection reset (LVD) function

   This function, when used together with the POR function, can generate a reset when power is turned on and when the power level lowers. As a user option the use (enable) or non-use (disable) and the detection levels of this function can be specified.

4.7.3 Circuit Configuration

The internal reset circuit consists of the POR, LVD, pulse stretcher circuit, capacitor C_{RES} discharging transistor, external capacitor C_{RES} + pull-up resistor R_{RES} or pull-up resistor R_{RES} alone. The circuit diagram of the internal reset circuit is provided in Figure 4.7.1.

- Pulse stretcher circuit

  The pulse stretcher circuit stretches the POR and LVD reset signals. It is used to stretch the internal reset period and discharge the external capacitor C_{RES} connected to the reset pin. The stretching time lasts from 30 μs to 100 μs.

- Capacitor C_{RES} discharging transistor

  This is an N-channel transistor used to discharge the external capacitor C_{RES} connected to the reset pin. If the capacitor C_{RES} is not to be connected to the reset pin, it is possible to monitor the internal reset signal by connecting only the external pull-up resistor R_{RES}.

- Option selector circuit

  The option selector circuit is used to configure the LVD options. This circuit selects whether to use (enable) or non-use (disable) the LVD and selects its detection levels. See Subsection 4.7.4.

- External capacitor C_{RES} + Pull-up resistor R_{RES}

  After the reset signal from the internal reset circuit is released, the reset period is further stretched according to the external CR time constant. This enables the microcontroller to avoid repetitive entries and releases of the reset state from occurring when power-on chatter occurs. The circuit configuration shown in Figure 4.7.1, in which the capacitor C_{RES} and pull-up resistor R_{RES} are externally connected, is recommended when both POR and LVD functions are to be used. The recommended constant values are: C_{RES} = 0.022 μF and R_{RES} = 510 kΩ. The external pull-up resistor R_{RES} must always be installed even when the set's specifications inhibit the installation of the external capacitor C_{RES}.

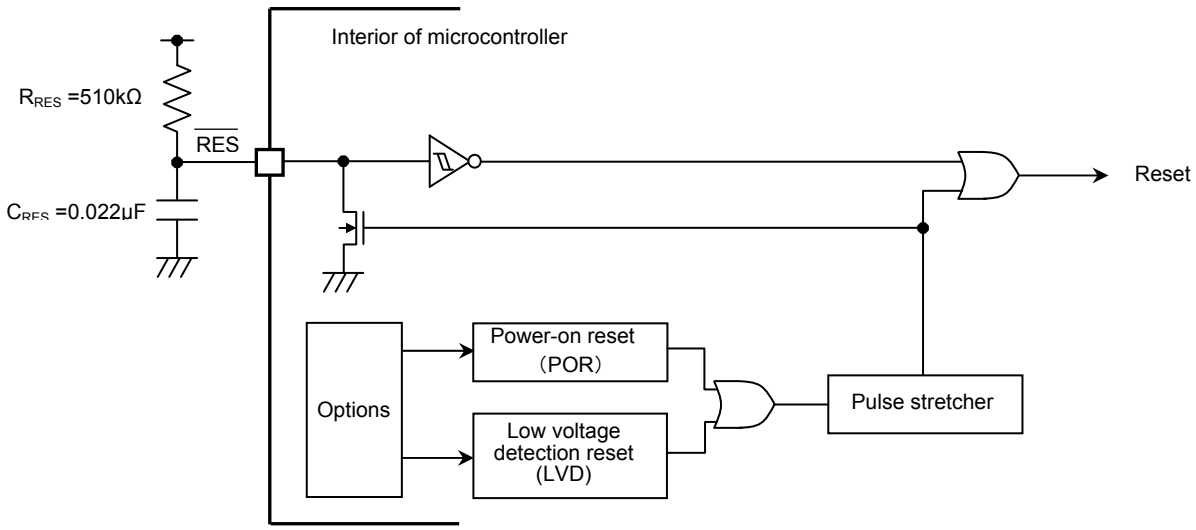


Figure 4.7.1 Internal Reset Circuit Configuration

4.7.4 Options

The POR and LVD options are available for the reset circuit.

<1> LVD Reset Function Options			
Enable: Use		Disable: Non-use	
<2> LVD Reset Level Option		<3> POR Release Level Option	
Typical Value of Selected Option	Min. Operating VDD Value (*)	Typical Value of Selected Option	Min. Operating VDD Value (*)
–	–	"1.67V"	1.8V -
"1.91V"	2.1V -	"1.97V"	2.1V -
"2.01V"	2.2V -	"2.07V"	2.2V -
"2.31V"	2.5V -	"2.37V"	2.5V -
"2.51V"	2.7V -	"2.57V"	2.7V -
"2.81V"	3.0V -	"2.87V"	3.0V -
"3.79V"	4.0V -	"3.86V"	4.0V -
"4.28V"	4.5V -	"4.35V"	4.5V -

* The minimum operating VDD value specifies the approximate lower limit value of the VDD value beyond which the selected POR release level or LVD reset level can not be effected without generating a reset.

<1> LVD reset function option

When “Enable” is selected, a reset is generated at the voltage that is selected by the LVD reset level option.

Note 1: In this configuration, an operating current of several μA always flows in all modes.

No LVD reset is generated when "Disable" is selected.

Note 2: In this configuration, no operating current will flow in all modes.

* See the sample operating waveforms of the reset circuit shown in Subsection 4.7.5 for details.

<2> LVD reset level option

The LVD reset level can be selected from 7 levels only when the LVD reset function is enabled. Select the appropriate detection level according to the user's operating conditions.

<3> POR release level option

The POR release level can be selected from 8 levels only when the LVD reset function is disabled. When not using the internal reset circuit, set the POR release level to the lowest level (1.67V) that will not affect the minimum guaranteed operating voltage.

Note 3: No operating current flows when the POR reset state is released.

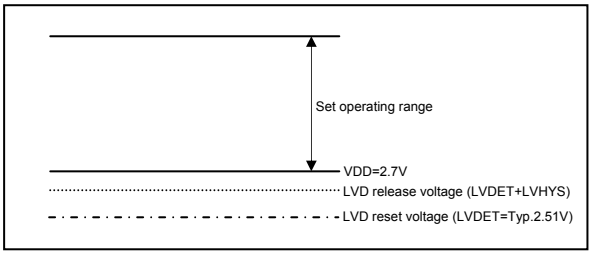
Note 4: See the notes in paragraph 2) of Subsection 4.7.6 when selecting a POR release level that is lower than the minimum guaranteed operating voltage (1.67V).

Internal Reset

- **Selection example 1**

  Selecting the optimum LVD reset level to keep the microcontroller running without resetting it until VDD falls below 2.7 according to the set's requirements

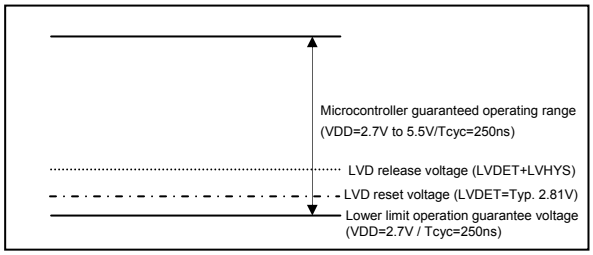
  Set the LVD reset function option to "Enable" and select "2.51V" as the LVD reset level.



- **Selection example 2**

  Selecting the optimum LVD reset level that meets the guaranteed operating conditions of VDD=2.7V/ Tcyc=250 ns

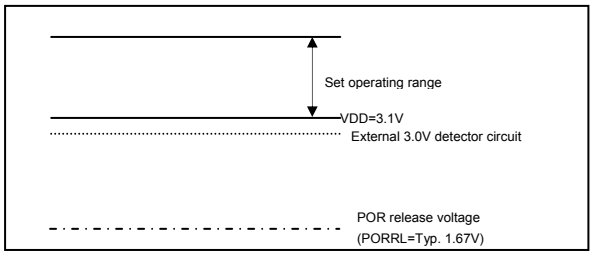
  Set the LVD reset function option to "Enable" and select "2.81V" as the LVD reset level option.



- **Selection example 3**

  Disabling the internal reset circuit and using an external reset IC that can detect and react at 3.0V (see also paragraph 1) of Subsection 4.7.7)

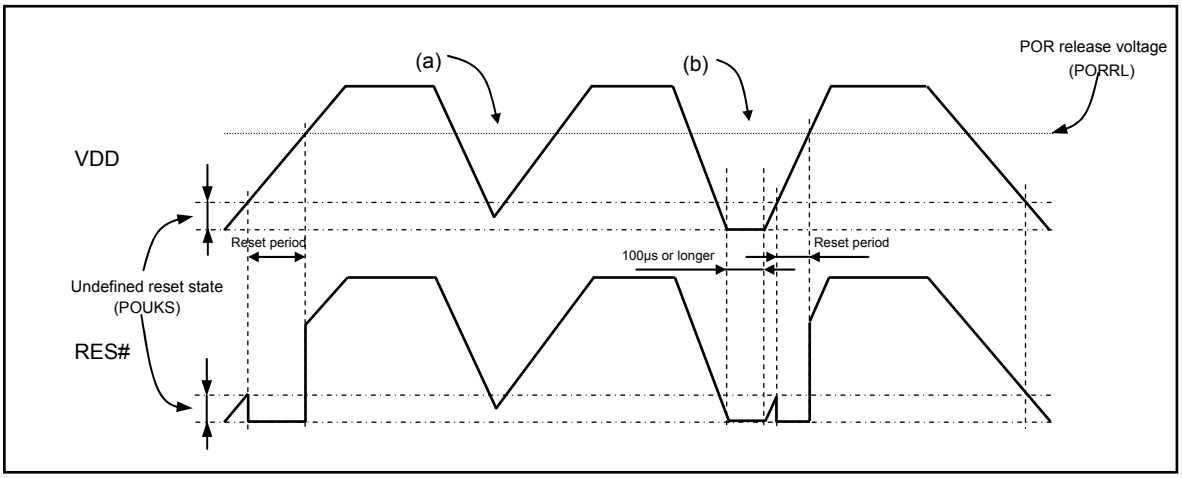
  Set the LVD reset function option to "Disable" and select "1.67V" as the POR release level option.



Note 5: The operation guarantee values (voltage/operating frequency) shown in the examples vary with the microcontroller type. Be sure to see the latest "SANYO Semiconductors Data Sheet" and select the appropriate setting level."

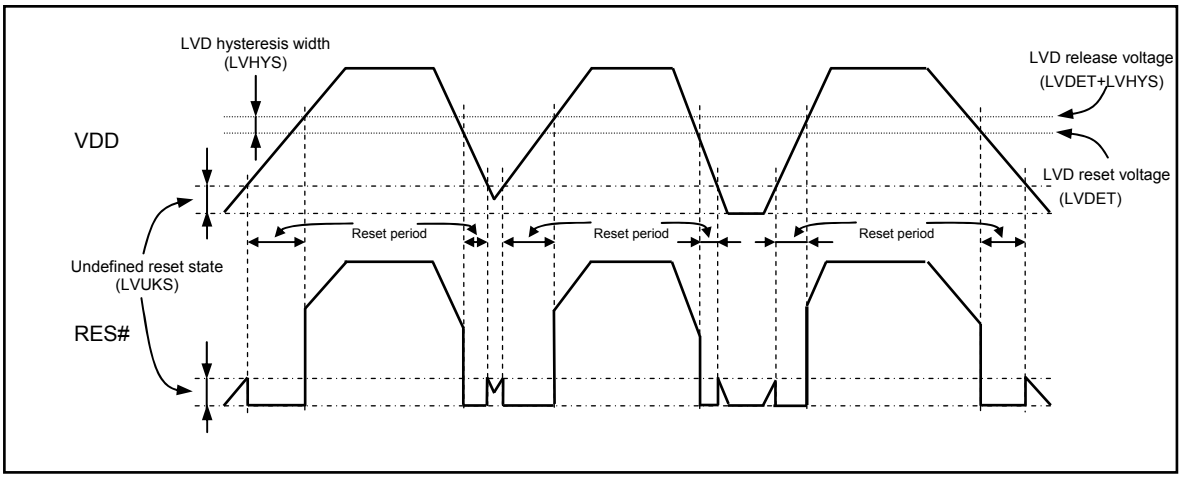
4.7.5 Sample Operating Waveforms of the Internal Reset Circuit

1) Waveform observed when only POR is used (LVD not used)
(Reset pin: Pull-up resistor R_{RES} only)



- There exists an undefined state (POUKS) before the POR transistor starts functioning normally.
- The POR function generates a reset only when the power is turned on starting at the VSS level. The reset release voltage in this case may have some range. Refer to the latest "SANYO Semiconductors Data Sheet" for details.
- No stable reset will be generated if power is turned on again when the power level does not go down to the VSS level as shown in (a). If such a case is anticipated, use the LVD function together with the POR function as explained in 2) or implement an external reset circuit.
- A reset is generated only when the power level goes down to the VSS level as shown in (b) and power is turned on again after this condition continues for 100 µs or longer.

2) Waveform observed when both POR and LVD functions are used
(Reset pin: Pull-up resistor R_{RES} only)



- There also exists an undefined state (LVUKS) before the transistor starts functioning normally when both POR and LVD functions are used.
- Resets are generated both when power is turned on and when the power level lowers. The reset release voltage and entry voltage in this case may have some range. Refer to the latest "SANYO Semiconductors Data Sheet" for details.
- A hysteresis width (LVHYS) is provided to prevent the repetitions of reset release and entry cycles near the detection level.

4.7.6 Notes on the Use of the Internal Reset Circuit

1) When generating resets only with the POR function

When generating resets using only the POR function, do not short the reset pin directly to VDD as when using it with the LVD function. Be sure to use an external capacitor C_{RES} of an appropriate capacitance and a pull-up resistor R_{RES} or a pull-up resistor R_{RES} alone. Test the circuit extensively under the anticipated power supply conditions to verify that resets are reliably generated.

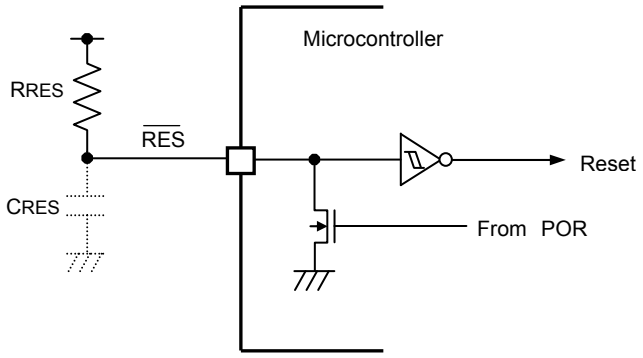


Figure 4.7.2 Reset Circuit Configuration Using only the internal POR Function

2) When selecting a release voltage level of 1.67V only with the internal POR function

When selecting an internal POR release level of 1.67V, connect the external capacitor C_{RES} and pull-up resistor R_{RES} of the values that match the power supply's rise time to the reset pin and make necessary adjustments so that the reset state is released after the release voltage exceeds the minimum guaranteed operating voltage. Alternatively, set and hold the voltage level of the reset pin at the low level until the release voltage exceeds the minimum guaranteed operating voltage.

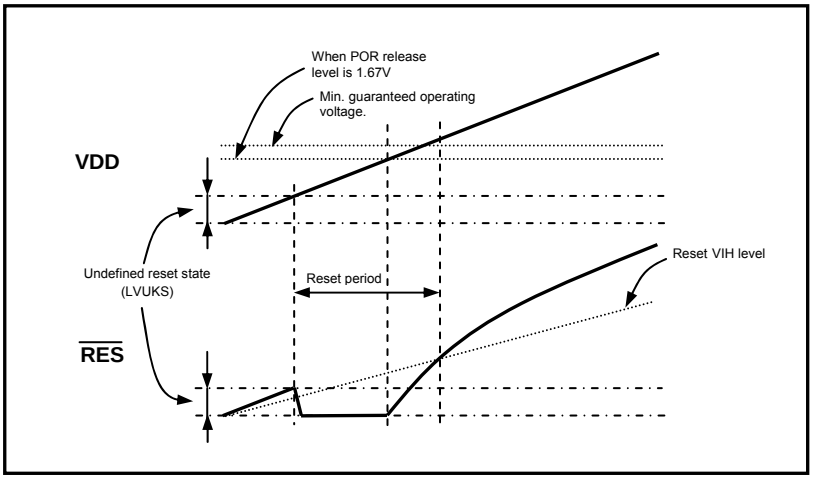


Figure 4.7.3 Sample Release Level Waveform in Internal POR Only Configuration

3) When momentary power loss or voltage fluctuations shorter than several hundred μs are anticipated

The response time measured from the time the LVD detects a power voltage drop at the option-selected level until it generates a reset signal is defined as the minimum low-voltage detection width TLVDW shown in Figure 4.7.4 (see "SANYO Semiconductors Data Sheet"). If momentary power loss or power voltage fluctuations shorter than this minimum low-voltage detection width are anticipated, be sure to take the preventive measures shown in Figure 4.7.5 or other necessary measures.

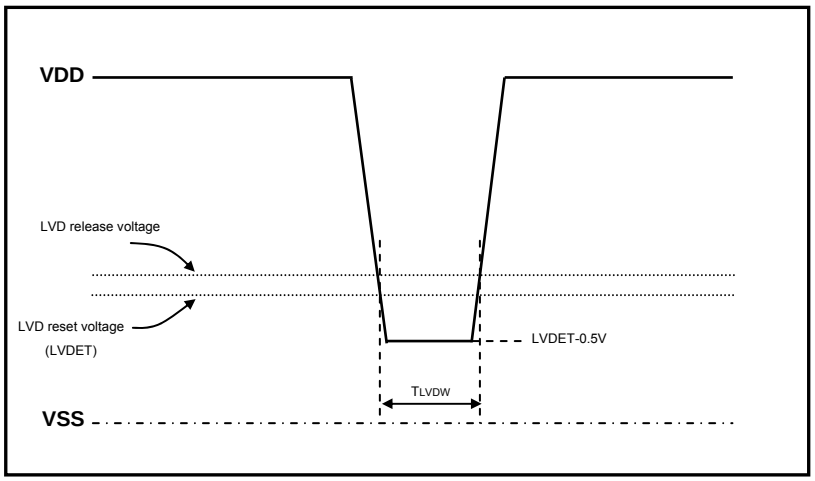


Figure 4.7.4 Example of Momentary Power Loss or Voltage Fluctuation Waveform

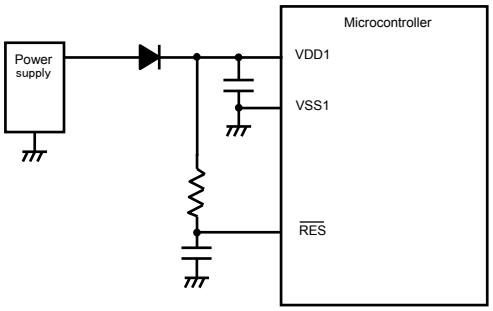


Figure 4.7.5 Example of Momentary Power Loss/Voltage Fluctuation Countermeasures

4.7.7 Notes to be Taken When Not Using the Internal Reset Circuit

1) When configuring an external reset IC without using the internal reset circuit

The POR function is activated and the capacitor C_{RES} discharging N-channel transistor connected to the reset pin turns on when power is turned on even if the internal reset circuit is not used. For this reason, when connecting an external reset IC, adopt a reset IC of a type whose detection level is not lower than the minimum guaranteed operating voltage level and select the lowest POR release level (2.57V) that does not affect the minimum guaranteed operating voltage. The figures provided below show sample reset circuit configurations that use reset ICs of N-channel open drain and CMOS types, respectively.

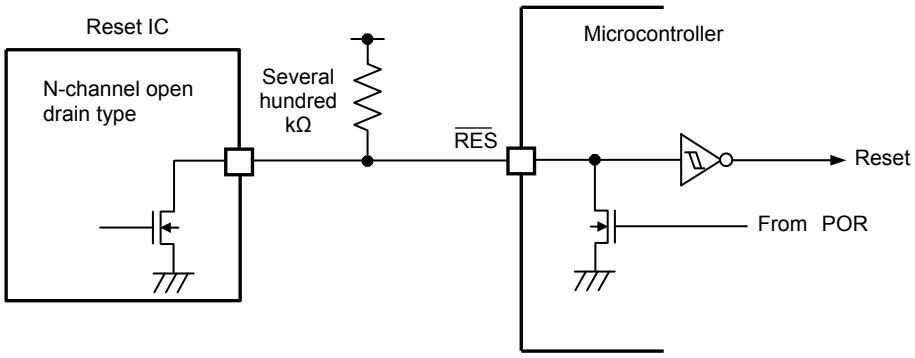


Figure 4.7.6 Sample Reset Circuit Configuration Using an N-channel Open Drain Type Reset IC

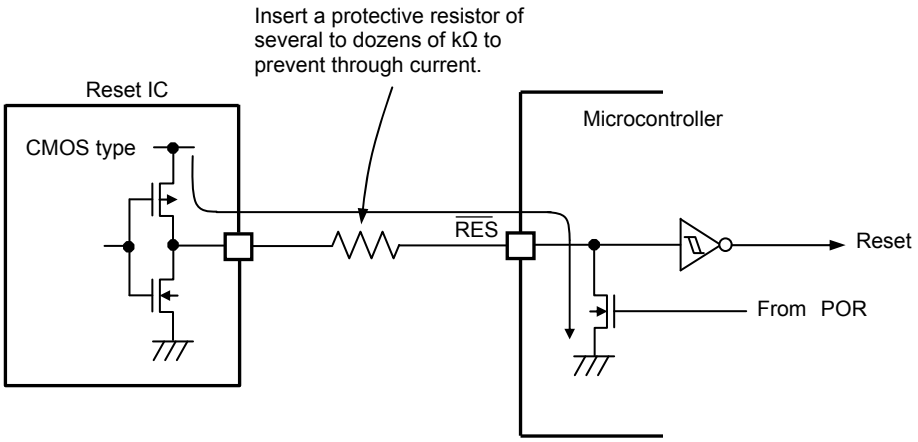


Figure 4.7.7 Sample Reset Circuit Configuration Using a CMOS Type Reset IC

2) When configuring the external POR circuit without using the internal reset circuit

The internal POR is active when the power is turned on, even if the internal reset circuit is not used as in case 1) in Subsection 4.7.7. When configuring an external POR circuit with a C_{RES} value of 0.1μF or larger to obtain a longer reset period than with the internal POR, however, be sure to connect an external diode D_{RES} as shown in Figure 4.7.8.

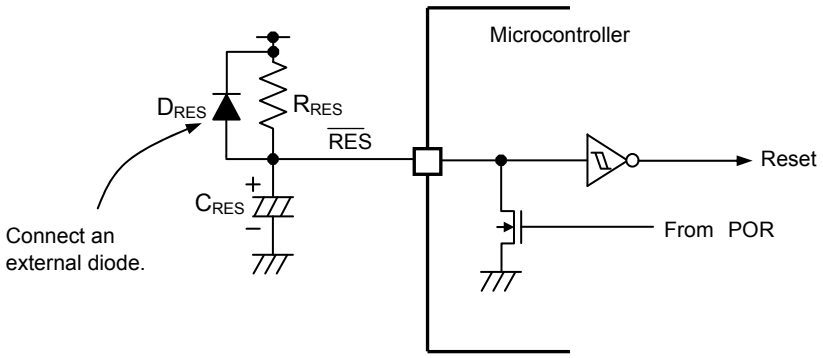


Figure 4.7.8 Sample External POR Circuit Configuration

Internal Reset

Appendixes

Table of Contents

Appendix-I

Appendix-II

Address	Initial value	R/W	LC871K00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0-1FFF	XXXX XXXX	R/W	RAM8KB	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		-	USBDIV7	DVCKON	DVCKDR	USBDIV4	USBDIV3	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-bit long (max.256Tcyc)	-	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

Special Function Register Map

Address	Initial value	R/W	LC871K00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE1E	XXXX XXXX	R	T0CA1L	Timer 0 capture register 1L	-	T0CA1L7	T0CA1L6	T0CA1L5	T0CA1L4	T0CA1L3	T0CA1L2	T0CA1L1	T0CA1L0
FE1F	XXXX XXXX	R	T0CA1H	Timer 0 capture register 1H	-	T0CA1H7	T0CA1H6	T0CA1H5	T0CA1H4	T0CA1H3	T0CA1H2	T0CA1H1	T0CA1H0
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	PWM00V	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	RM2DTOW		-	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		-	SI0BNK	SI0WRT	SI0RUN	SI0CTR	SI0DIR	SI00VR	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
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	SOXBYT[4:0] is read-only	-	SOWSTP	SWCONB6	SWCONB5	SOXBYT4	SOXBYT3	SOXBYT2	SOXBYT1	SOXBYT0
FE38	HHHH HHHH		None		-								
FE39	HHHH HHHH		None		-								
FE3A	HHHH HHHH		None										
FE3B	HHHH HHHH		None										
FE3C	0000 0000	R/W	T45CNT		-	T5C1	T5C0	T4C1	T4C0	T50V	T5IE	T40V	T4IE
FE3D	0000 0000	R/W	BTPRR		-	BTPRR7	BTPRR6	BTPRR5	BTPRR4	BTPRR3	BTPRR2	BTPRR1	BTPRR0

Address	Initial value	R/W	LC871K00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE3E	0000 0000	R/W	T4R	8-bit timer with a 6-bit prescaler	-	T4R7	T4R6	T4R5	T4R4	T4R3	T4R2	T4R1	T4R0
FE3F	0000 0000	R/W	T5R	8-bit timer with a 6-bit prescaler	-	T5R7	T5R6	T5R5	T5R4	T5R3	T5R2	T5R1	T5R0
FE40	0000 0000	R/W	P0		-	P07	P06	P05	P04	P03	P02	P01	P00
FE41	0000 0000	R/W	P0DDR		-	P07DDR	P06DDR	P05DDR	P04DDR	P03DDR	P02DDR	P01DDR	P00DDR
FE42	0000 0000	R/W	P0FCR		-	P0FCR7	P0FCR6	P0FLG	P01E	P0FCR3	P0FCR2	P0FCR1	P0FCR0
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	0000 0000	R/W	I67CR		-	INT7HEG	INT7LEG	INT7IF	INT7IE	INT6HEG	INT6LEG	INT6IF	INT6IE
FE4F	0000 0000	R/W	P0FCRU		-	T70E	T60E	SCK0SL5	SCK0SL4	CLKOEN	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 I/O (7-4:DDR 3:0:DATA)	-	P73DDR	P72DDR	P71DDR	P70DDR	P73	P72	P71	P70
FE5D	0000 0000	R/W	I01CR		-	INT1LH	INT1LV	INT1IF	INT1IE	INT0LH	INT0LV	INT0IF	INT0IE

Special Function Register Map

Address	Initial value	R/W	LC871K00	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	0000 0000	R/W	P01NTE		-	P07INTE	P06INTE	P05INTE	P04INTE	P03INTE	P02INTE	P01INTE	P00INTE
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	T7OV	T7IE	T6OV	T6IE
FE79	0000 0000	R/W	WDTCNT		-	WDTRSTF	WDTCKSL	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	0000 0000	R/W	OCR3		-	OCR3B7	OCR3B6	OCR3B5	OCR3B4	OCR3B3	OCR3B2	SRCSEL	SRCSTART

Address	Initial value	R/W	LC871K00	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 (bit4 is R/0)	-	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	HHHH HHHH		None										
FE81	HHHH HHHH		None										
FE82	HHHH HHHH		None										
FE83	HHHH HHHH		None										
FE84	HHHH HHHH		None										
FE85	HHHH HHHH		None										
FE86	HHHH HHHH		None										
FE87	HHHH HHHH		None										
FE88	HHHH HHHH		None										
FE89	HHHH HHHH		None										
FE8A	HHHH HHHH		None										
FE8B	HHHH HHHH		None										
FE8C	HHHH HHHH		None										
FE8D	HHHH HHHH		None										
FE8E	HHHH HHHH		None										
FE8F	HHHH HHHH		None										
FE90	HHHH HHHH		None										
FE91	HHHH HHHH		None										
FE92	HHHH HHHH		None										
FE93	HHHH HHHH		None										
FE94	HHHH HHHH		None										
FE95	HHHH HHHH		None										
FE96	HHHH HHHH		None										
FE97	HHHH HHHH		None										
FE98	HHHH HHHH		None										
FE99	HHHH HHHH		None										
FE9A	HHHH HHHH		None										
FE9B	HHHH HHHH		None										

Special Function Register Map

Address	Initial value	R/W	LC871K00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FE9C	HHHH HHHH		None										
FE9D	HHHH HHHH		None										
FE9E	HHHH HHHH		None										
FE9F	HHHH HHHH		None										
FEA0	0000 0000	R/W	SMIC0CNT		-	RUN	MST	TRX	SCL8	MCK	BB	END	IE
FEA1	0000 0000	R/W	SMIC0STA		-	SMD	RQL9	STD	SPD	AL	OVR	TAK	RAK
FEA2	0000 0000	R/W	SMIC0BRG		-	BRP		BRDQ	BRD				
FEA3	0000 0000	R/W	SMIC0BUF		-	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEA4	HHHH 0000	R/W	SMIC0PCT		-	-	-	-	-	SHDS	PHV	PCLV	PSLW
FEA5	0000 0000	R/W	SMIC0PSL		-	BIT7	OPSD00	OPSDA0	OPSCL0	SMI00SL	SD00EN	SDA0EN	SCL0EN
FEA6	HHHH HHHH		None										
FEA7	HHHH HHHH		None										
FEA8	0000 0000	R/W	AIFRPG		-	MCKDV2	MCKDV1	MCKDV0	FMTCEN	PRTYEN	AEROE	AERDET	AERIE
FEA9	0000 0000	R/W	AIFSTC		-	ARUNOE	MCKSEL	ASTREN	ASTRSTP	ASTPCEN	ASTPCHI	ASTPSL1	ASTPSL0
FEAA	HHHH H000	R/W	AIFDSL		-	-	-	-	-	-	ADTOSL2	ADTOSL1	ADTOSL0
FEAB	0000 0000	R/W	AIFRAT		-	APLTST0	MCLKPL	BCLKSL	AIFSWP	AFLTSL1	AFLTSL0	RATESEL1	RATESEL0
FEAC	0000 0000	R/W	AIFCON		-	APLTST1	ACLKON	AIFRUN	AIFOFF	AIFWRT	AIFDIR	AENDIF	AENDIE
FEAD	HHHH HHHH		None										
FEAE	HHHH HHHH		None										
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	0000 0000	R/W	UHCCTR		-	UHCON	UHCRUN	NKRI1	NKRI0	DRIRST	FTSHRT	UHPIEZ	UHMIEZ
FEB7	0000 00XX	R/W	UHCPRT		-	FTCK0E	UHPSEL1	UHPSEL0	UHSTBY	UHDRP	UHDRM	UHDTP	UHDTM
FEB8	0000 0000	R/W	UHCINT		-	DTCIF	DTCIE	RSMIF	RSMIE	SOFIF	SOFIE	ATCIF	ATCIE
FEB9	0000 0000	R/W	TRSINT		-	ACKIF	ACKIE	NAKIF	NAKIE	ERRIF	ERRIE	STLIF	STLIE
FEBA	0000 0000	R/W	UHFRML		-	HFRM7	HFRM6	HFRM5	HFRM4	HFRM3	HFRM2	HFRM1	HFRM0

Address	Initial value	R/W	LC871K00	Remarks	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FEBB	0000 0000	R/W	UHFRMH		-	HFRM15	HFRM14	HFRM13	HFRM12	HFRM11	HFRM10	HFRM9	HFRM8
FEBC	H000 0000	R/W	DEVADR		-	-	DADR6	DADR5	DADR4	DADR3	DADR2	DADR1	DADR0
FEBD	HHHH 0000	R/W	DEVEPN		-	-	-	-	-	DEPN3	DEPN2	DEPN1	DEPN0
FEBE	0000 0000	R/W	BUSCTR		-	RSTRUN	RSMRUN	SOFRUN	NKRTRY	ERRTRY	ALS	CBS1	CBS0
FEBF	0000 0000	R/W	UHCSTA		-	TRSRUN	TRSTGL	TRSOVR	TGLERR	TTKN1	TTKN0	TRSIS0	TGLMON
FEC0	0000 0000	R/W	UHCCNT		-	UHCCN7	UHCCN6	UHCCN5	UHCCN4	UHCCN3	UHCCN2	UHCCN1	UHCCN0
FEC1	0000 0000	R/W	UHCRX		-	UHCRX7	UHCRX6	UHCRX5	UHCRX4	UHCRX3	UHCRX2	UHCRX1	UHCRX0
FEC2	0000 00HH	R/W	UHCADL		-	UHAD7	UHAD6	UHAD5	UHAD4	UHAD3	UHAD2	-	-
FEC3	HHH0 0000	R/W	UHCADH		-	-	-	-	UHAD12	UHAD11	UHAD10	UHAD9	UHAD8
FEC4	0000 0000	R/W	BACINT		-	UHCRX9	UHCRX8	UHCCN9	UHCCN8	BACBIF	BACBIE	BACIF	BACIE
FEC5	0000 0000	R/W	UHTST0		-	UHTEST	UHCMPT	UHCMPK	VD3KIL	R2RAM	UHTAD2	UHTAD1	UHTAD0
FEC6	0000 0000	R/W	UHTST1		-	UHTDT7	UHTDT6	UHTDT5	UHTDT4	UHTDT3	UHTDT2	UHTDT1	UHTDT0
FEC7	HHHH 0000	R/W	BPTCTR		-	-	-	-	-	BLS	BPTRST	BPTRSM	BPTSOF
FEC8	0000 0000	R/W	BPTINT		-	BDTCIF	BDTCIE	BRSMIF	BRSMIE	BATCIF	BATCIE	BCBS1	BCBS0
FEC9	0000 0000	R/W	AIFMOD		-	AIFLN1	AIFLN0	AIFFB	SDATISEL	IISEN	MCLKIN	ASTAIF	ASTAIE
FECA	0000 0000	R/W	AIFCLK		-	AVCOON	ACMPON	AIFCK1	AIFCK0	AVCOSEL	LRCKPL	BCLKPL	BCKSEL
FECB	0000 0000	R/W	AIFPC		-	BCLKL0	LRCKIN	BCLKIN	SDATIN	MCLKDR	LRCKDR	BCLKDR	SDATDR
FECC	0000 0000	R/W	AIFCNL		-	ACNT7	ACNT6	ACNT5	ACNT4	ACNT3	ACNT2	ACNT1	ACNT0
FECD	HHH0 0000	R/W	AIFCNH		-	-	-	-	ACNT12	ACNT11	ACNT10	ACNT9	ACNT8
FECE	0000 0000	R/W	AIFADL		-	AADR7	AADR6	AADR5	AADR4	AADR3	AADR2	AADR1	AADR0
FECF	HHH0 0000	R/W	AIFADH		-	-	-	-	AADR12	AADR11	AADR10	AADR9	AADR8
FED0	0000 0000	R/W	UCON0		-	UBRSEL	STRDET	RECRUN	STPERR	UOB3	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		-	TBUF7	TBUF6	TBUF5	TBUF4	TBUF3	TBUF2	TBUF1	TBUF0
FED4	0000 0000	R/W	RBUF		-	RBUF7	RBUF6	RBUF5	RBUF4	RBUF3	RBUF2	RBUF1	RBUF0
FED5	0000 0000	R/W	UMDSL		-	UMB7	UMB6	UMB5	UMXTS1	UMXTS0	UMMCS	TEND	TENIE
FED6	0000 0000	R/W	S4ADRL		-	S4ADL7	S4ADL6	S4ADL5	S4ADL4	S4ADL3	S4ADL2	S4ADL1	S4ADL0
FED7	0000 0000	R/W	S4BYTH		-	S4STPWD	S4BYTRD	S4ADR6	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

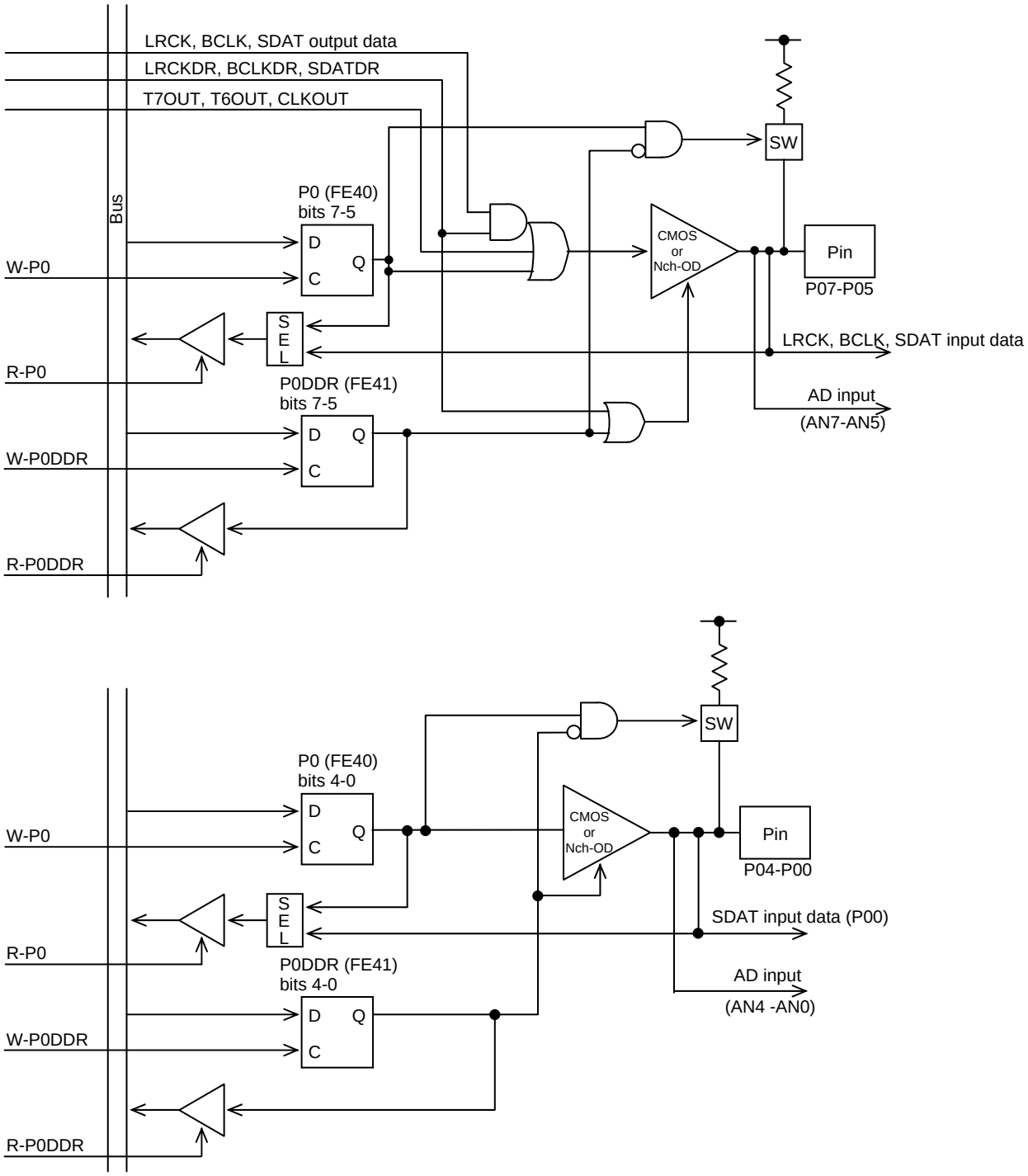
Special Function Register Map

Address	Initial value	R/W	LC871K00	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	HHHH HHHH		None										
FEE2	HHHH HHHH		None										
FEE3	HHHH HHHH		None										
FEE4	HHHH HHHH		None										
FEE5	HHHH HHHH		None										
FEE6	HHHH HHHH		None										
FEE7	HHHH HHHH		None										
FEE8	HHHH HHHH		None										
FEE9	HHHH HHHH		None										
FEEA	HHHH HHHH		None										
FEEB	HHHH HHHH		None										
FEEC	HHHH HHHH		None										
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	LC871K00	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										

Table of Port 0 Multiplexed Functions

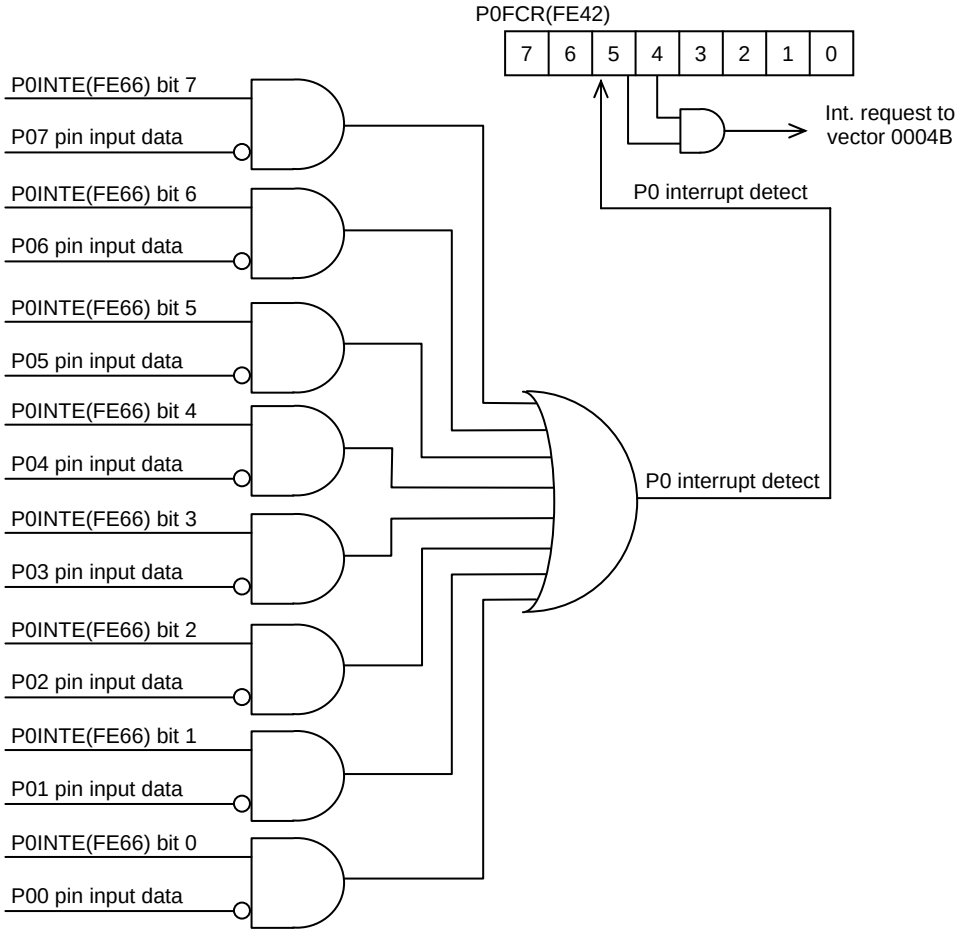
Port	Function Output		Function Input
P07	Timer 7 toggle output	LRCK output	LRCK input
P06	Timer 6 toggle output	BCLK output	BCLK input
P05	Clock output (system clock/subclock selectable)	SDAT output	SDAT input
P00	None		SDAT input



Port 0 Block Diagram

Option: Output type (CMOS or N-channel OD) selectable in 1-bit units

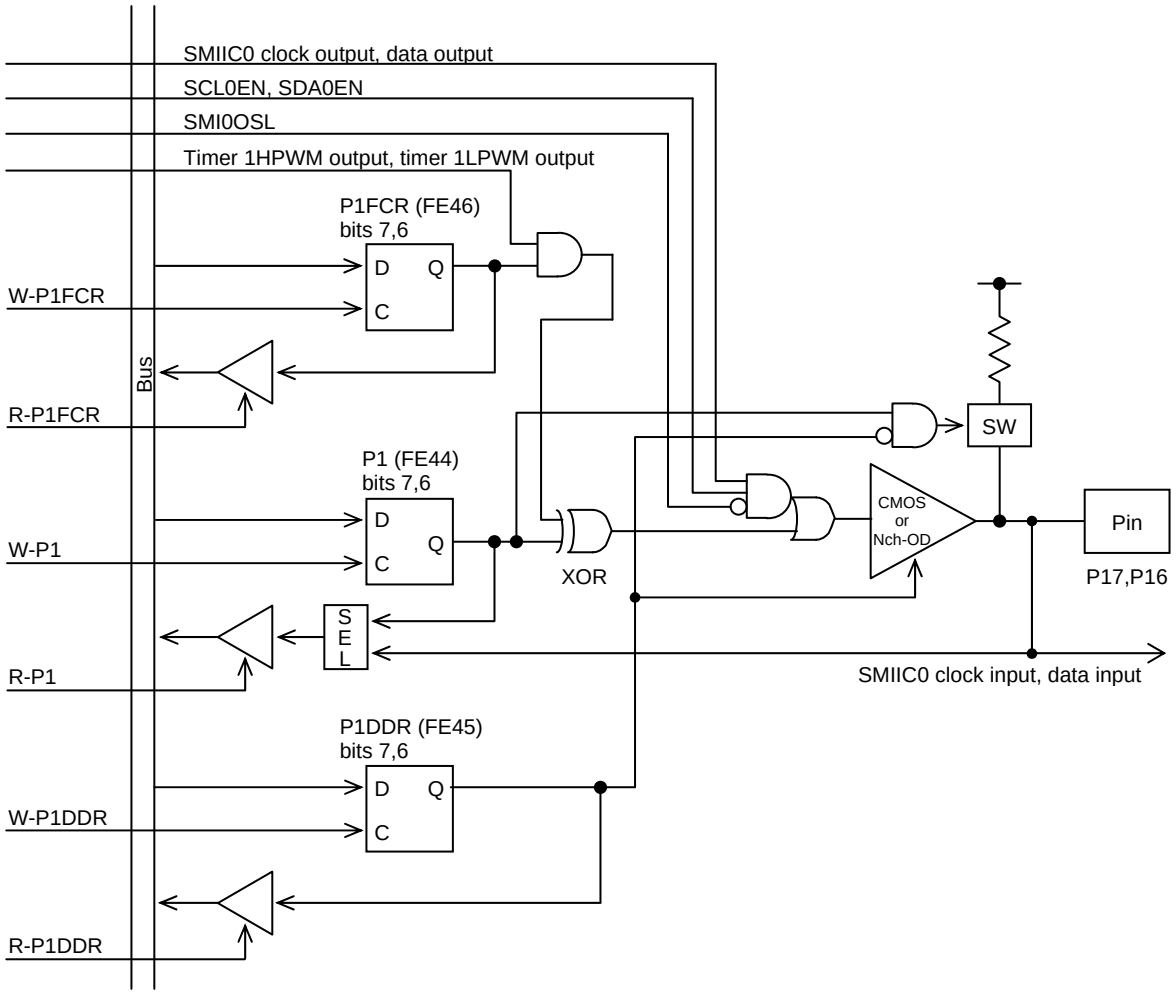
Port Block Diagram



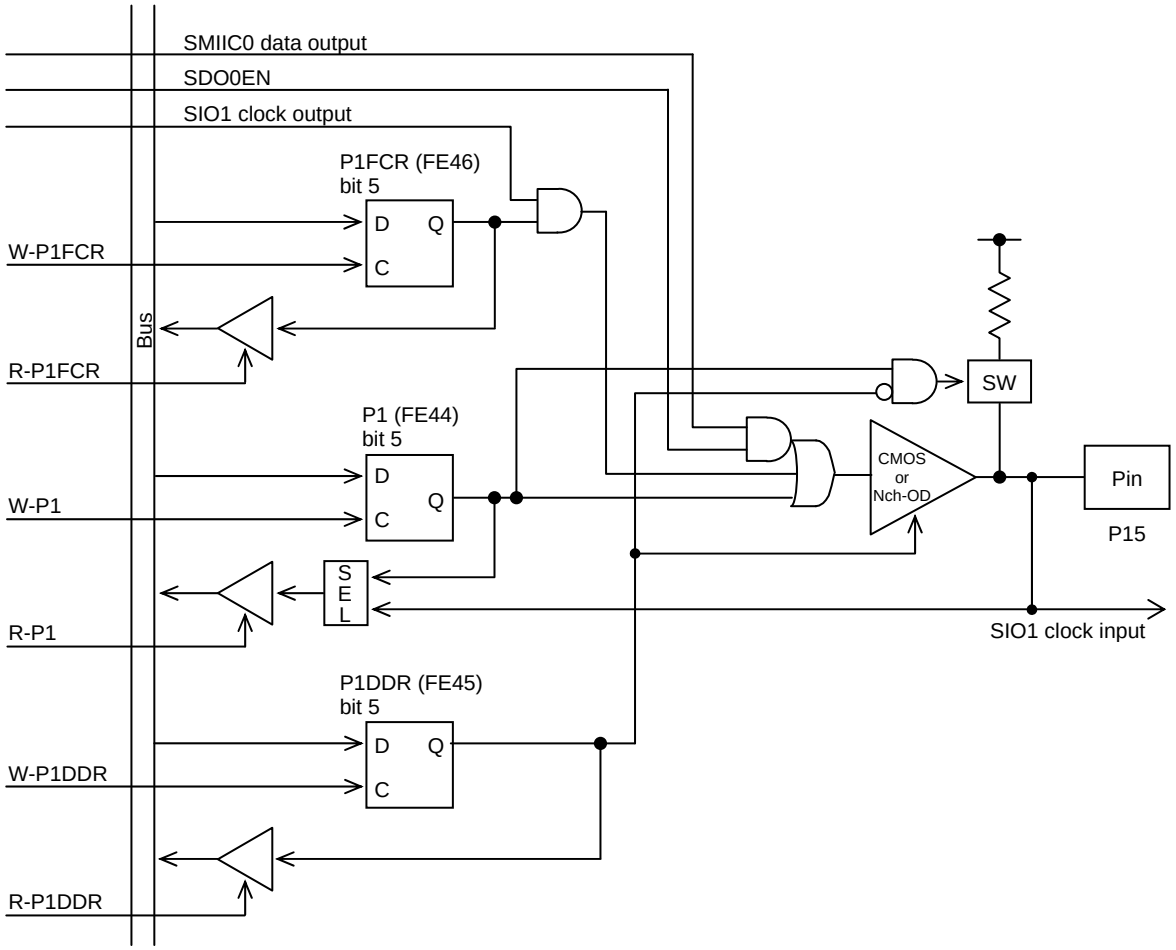
Port 0 (Interrupt) Block Diagram

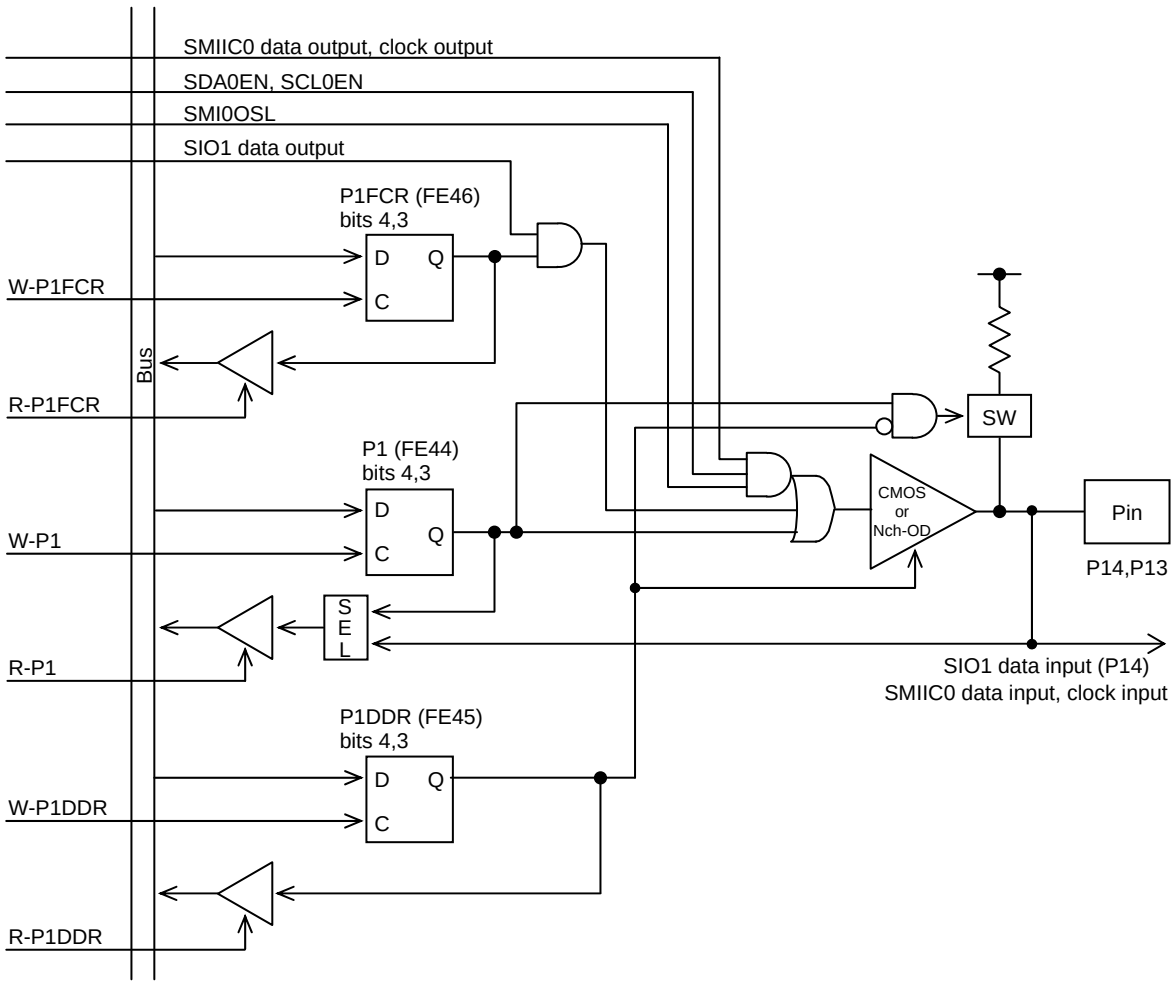
Table of Port 1 Multiplexed Functions

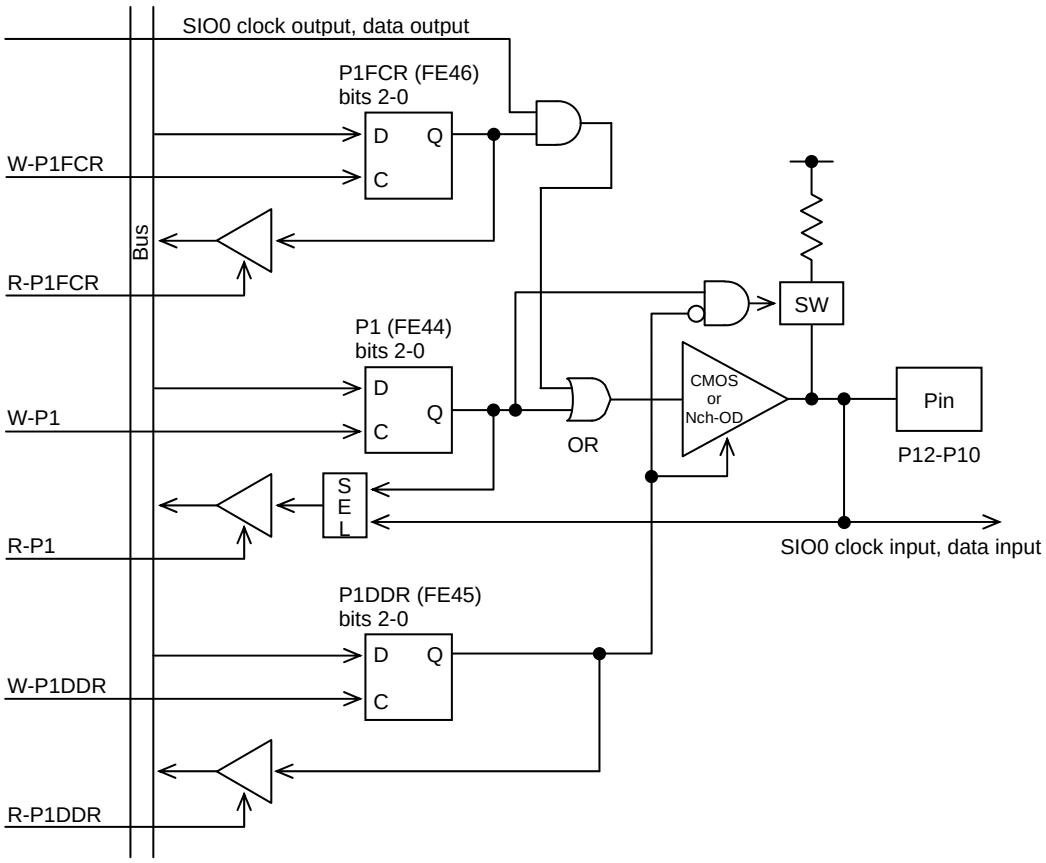
Port	Function Output		Function Input	
P17	Timer 1HPWM output	SMIIC0 clock output	SMIIC0 clock input	
P16	Timer 1LPWM output	SMIIC0 data output	SMIIC0 data input	
P15	SIO1 clock output	SMIIC0 data output	SIO1 clock input	
P14	SIO1 data output	SMIIC0 data output	SIO1 data input	SMIIC0 data input
P13	SIO1 data output	SMIIC0 clock output	SMIIC0 clock input	
P12	SIO0 clock output		SIO0 clock input	
P11	SIO0 data output		SIO0 data input	
P10	SIO0 data output		None	



Port Block Diagram







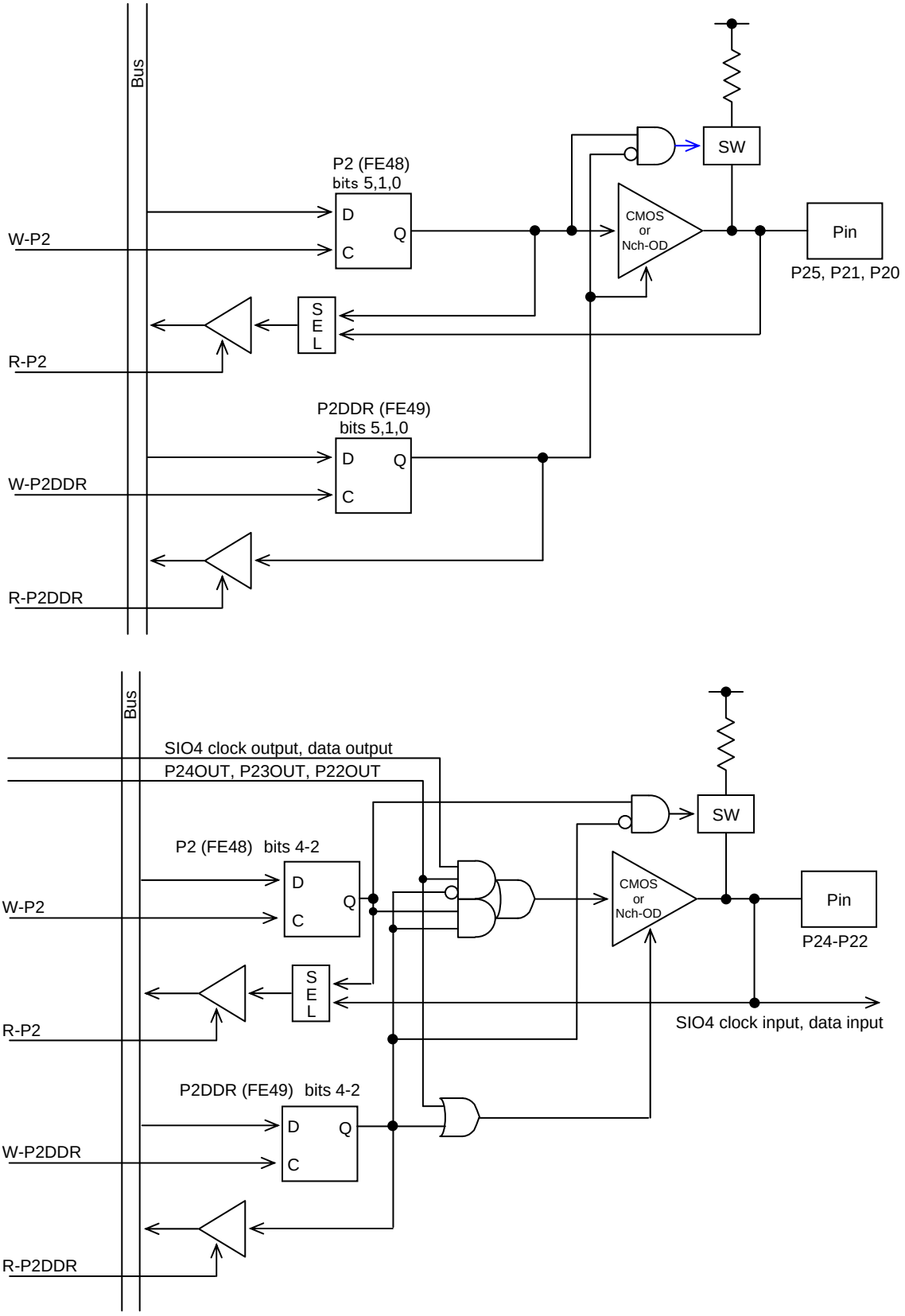
Port 1 Block Diagram

Option: Output type (CMOS or N-channel OD) selectable in 1-bit units

Port Block Diagram

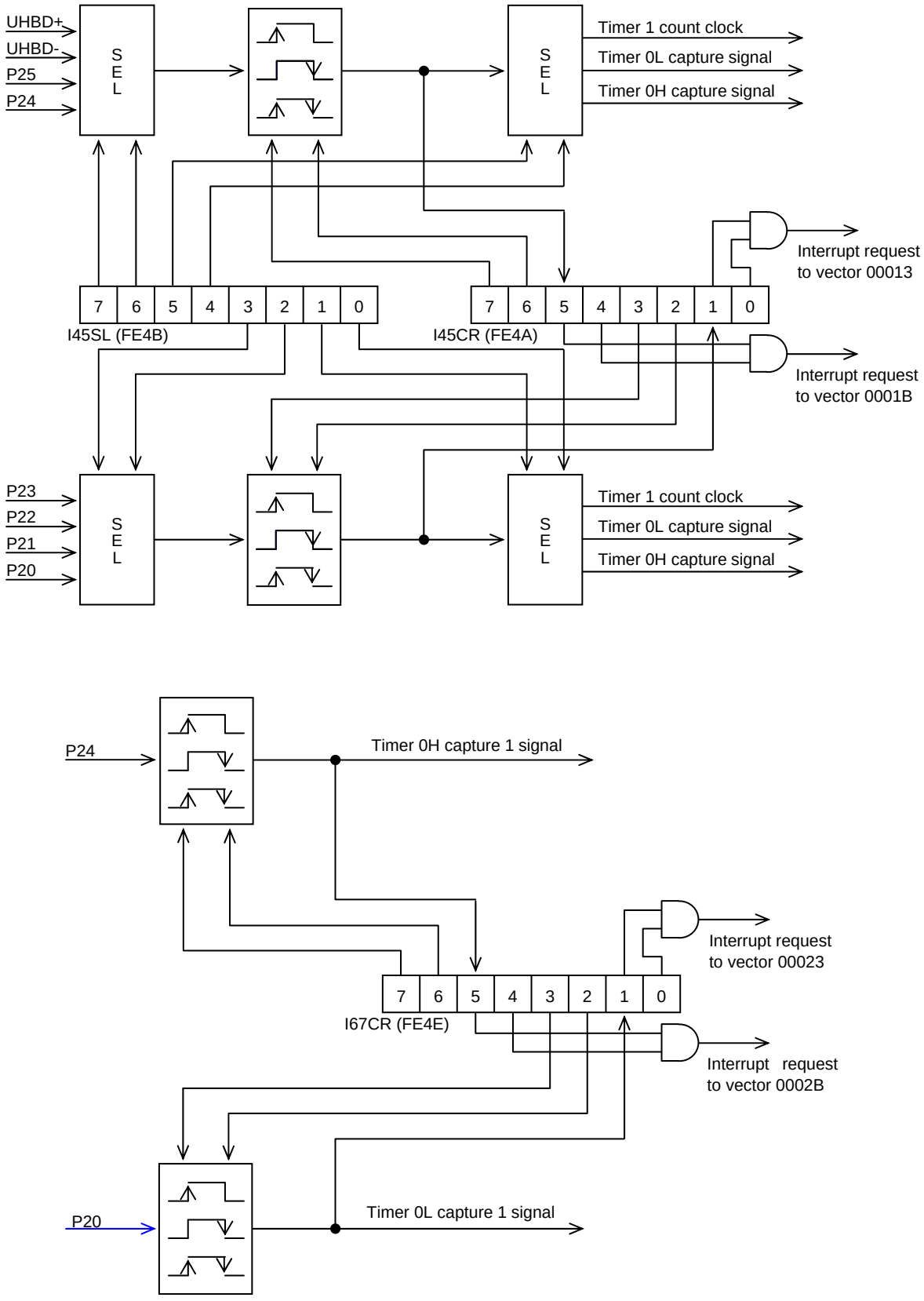
Table of Port 2 Multiplexed Functions

Port	Function Output	Function Input
P24	SIO4 clock output	SIO4 clock input
P23	SIO4 data output	SIO4 data input
P22	SIO4 data output	SIO4 data input



Port 2 (Pin) Block Diagram

Option: Output type (CMOS or N-channel OD) selectable in 1-bit units

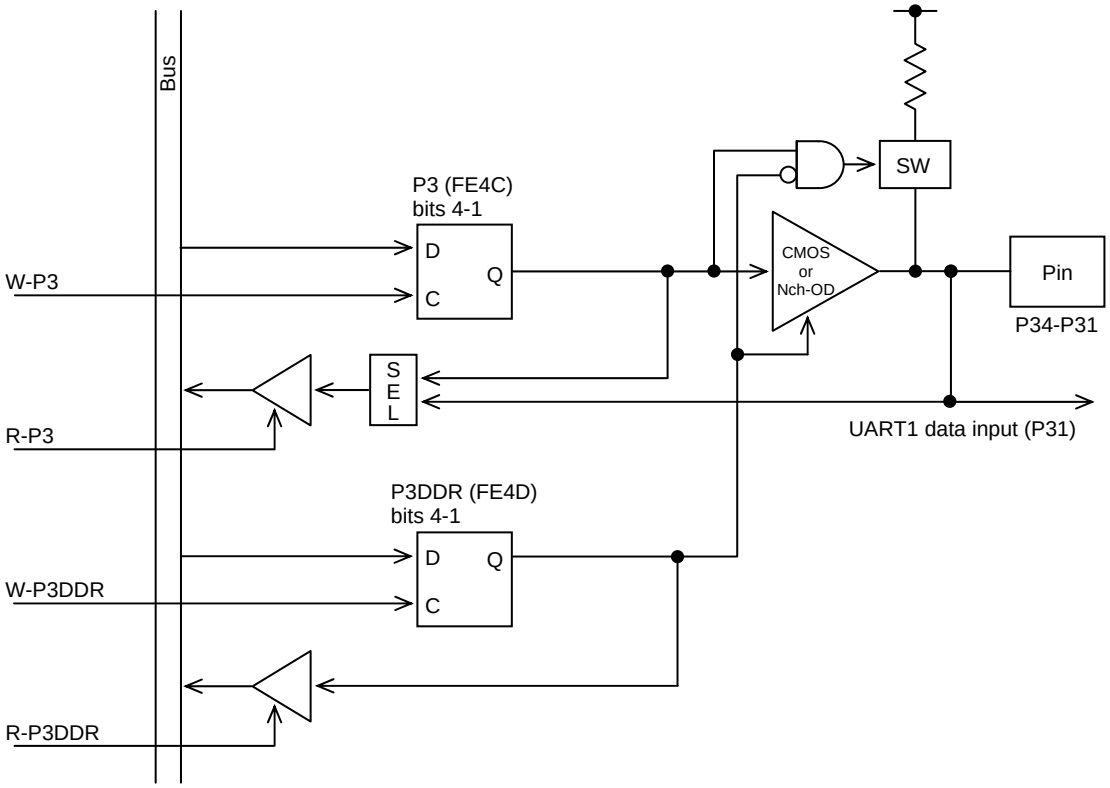


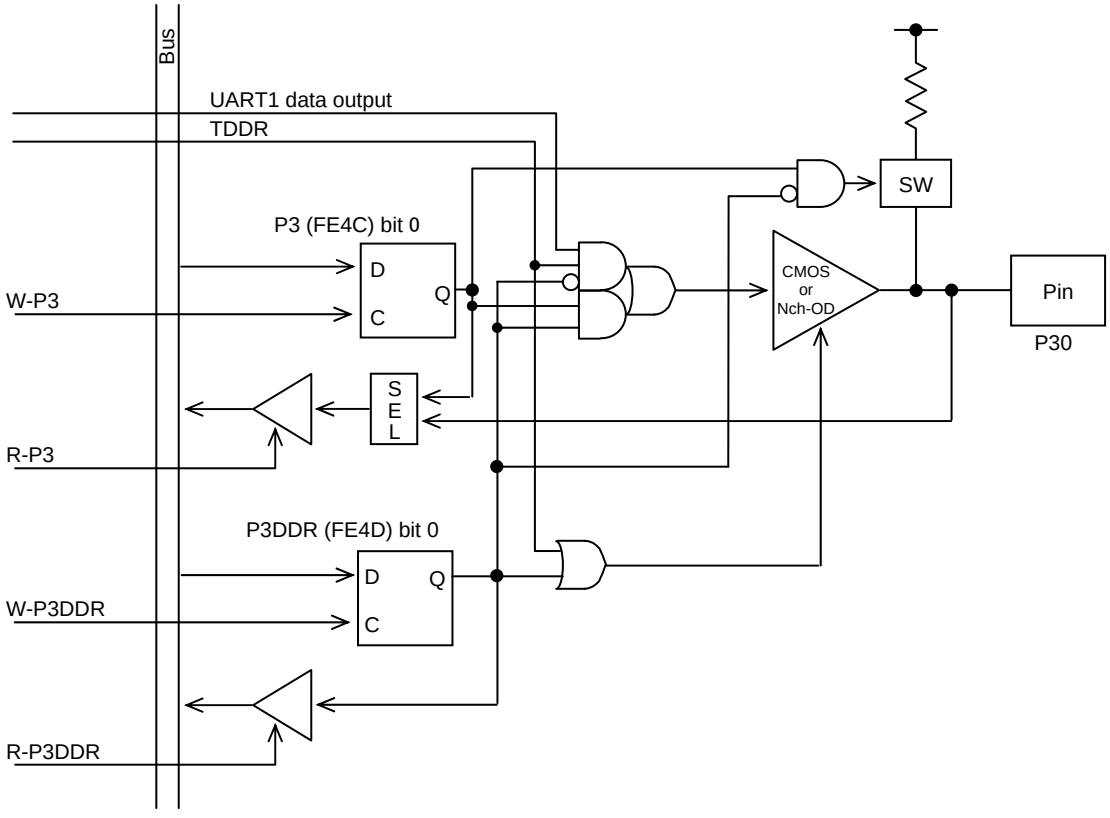
Port 2 (Interrupt) Block Diagram

Port Block Diagram

Table of Port 3 Multiplexed Functions

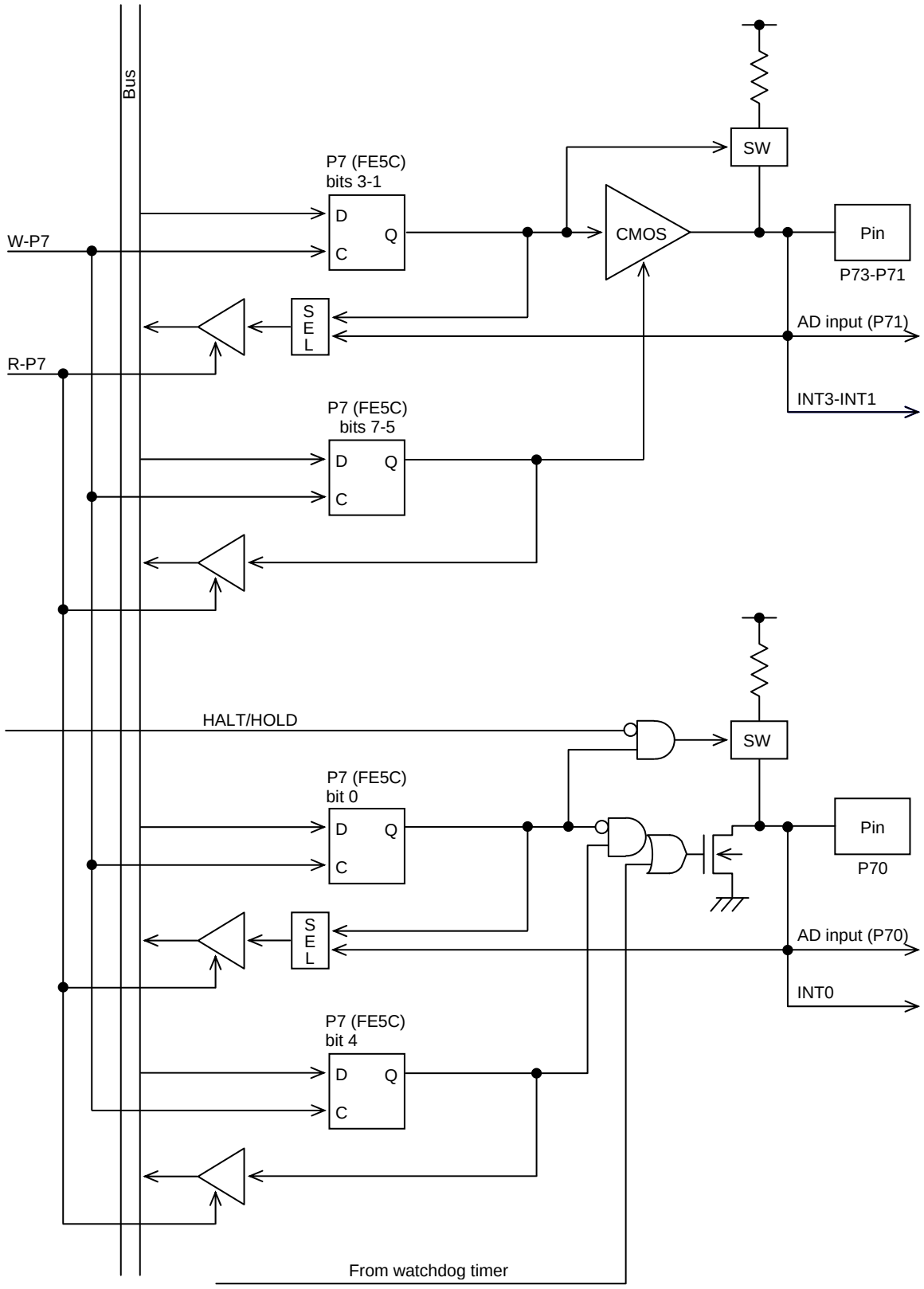
Port	Function Output	Function Input
P31	None	UART1 data input
P30	UART1 data output	None





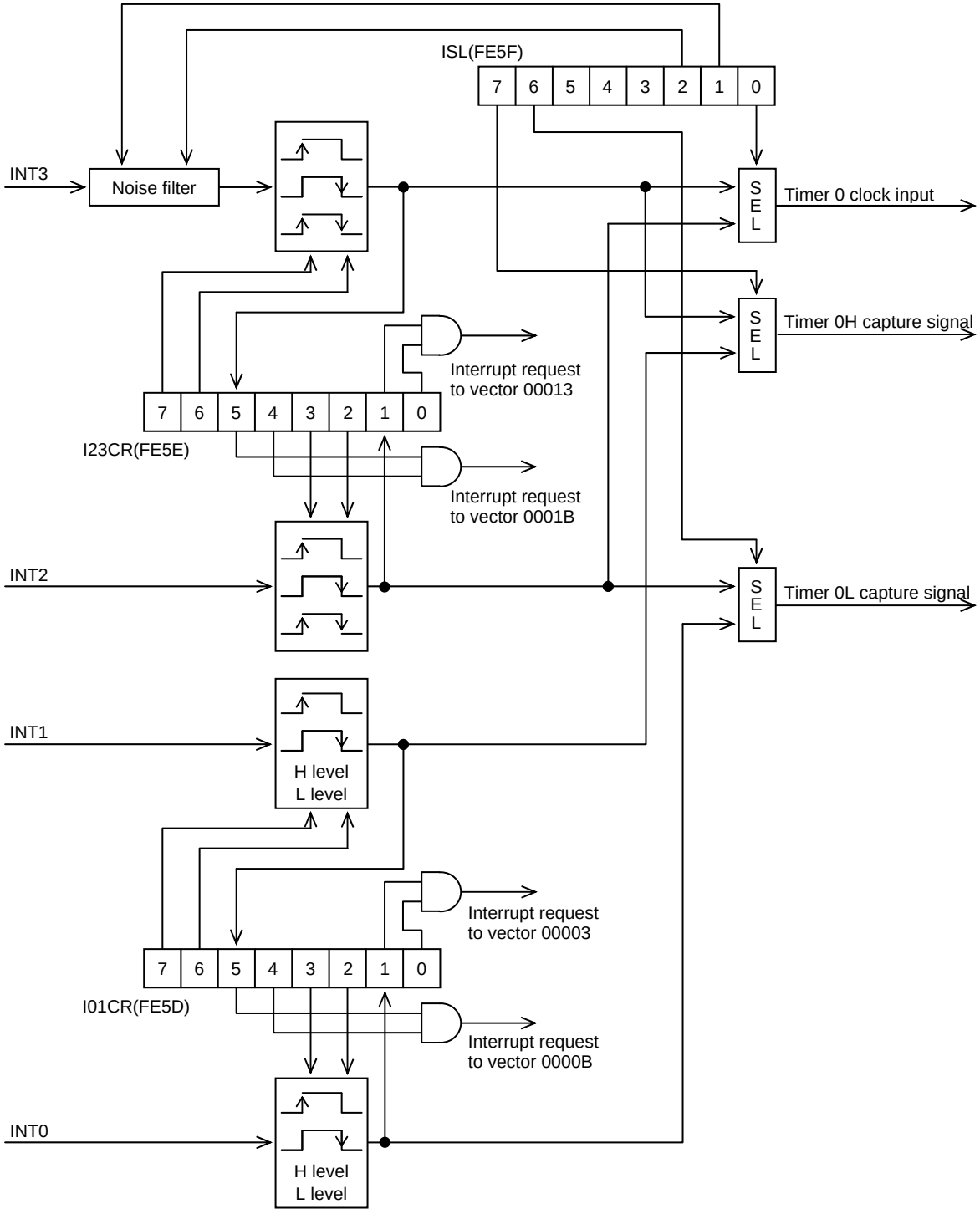
Port 3 Block Diagram

Option: Output type (CMOS or N-channel OD) selectable in 1-bit units

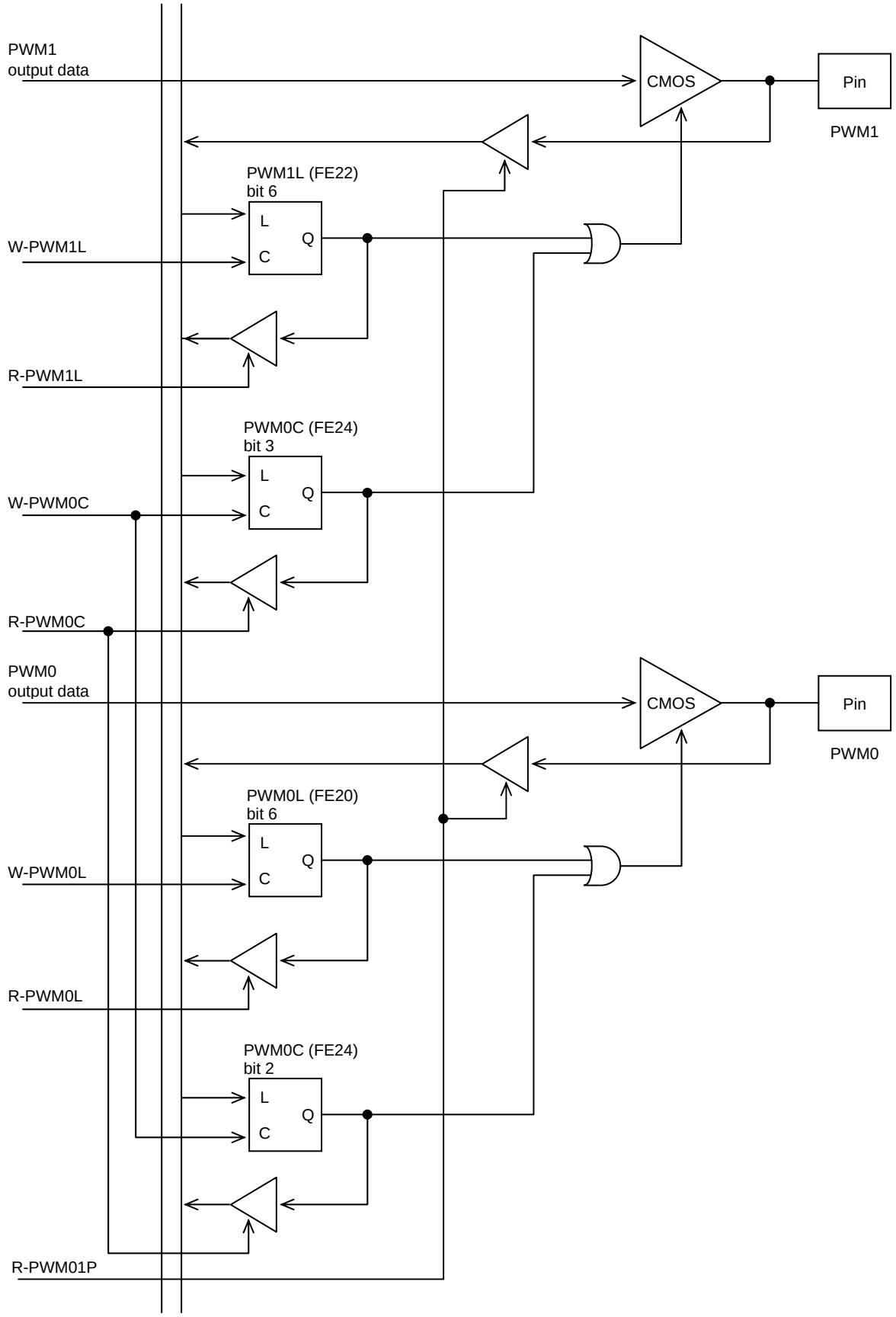


Port 7 (Pin) Block Diagram
Option: None

Port Block Diagram

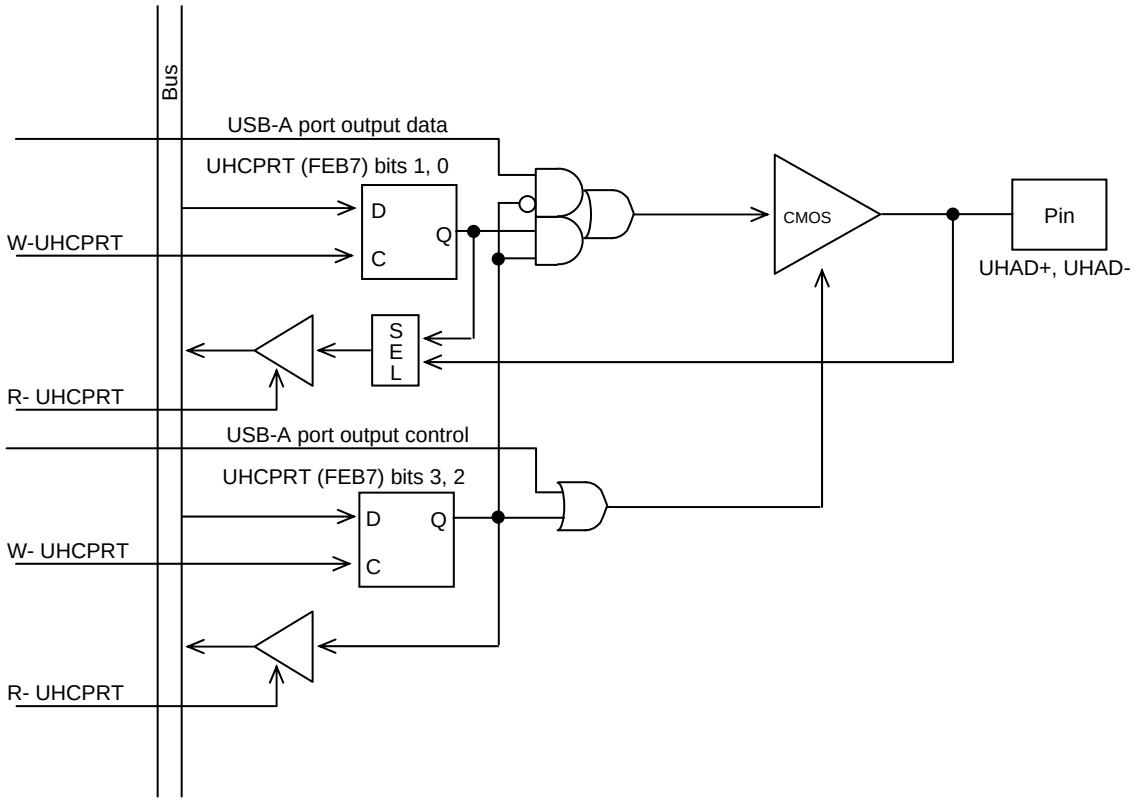


Port 7 (Interrupt) Block Diagram

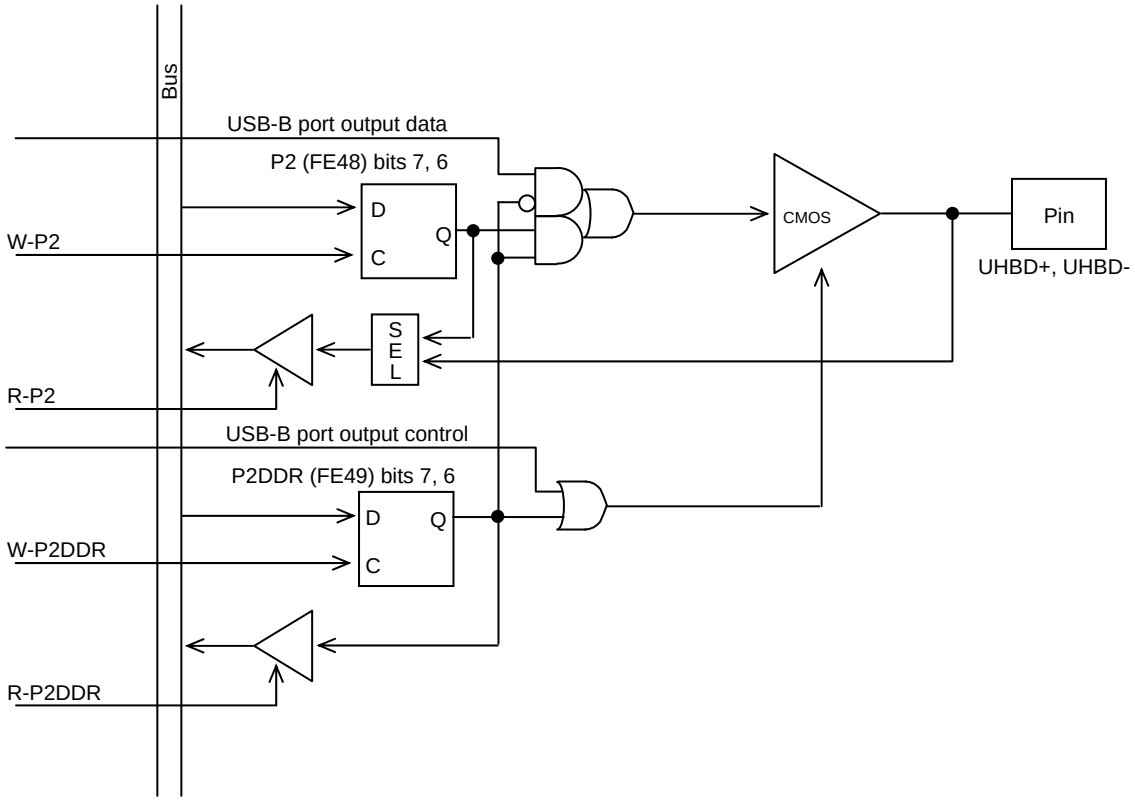


PWM0, PWM1 Block Diagram
Option: None

Port Block Diagram



USB-A Port (Pin) Block Diagram



USB-B Port (Pin) Block Diagram

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.

LC871K00 SERIES USER'S MANUAL

Rev : 1.00 April 24, 2012

ON Semiconductor

Digital Solution Division

Microcontroller & Flash Business Unit