# ON Semiconductor

# Is Now

# onsemi™

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

CMOS 8-BIT MICROCONTROLLER

# LC871M00 SERIES USER'S MANUAL

REV : 1.00

ON Semiconductor®

http://onsemi.com

ON Semiconductor
Digital Solution Division
Microcontroller & Flash Business Unit

# Contents

## Contents

# Contents

## Appendixes

# 1. Overview

## 1.1 Overview

The LC871M00 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 16K-byte flash ROM (onboard programmable), 1024-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 automatic transfer functions), an asynchronous/synchronous SIO interface, a UART interface (full duplex), a UART interface with a smart card interface function (full duplex), a full-speed USB interface (with function control functions), a 20-channel AD converter with 12-/8-bit resolution selector , two 12-bit PWM channels, a system clock frequency divider, an internal reset circuit, and 35-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 possible
- Writing in two-byte units
- 16384 × 8 bits

● RAM

- 1024 × 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<br>Ports whose input/output can be specified in 1-bit units: | 35 (P00 to P07, P10 to P17, P20 to P27, P31 to P34, P70 to P73, PWM0, PWM1, XT2) |
| • USB ports: | 2 (D+, D$^-$) |
| • Dedicated oscillator ports: | 2 (CF1, CF2) |
| • Input-only port (also used for oscillation): | 1 (XT1) |
| • Reset pin: | 1 ($\overline{\text{RES}}$) |
| • Debugger pin: | 1 (OWP0) |
| • 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 low-order 8 bits)

    Mode 3: 16-bit timer with an 8-bit prescaler (with toggle output) (low-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), 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 1024 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

- Full duplex UART
  - UART1
    1) Data length: 7/8/9 bits
    2) Stop bits: 1 bit (2 bits in continuous transmission mode)
    3) Baudrate: 16/3 to 8192/3 Tcyc
  - SCUART
    1) Data length: 7/8 bits
    2) Stop bits: 1 bit/2 bits
    3) Parity bit: None/even parity/odd parity
    4) Baudrate: 8/3 to 8192/3 Tcyc
    5) LSB first/MSB first selectable
    6) Smart card interface function

- AD converter: 12 bits × 20 channels
  - 12-/8-bit AD converter resolution selectable

- PWM: Multifrequency 12-bit PWM × 2 channels

- USB interface (with function control functions)
  - Conforms to USB specification version 2.0 (full speed)
  - Supports up to 6 user-defined endpoints.

| Endpoint | | EP0 | EP1 | EP2 | EP3 | EP4 | EP5 | EP6 |
|---|---|---|---|---|---|---|---|---|
| Transfer type | Control | ○ | — | — | — | — | — | — |
| | Bulk | — | ○ | ○ | ○ | ○ | ○ | ○ |
| | Interrupt | — | ○ | ○ | ○ | ○ | ○ | ○ |
| | Isochronous | — | ○ | ○ | ○ | ○ | ○ | ○ |
| Maximum payload | | 64 | 64 | 64 | 64 | 64 | 64 | 64 |

- Watchdog timer
  - Watchdog timer with an internal counter
    1) Capable of generating an internal reset on an overflow of a timer that runs on either a dedicated low-speed RC oscillator clock (30 kHz) or subclock.
    2) WDT operation on entry into HALT or HOLD mode can be selected from three modes (count 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
  - 35 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/USB bus active |
| 4 | 0001BH | H or L | INT3/INT5/base timer |
| 5 | 00023H | H or L | T0H/ INT6 |
| 6 | 0002BH | H or L | T1L/T1H/INT7 |
| 7 | 00033H | H or L | SIO0/USB bus reset/USB suspend/UART1 receive end/ SCUART receive end |
| 8 | 0003BH | H or L | SIO1/USB endpoint/USB-SOF/SIO4/UART1 buffer empty/UART1 transmit end/ SCUART buffer empty/SCUART transmit end |
| 9 | 00043H | H or L | ADC/T6/T7 |
| 10 | 0004BH | H or L | Port 0/PWM0/PWM1/T4/T5 |

- 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.

- Subroutine stack levels: Up to 512 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

| | |
|---|---|
| • RC oscillator circuit (internal): | For system clock (1 MHz) |
| • Low-speed RC oscillator circuit (internal): | For watchdog timer (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 |

- 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 4 levels (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 (3 levels: 2.81V, 3.79V, and 4.28V) can be selected 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 base clock generator, and CF, RC and crystal oscillators automatically stop operation.

    *Note: The low-speed RC oscillation is 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 interface circuit.
  - X’tal HOLD mode: Suspends instruction execution and the operation of the peripheral circuits except the base timer.
    1) The PLL base clock generator, and CF and RC oscillators automatically stop operation.

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

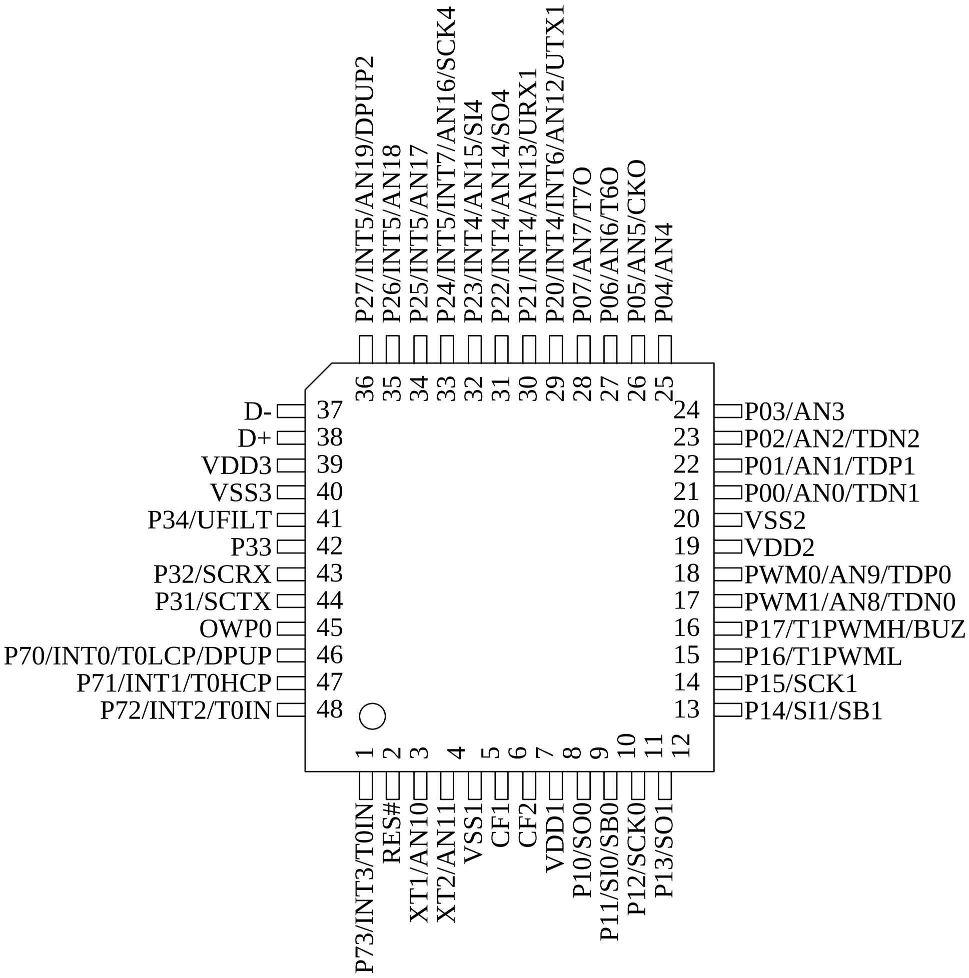
    2) The state of crystal oscillation established when X'tal HOLD mode is entered is retained.
    3) There are six ways of releasing X'tal HOLD mode.
       <1> Low level input to the reset pin
       <2> Generating a reset by the watchdog timer or low-voltage detection.
       <3> Establishing an interrupt 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 interface circuit

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

- Development tools
  - On-chip debugger: TCB87-Type C (single wire communication cable) + LC87F1M16A

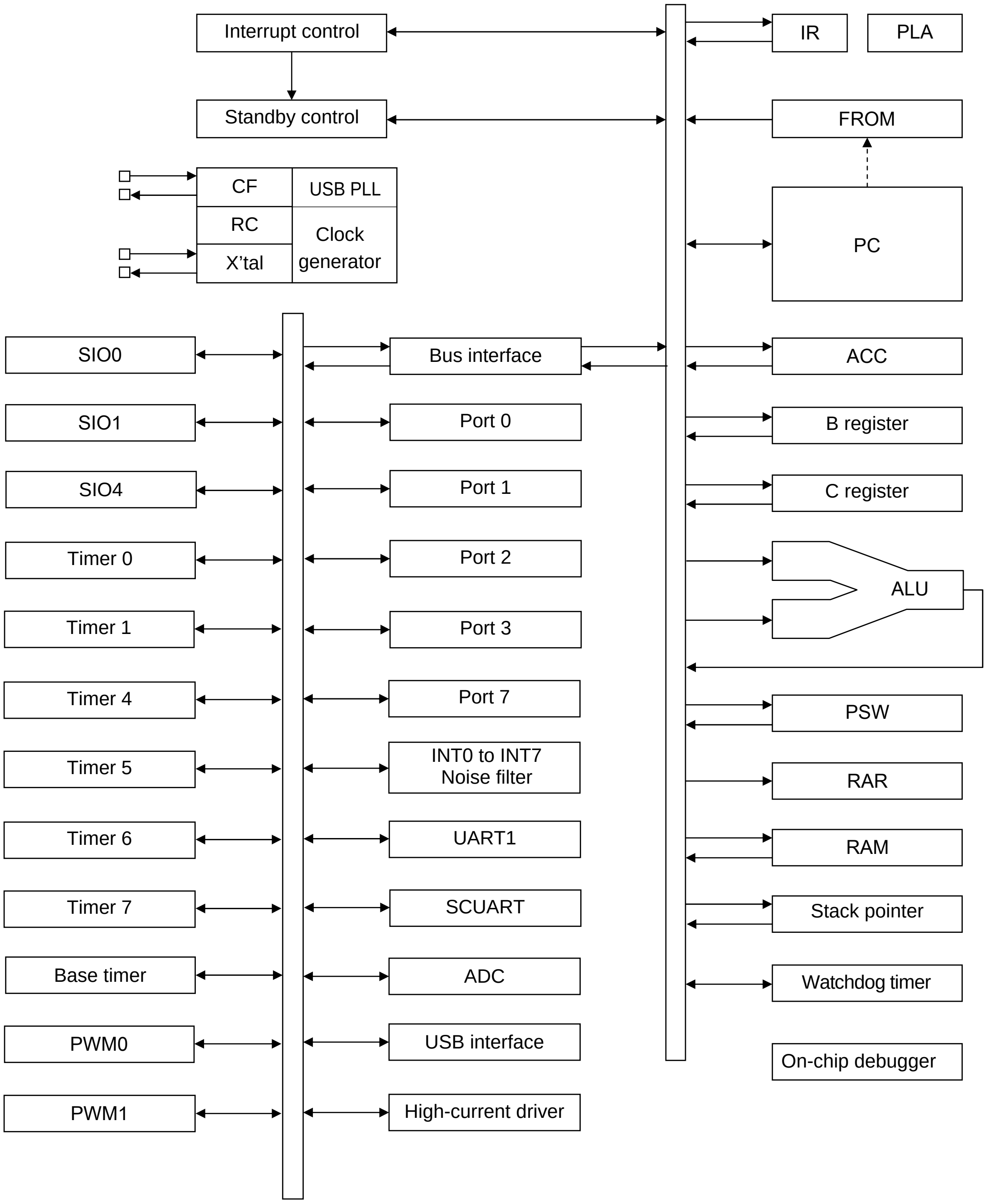
## 1.3 Pinout

**SQFP48**



| SQFP48 | NAME | SQFP48 | NAME |
|---|---|---|---|
| 1 | P73/INT3/T0IN | 25 | P04/AN4 |
| 2 | RES# | 26 | P05/AN5/CKO |
| 3 | XT1/AN10 | 27 | P06/AN6/T6O |
| 4 | XT2/AN11 | 28 | P07/AN7/T7O |
| 5 | VSS1 | 29 | P20/INT4/INT6/AN12/UT |
| 6 | CF1 | 30 | P21/INT4/AN13/URX1 |
| 7 | CF2 | 31 | P22/INT4/AN14/SO4 |
| 8 | VDD1 | 32 | P23/INT4/AN15/SI4 |
| 9 | P10/SO0 | 33 | P24/INT5/INT7/AN16/SCK |
| 10 | P11/SI0/SB0 | 34 | P25/INT5/AN17 |
| 11 | P12/SCK0 | 35 | P26/INT5/AN18 |
| 12 | P13/SO1 | 36 | P27/INT5/AN19/DPUP2 |
| 13 | P14/SI1/SB1 | 37 | D- |
| 14 | P15/SCK1 | 38 | D+ |
| 15 | P16/T1PWML | 39 | VDD3 |
| 16 | P17/T1PWMH/BUZ | 40 | VSS3 |
| 17 | PWM1/AN8/TDN0 | 41 | P34/UFILT |
| 18 | PWM0/AN9/TDP0 | 42 | P33 |
| 19 | VDD2 | 43 | P32/SCRX |
| 20 | VSS2 | 44 | P31/SCTX |
| 21 | P00/AN0/TDN1 | 45 | OWP0 |
| 22 | P01/AN1/TDP1 | 46 | P70/INT0/T0LCP/DPUP |
| 23 | P02/AN2/TDN2 | 47 | P71/INT1/T0HCP |
| 24 | P03/AN3 | 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,<br>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<br>P00 to P07</td><td>I/O</td><td>• 8-bit I/O port<br>• I/O can be specified in 1-bit units<br>• Pull-up resistors can be turned on and off in 1-bit units<br>• HOLD release input<br>• Port 0 interrupt input<br>• Pin functions<br>AD converter input port: AN0 to AN7 (P00 to P07)<br>P00: High-current N-channel driver<br>P01: High-current P-channel driver<br>P02: High-current N-channel driver<br>P05: System clock output<br>P06: Timer 6 toggle output<br>P07: Timer 7 toggle output</td><td>Yes</td></tr>
<tr><td>Port 1<br>P10 to P17</td><td>I/O</td><td>• 8-bit I/O port<br>• I/O can be specified in 1-bit units<br>• Pull-up resistors can be turned on and off in 1-bit units<br>• Pin functions<br>P10: SIO0 data output<br>P11: SIO0 data input/ bus I/O<br>P12: SIO0 clock I/O<br>P13: SIO1 data output<br>P14: SIO1 data input/ bus I/O<br>P15: SIO1 clock I/O<br>P16: Timer 1 PWML output<br>P17: Timer 1 PWMH output/buzzer output</td><td>Yes</td></tr>
<tr><td>Port 2<br>P20 to P27</td><td>I/O</td><td>• 8-bit I/O port<br>• I/O can be specified in 1-bit units<br>• Pull-up resistors can be turned on and off in 1-bit units<br>• Pin functions<br>AD converter input port: AN12 to AN19 (P20 to P27)<br>P20 to P23: INT4 I/O /HOLD release input/timer 1 event input/ timer 0L capture input/timer 0H capture input<br>P24 to P27: INT5 input/HOLD release input/timer 1 event input/timer 0L capture input/timer 0H capture input<br>P20: INT6 input/timer 0L capture 1 input/UART1 transmit<br>P21:UART1 receive<br>P22: SIO4 data I/O<br>P23: SIO4 data I/O<br>P24: INT7 input/timer 0H capture 1 input/SIO4 clock I/O<br>P27: D+ 1.5 kΩ pull-up resistor connection pin<br>Interrupt detection mode
<table>
<tr><th></th><th>Rising</th><th>Falling</th><th>Rising &amp; 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<br>P31 to P34</td><td>I/O</td><td>• 4-bit I/O port<br>• I/O can be specified in 1-bit units<br>• Pull-up resistors can be turned on and off in 1-bit units<br>• Pin functions<br>P31: SCUART transmit<br>P32: SCUART receive<br>P34: USB interface PLL filter circuit connection pin</td><td>Yes</td></tr>
<tr><td>Port 7<br>P70 to P73</td><td>I/O</td><td>• 4-bit I/O port<br>• I/O can be specified in 1-bit units<br>• Pull-up resistors can be turned on and off in 1-bit units<br>• Pin functions<br>P70: INT0 input/HOLD release input/timer 0L capture input/ D+ 1.5 kΩ pull-up resistor connection pin<br>P71: INT1 input/HOLD release input/timer 0H capture input<br>P72: INT2 input/HOLD release input/timer 0 event input/timer 0L capture input/high-speed clock counter input<br>P73: INT3 input (with noise filter)/timer 0 event input/timer 0H capture input<br>Interrupt detection mode
<table>
<tr><th></th><th>Rising</th><th>Falling</th><th>Rising &amp; 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<br>PWM1</td><td>I/O</td><td>• PWM0 and PWM1 output port<br>• Pin functions<br>General-purpose input port<br>AD converter input port: AN8, AN9 (PWM1, PWM0)<br>PWM0: High-current P-channel driver<br>PWM1: High-current N-channel driver</td><td>No</td></tr>
<tr><td>D-</td><td>I/O</td><td>USB data I/O pin D- /general-purpose I/O port</td><td>No</td></tr>
<tr><td>D＋</td><td>I/O</td><td>USB data I/O pin D＋/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<br>• Pin functions<br>General-purpose input port<br>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<br>• Pin functions<br>General-purpose I/O port<br>AD converter input port: AN11</td><td>No</td></tr>
<tr><td>CF1</td><td>I</td><td>Ceramic resonator input pin</td><td>No</td></tr>
<tr><td>CF2</td><td>O</td><td>Ceramic resonator output pin</td><td>No</td></tr>
<tr><td>OWP0</td><td>I/O</td><td>Dedicated debugger 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 P07 | Open | Output low |
| P10 to P17 | Open | Output low |
| P20 to P27 | Open | Output low |
| P31 to P34 | Open | Output low |
| P70 to P73 | Open | Output low |
| PWM0, PWM1 | Open | Output low |
| D+, D- | Open | Output low |
| XT1 | Pulled down with a resistor of 100kΩ or less | - |
| XT2 | Open | Output low |
| OWP0 | Pulled down with a resistor of 100kΩ | - |

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

## 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<br>P10 to P17<br>P20 to P27<br>P31 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 |
| D+, D- | – | No | CMOS | No |
| XT1 | – | No | Input only | No |
| XT2 | – | No | 32.768 kHz crystal resonator output<br>(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 P27 | ○ | 1 bit | CMOS |
| | | | | N-channel open drain |
| | P31 to P34 | ○ | 1 bit | CMOS |
| | | | | N-channel open drain |
| Program start address | – | ○ | – | 00000h |
| | | | | 03E00h |
| 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 | ○ | – | 3 levels |
| Power-on reset function | Power-on reset level | ○ | – | 4 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.

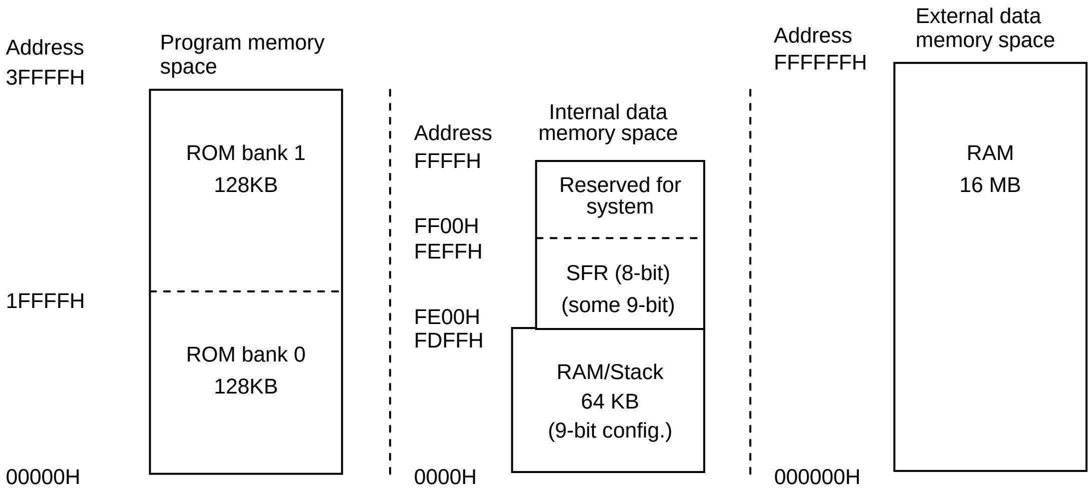
When the reference voltage circuit is activated, the current drain increases by approximately 100 μA compared with that when the reference voltage circuit is inactive.

# 2. Internal Configuration

## 2.1 Memory Space

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

1) Program memory space: 256K bytes (128K bytes × 2 banks)
2) Internal data memory space: 64K bytes (0000H to FDFFH out of 0000H to FFFFH is shared with the stack area.)
3) External data memory space: 16M bytes



*Note: SFR is the area in which special function registers such as the accumulator are allocated (see 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 |
|---|---|---|---|
| Interrupt | Reset (Note) | 00000H | 0 |
| | | 03E00H | 0 |
| | INT0 | 00003H | 0 |
| | INT1 | 0000BH | 0 |
| | INT2/T0L/INT4/USB bus active | 00013H | 0 |
| | INT3/INT5/base timer | 0001BH | 0 |
| | T0H/INT6 | 00023H | 0 |
| | T1L/T1H/INT7 | 0002BH | 0 |
| | SIO0/USB bus reset/USB suspend/UART1 receive end/ SCUART receive end | 00033H | 0 |
| | SIO1/USB endpoint/USB-SOF/SIO4/UART1 buffer empty/UART1 transmit end/SCUART buffer empty/ SCUART transmit end | 0003BH | 0 |
| | ADC/T6/T7 | 00043H | 0 |
| | Port 0/PWM0/ PWM1/T4/T5 | 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]<br>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)<br>PC07 to 00=(SP-1)<br>(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<br>nb: Number of instruction bytes | Unchanged |

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

## 2.3 Program Memory (ROM)

This series of microcontrollers has a program memory space of 256K bytes, but the size of the ROM that is actually incorporated in the microcontroller varies with the type of microcontroller. The ROM table look-up instruction (LDC) can be used to reference all ROM data within the bank. Of the ROM space, the 256 bytes in ROM bank 0 (1FF00H 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.

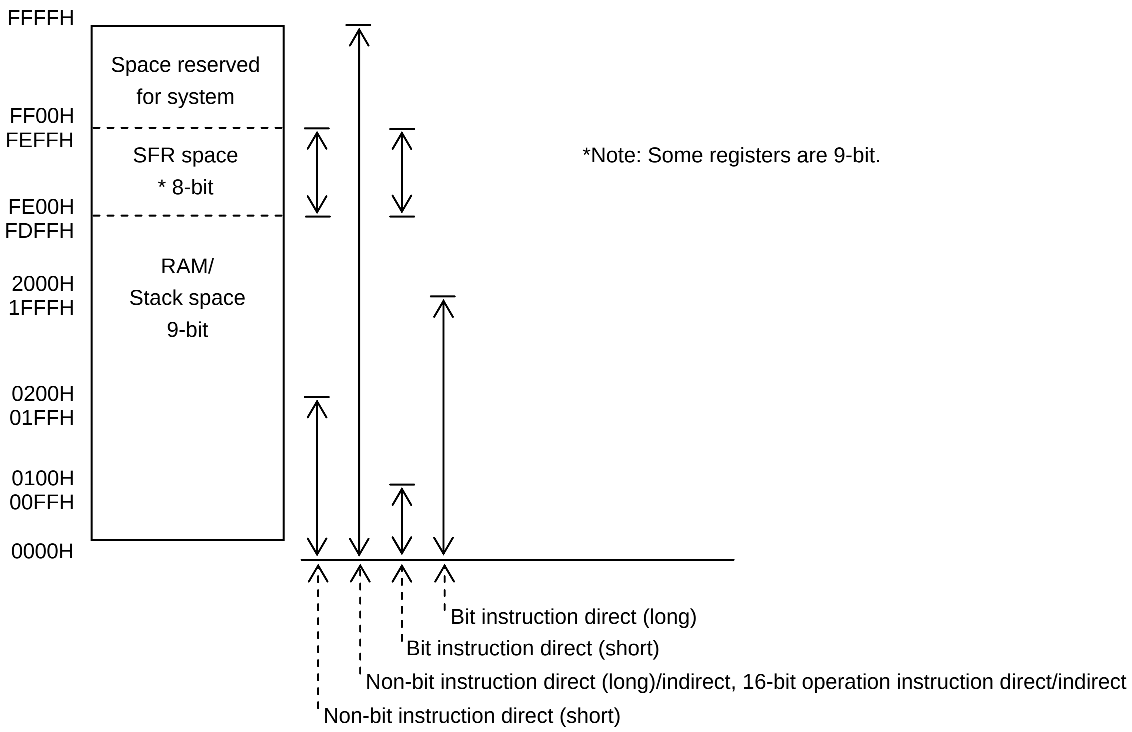

FFFFH
FF00H
FEFFH
FE00H
FDFFH
2000H
1FFFH
0200H
01FFH
0100H
00FFH
0000H
Space reserved for system
SFR space
* 8-bit
RAM/
Stack space
9-bit
*Note: Some registers are 9-bit.
Bit instruction direct (long)
Bit instruction direct (short)
Non-bit instruction direct (long)/indirect, 16-bit operation instruction direct/indirect
Non-bit instruction direct (short)

**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.

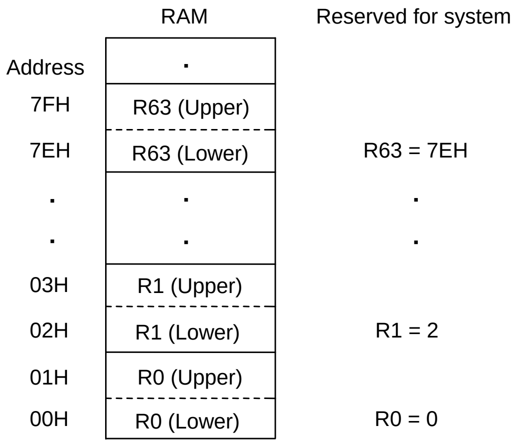


**Figure 2.10.1 Allocation of Indirect Registers**

## 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 ≤ n ≤ 63)
3) Indirect register (Rn) + C register indirect (0 ≤ n ≤ 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 performed on each 8-bit data transfer.
3) When transmission or reception of a data packet is performed in the USB interface circuit, a wait request occurs and 1 cycle of wait operation (RAM data transfer) is performed on each 4-byte data transfer.

## 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,<br>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/<br>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 consists 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.

### 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

P00, P02 also serves as the high-current N-channel driver output, P01 as the high-current P-channel driver output, P05 as the system clock output, P06 as the timer 6 toggle output, P07 as the timer 7 toggle output, 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 |

#### P0FCR7 (bit 7): Fixed bit

This bit must always be set to 0.

#### P0FCR6 (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 RC clock |
| 1 | 0 | USB frequency-divided clock |
| 1 | 1 | CF clock |

### 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 SCOSL5 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.

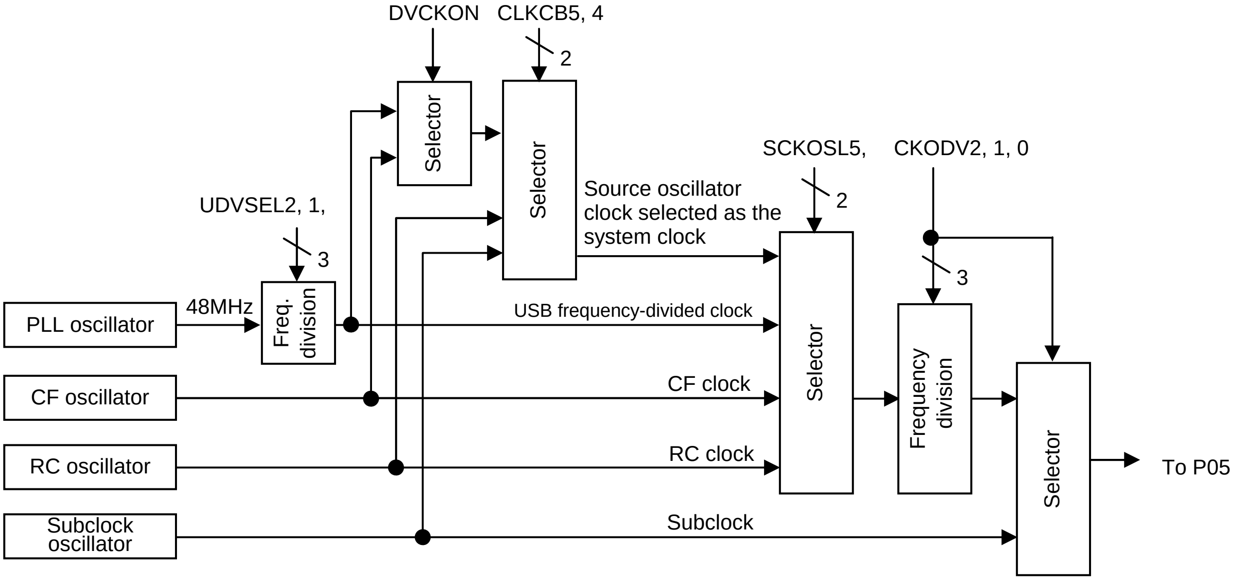


**Figure 3.1.1 P05 Output Clock Selection**

### 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 consists 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 pin 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.

### 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 8-bit I/O port equipped with programmable pull-up resistors. It consists 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 (8 bits: P20 to P27)

- 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 to P27 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 (main oscillation by 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.

4) Multiplexed pin function

P27 also serves as the D+ 1.5 kΩ pull-up resistor connection pin, P24 to 22 as SIO4 I/O, and P21, P20 as UART1 I/O. P20 to P27 also serve as analog input channel AN12 to AN19. Refer to the description in the chapter on each functional block.

| 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 |
| FE67 | 0000 0000 | R/W | P2INH | P27INH | P26INH | P25INH | P24INH | P23INH | P22INH | P21INH | P20INH |

## 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 output data from port 2 and pull-up resistors.
2) When this register is read with an instruction, data at pins P20 to P27 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 can always be read regardless of the I/O state of the port.

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

### 3.3.3.2 Port 2 data direction register (P2DDR)

1) This register is an 8-bit register that controls the I/O direction of port 2 in 1-bit units. Port P2n functions as an output mode when bit P2nDDR is set to 1 and functions as input mode when bit P2nDDR is set to 0.
2) When bit P2nDDR is set to 0 and bit P2n of the port 2 data latch is set to 1, port P2n becomes an input with a pull-up resistor.

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

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

### 3.3.3.3 Port 2 input disable control register (P2INH)

1) This register is an 8-bit register that control the function to disable port 2 input in 1-bit units.
2) When port 2 is to be used as analog input ports for an AD converter, set the corresponding bits of this register to 1. The digital input is disabled, and the through current flowing in the digital input circuit can be prevented.
3) When P2nINH bits are set to 1, the corresponding P2n ports are always read as 1.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE67 | 0000 0000 | R/W | P2INH | P27INH | P26INH | P25INH | P24INH | P23INH | P22INH | P21INH | P20INH |

### 3.3.3.4 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.

### 3.3.3.5 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 | Port P26 |
| 1 | 1 | Port P27 |

**I5SL1 (bit 5): INT5 pin function select**

**I5SL0 (bit 4): INT5 pin function select**

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

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

**I4SL3 (bit 3): INT4 pin select**

**I4SL2 (bit 2): INT4 pin select**

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

**I4SL1 (bit 1): INT4 pin function select**

**I4SL0 (bit 0): INT4 pin function select**

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

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

*Notes:*

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

2) *If INT4 and INT5 are specified 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.*

### 3.3.3.6 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 4-bit I/O port equipped with programmable pull-up resistors. It consists 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 (4 bits: P31 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.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE4C | HHH0 000H | R/W | P3 | - | - | - | P34 | P33 | P32 | P31 | - |
| FE4D | HHH0 000H | R/W | P3DDR | - | - | - | P34DDR | P33DDR | P32DDR | P31DDR | - |

## 3.4.3 Related Registers

### 3.4.3.1 Port 3 data latch (P3)

1) This data latch is a 4-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 P31 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 000H | R/W | P3 | - | - | - | P34 | P33 | P32 | P31 | - |

### 3.4.3.2 Port 3 data direction register (P3DDR)

1) This register is a 4-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 000H | R/W | P3DDR | - | - | - | P34DDR | P33DDR | P32DDR | P31DDR | - |

| 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 consists 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 (main oscillation by 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.

7) Multiplexed pin function

P70 also serves as the D+ 1.5 kΩ pull-up resistor connection pin.

| | 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.

### 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, 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 | Subclock |
| 1 | 1 | Timer/counter 0 prescaler output |

**BUZON (bit 3): Buzzer output select**

When P17FCR (P1FCR, bit 7) 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 P27 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 P27 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 P27 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 P27 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 P27 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.

T0 period = ([T0HR, T0LR] + 1) × (T0PRR +1) × 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 P27 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.

T0 period = [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 P27, 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 P27 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 P27 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 |

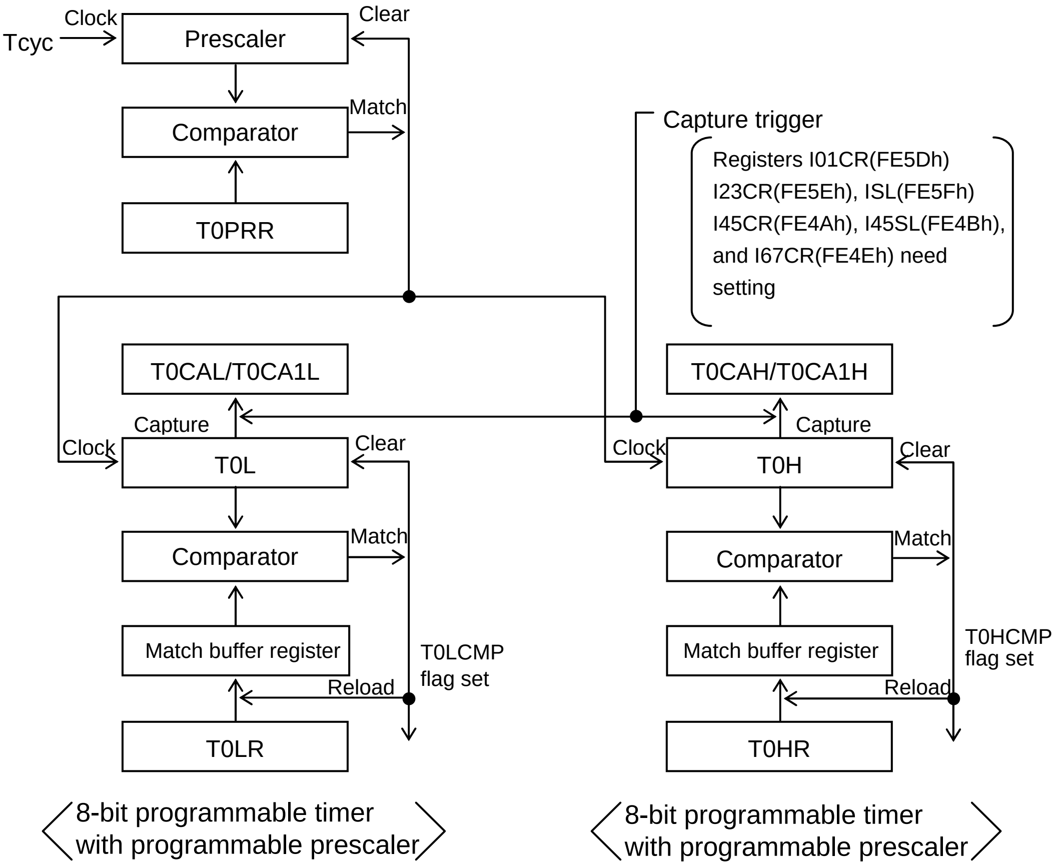


**Figure 3.6.1 Mode 0 Block Diagram (T0LONG =0 , T0LEXT = 0)**

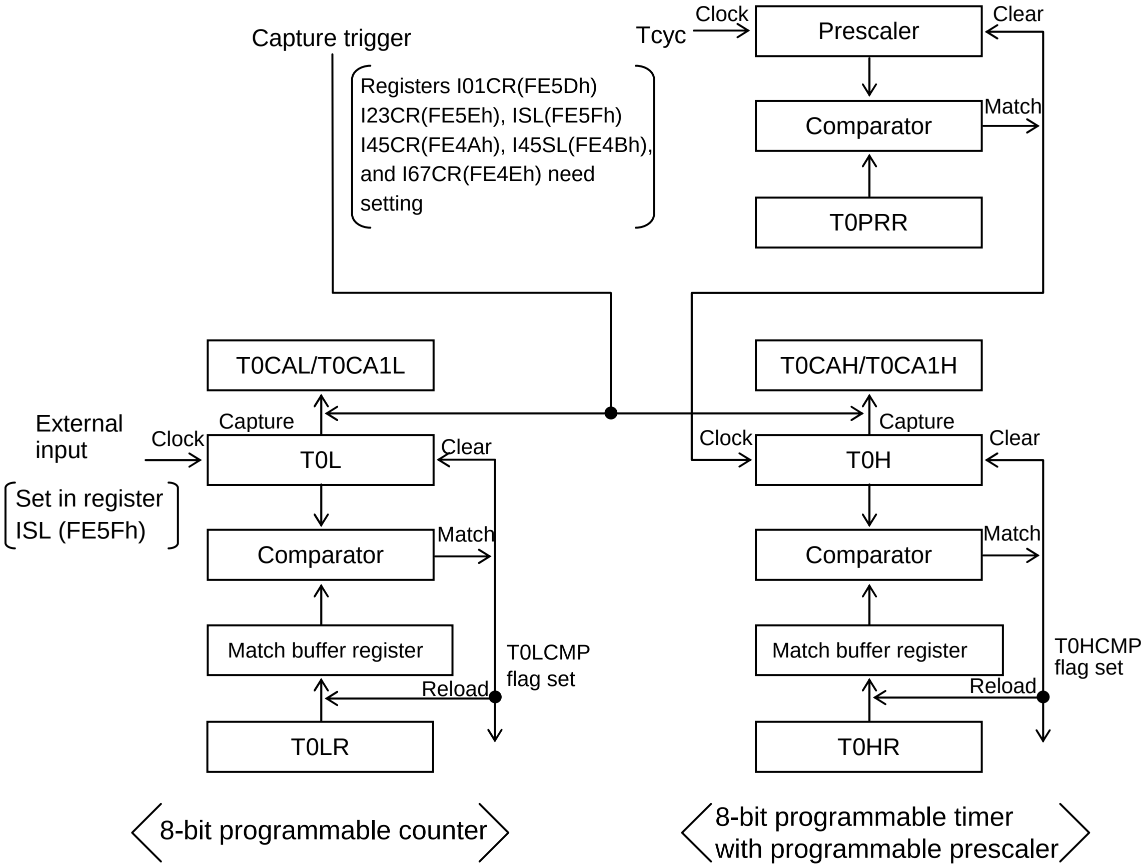


**Figure 3.6.2 Mode 1 Block Diagram (T0LONG = 0, T0EXT = 1)**

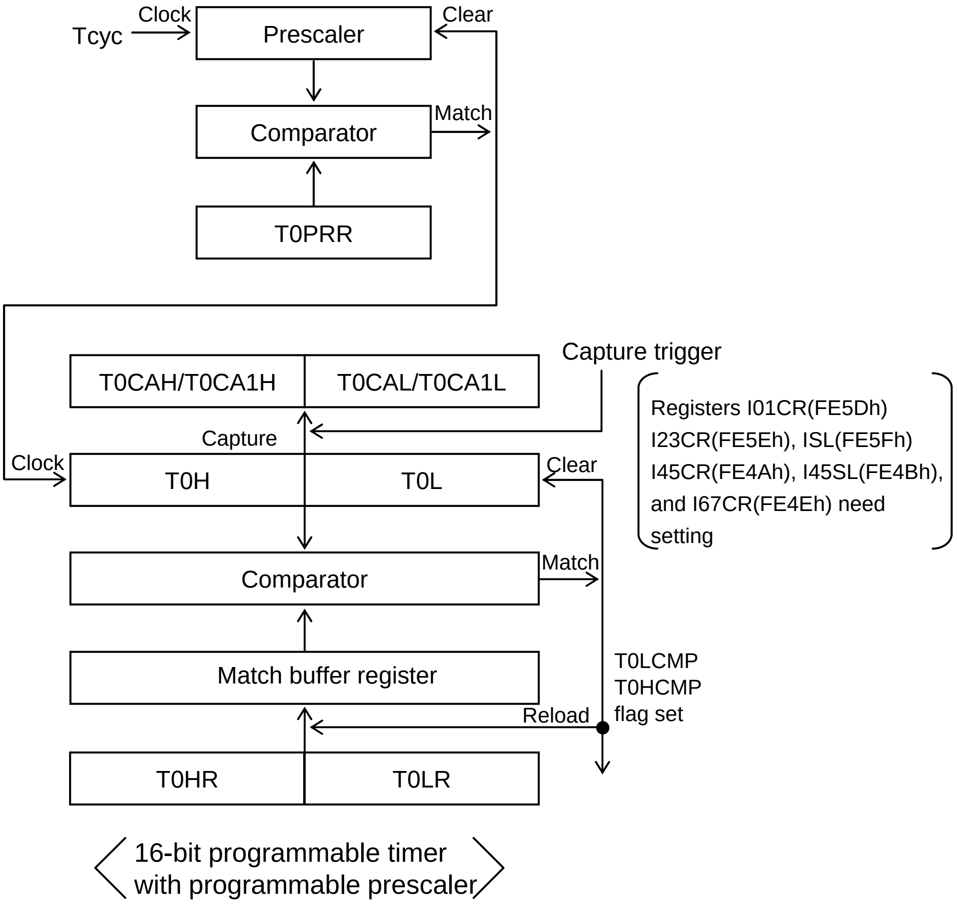


**Figure 3.6.3 Mode 2 Block Diagram (T0LONG = 1, T0LEXT = 0)**

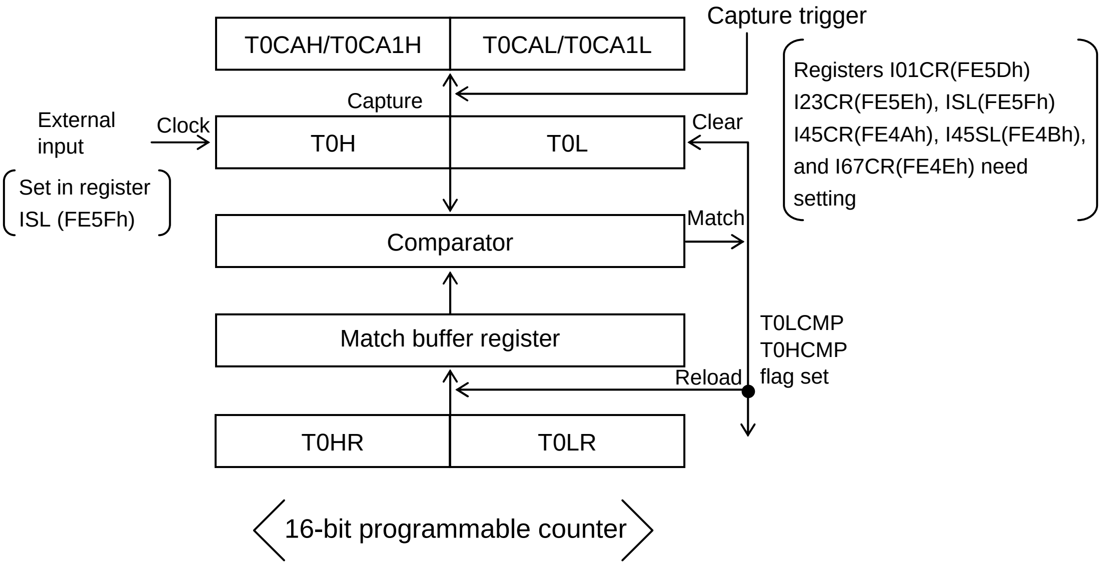

Capture trigger
T0CAH/T0CA1H
T0CAL/T0CA1L
Registers I01CR(FE5Dh) I23CR(FE5Eh), ISL(FE5Fh) I45CR(FE4Ah), I45SL(FE4Bh), and I67CR(FE4Eh) need setting
Capture
External input
Set in register ISL (FE5Fh)
Clock
T0H
T0L
Clear
Comparator
Match
Match buffer register
T0LCMP
T0HCMP
flag set
Reload
T0HR
T0LR
16-bit programmable counter

**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, bit 4) must be set to 1 when a high-speed clock counter is to be used.

When NKEN=1 and T0LONG (T0CNT register, bit 5)=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, bit 5)=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.

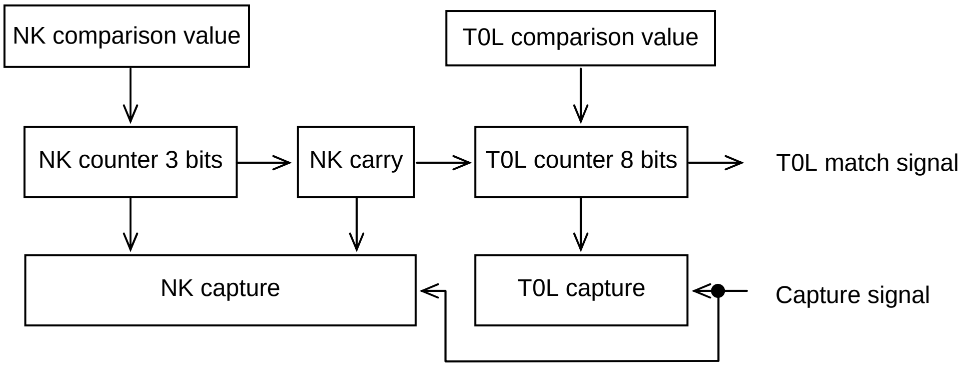


**Figure 3.7.1 T0LONG = 0 Block Diagram (Timer 0: 8-bit mode)**

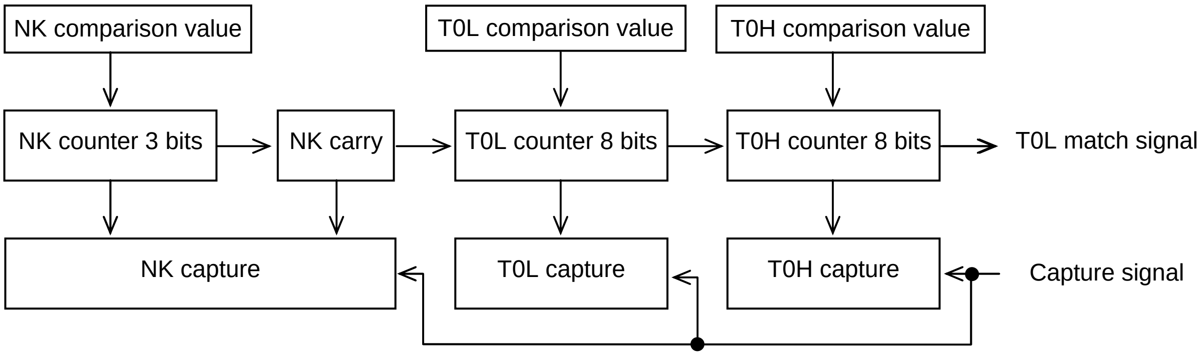


**Figure 3.7.2 T0LONG = 1 Block Diagram (Timer 0: 16-bit mode)**

# 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.

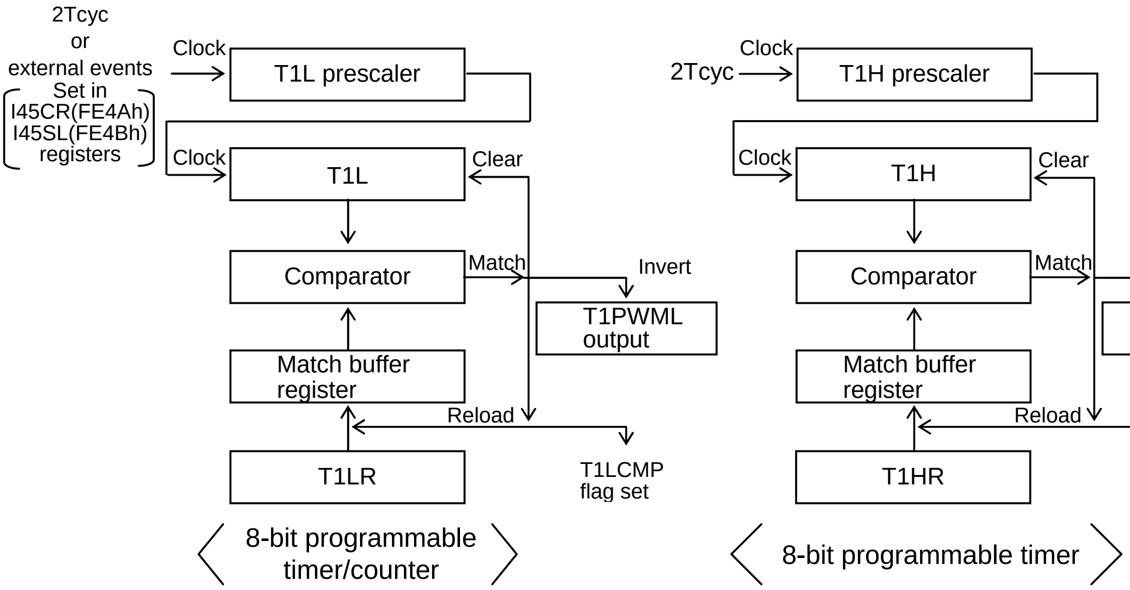


**Figure 3.8.1 Mode 0 Block Diagram (T1LONG = 0, T1PWM = 0)**

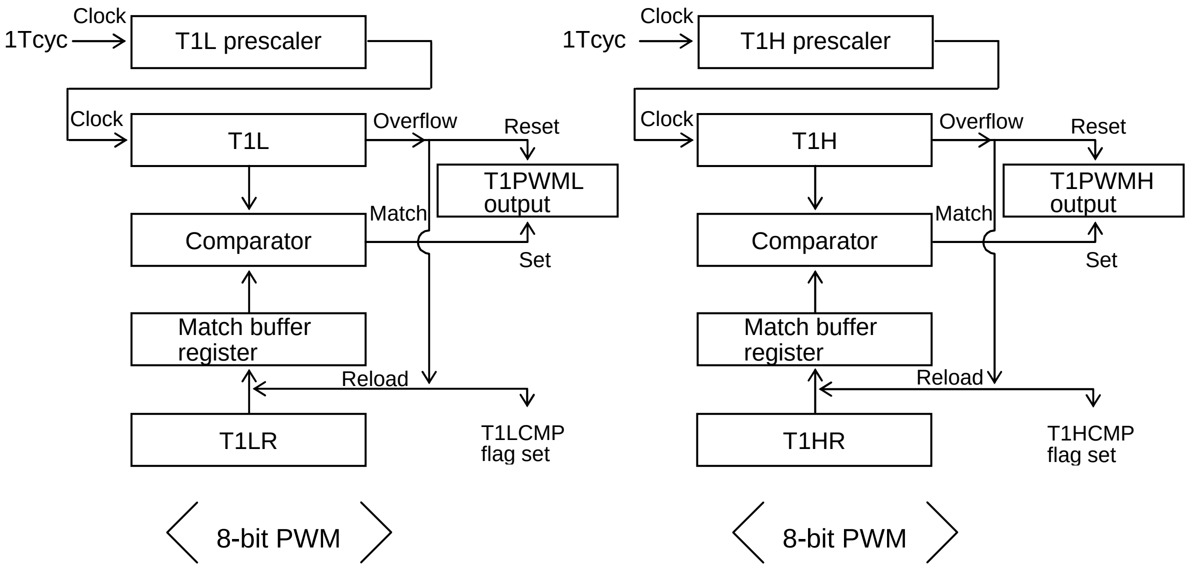


**Figure 3.8.2 Mode 1 Block Diagram (T1LONG = 0, T1PWM = 1)**

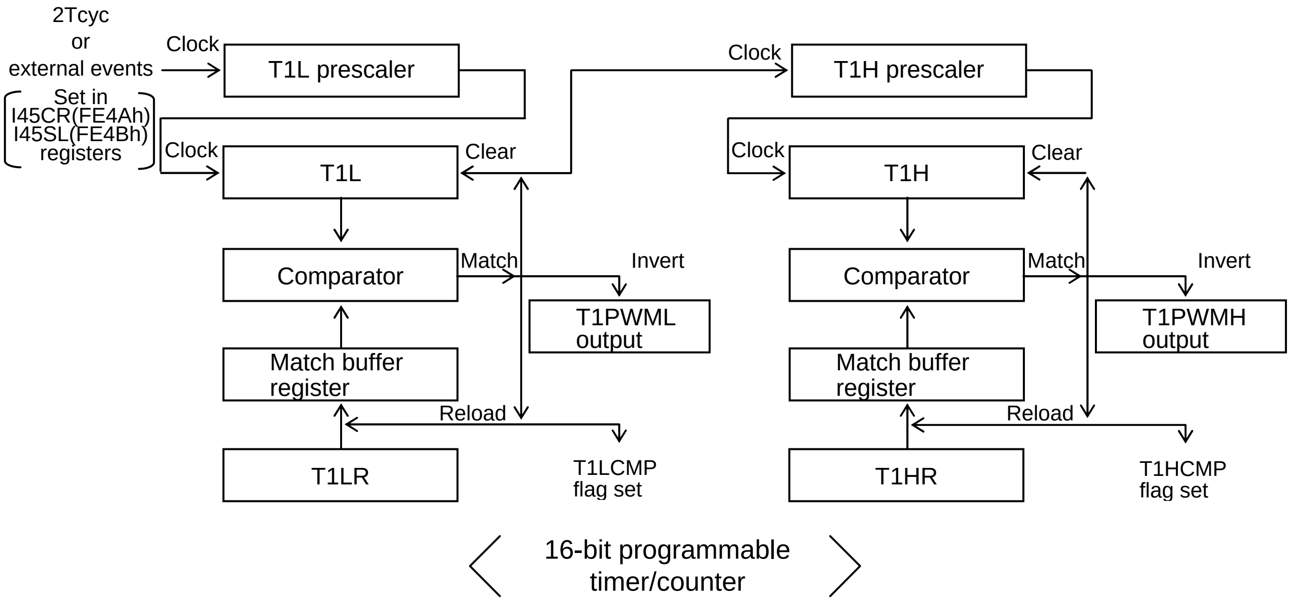


**Figure 3.8.3 Mode 2 Block Diagram (T1LONG = 1, T1PWM = 0)**

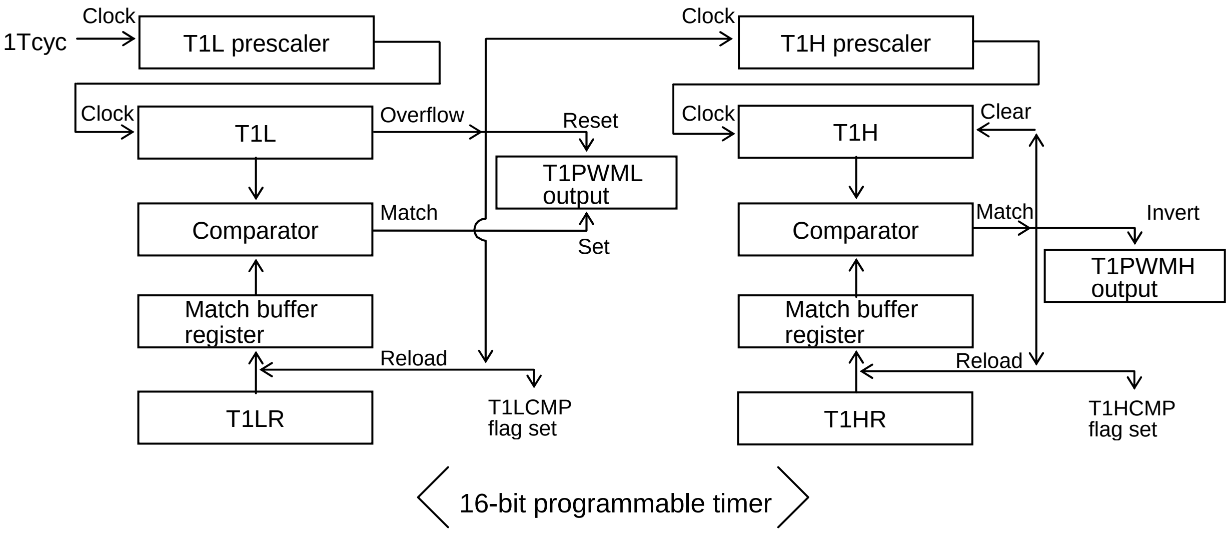


**Figure 3.8.4 Mode 3 Block Diagram (T1LONG = 1, T1PWM = 1)**

## 3.8.4 Related Registers

### 3.8.4.1 Timer 1 control register (T1CNT)

1) This 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 values match 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<br>or | Period: {(T1LR+1) × (T1LPRC count) × 2Tcyc} × 2<br>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<br>or | Period: {(T1HR+1) × (T1HPRC count) × (T1LR+1) × (T1LPRC count) × 2 Tcyc} ×2<br>Period: {(T1HR+1) × (T1HPRC count) × (T1LR+1) × (T1LPRC count) × events} × 2 | Toggle output<br>or | Period: {(T1LR+1) × (T1LPRC count) × 2Tcyc} × 2<br>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)

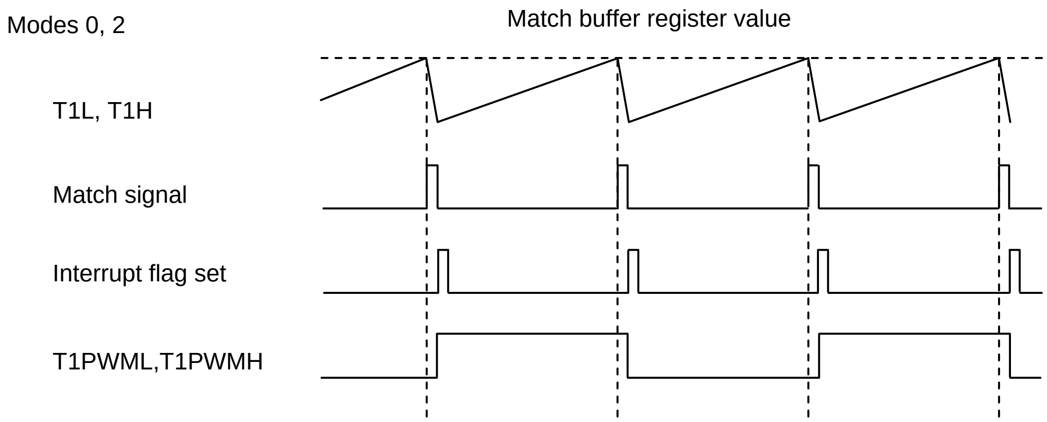
1) This register is used to store the match data for T1H. It has an 8-bit match buffer register. A match signal is generated when the value of this match buffer register matches the value of timer 1 high byte.

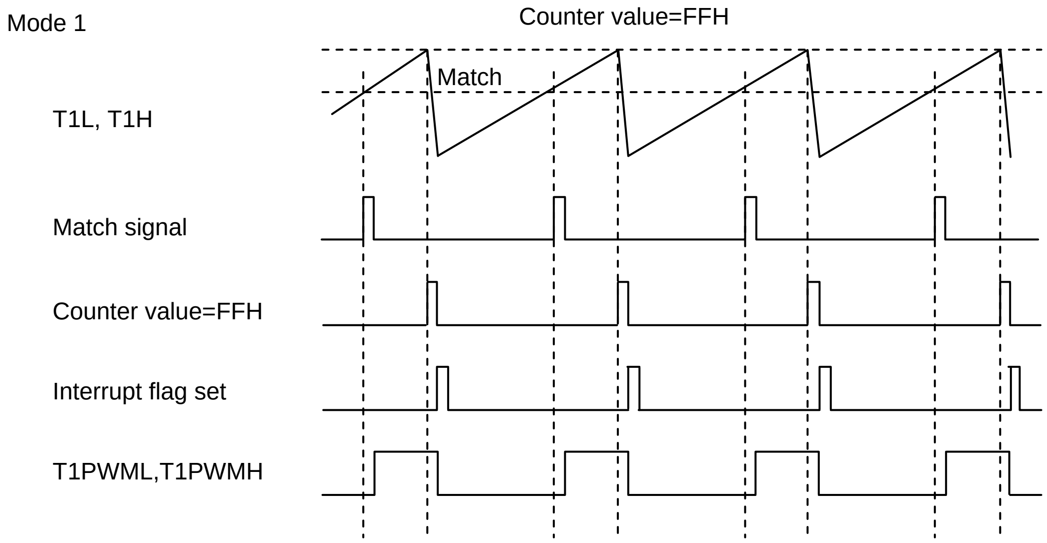
2) The match buffer register is updated as follows:

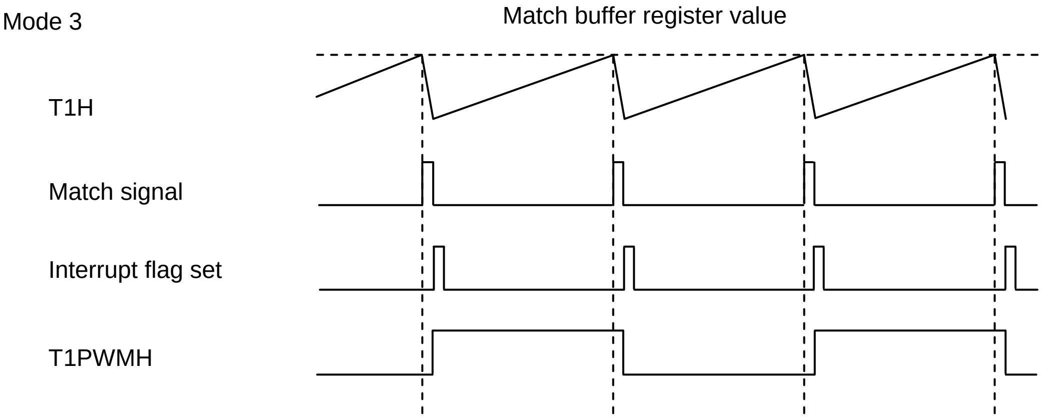
- When it is inactive (T1HRUN=0), the match buffer register matches T1HR.
- When it is active (T1HRUN=1), the match buffer register is loaded with the contents of T1HR when the value of T1H reaches 0.

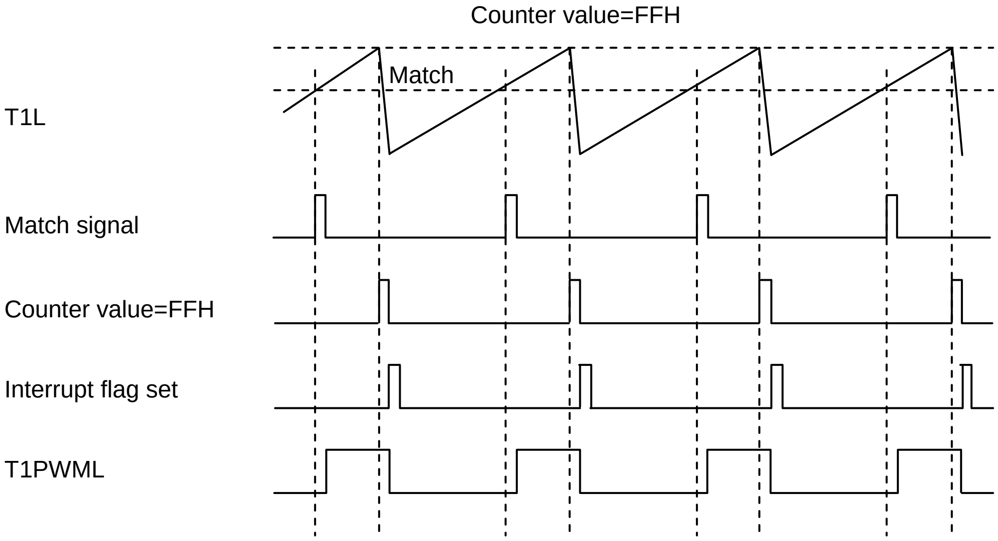
| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE1D | 0000 0000 | R/W | T1HR | T1HR7 | T1HR6 | T1HR5 | T1HR4 | T1HR3 | T1HR2 | T1HR1 | T1HR0 |

**T1**


Modes 0, 2
Match buffer register value
T1L, T1H
Match signal
Interrupt flag set
T1PWML,T1PWMH


Mode 1
Counter value=FFH
Match
T1L, T1H
Match signal
Counter value=FFH
Interrupt flag set
T1PWML,T1PWMH


Mode 3
Match buffer register value
T1H
Match signal
Interrupt flag set
T1PWMH


Counter value=FFH
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 4 Tcyc, 16 Tcyc, or 64 Tcyc clock.

T4 period = (T4R+1) × $4^n$ Tcyc (n=1, 2, 3)

Tcyc = Period of cycle clock

2) Timer 5 (T5)

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

T5 period = (T5R+1) × $4^n$ Tcyc (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 clock following the one 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 clock following the one 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..

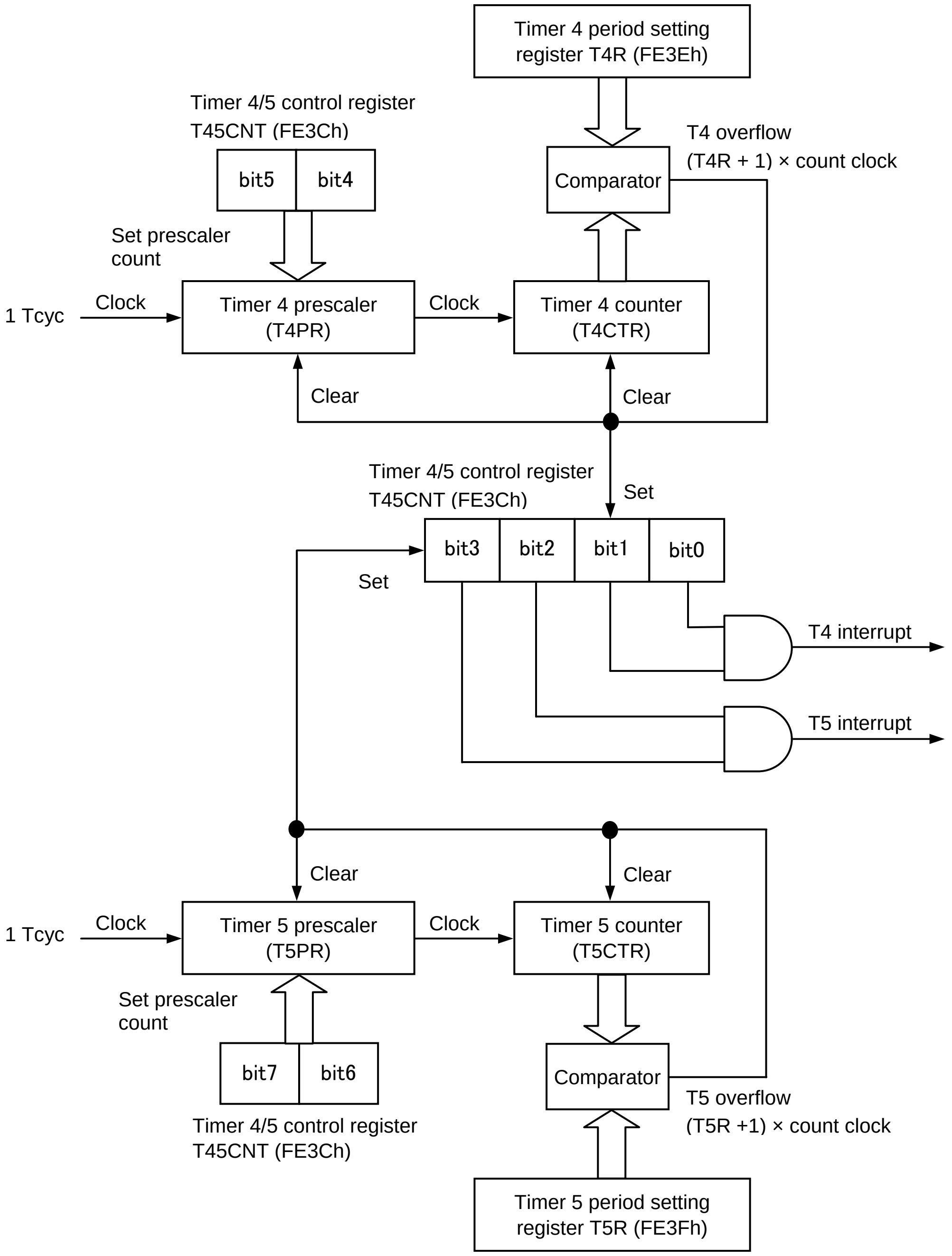


**Figure 3.9.1 Timer 4/5 Operation Block Diagram**

## 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 4 Tcyc, 16 Tcyc, or 64 Tcyc clock. It can generate toggle waveforms with the period of timer 6 at pin P06.

T6 period = (T6R+1) × $4^n$ Tcyc (n=1, 2, 3)

Tcyc = Period of cycle clock

2) Timer 7 (T7)

Timer 7 is an 8-bit timer that runs on either 4 Tcyc, 16 Tcyc, or 64 Tcyc clock. It can generate toggle waveforms with the period of timer 7 at pin P07.

T7 period = (T7R+1) × $4^n$ Tcyc (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 |
| FE4F | 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 clock following the one 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 clock following the one 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.

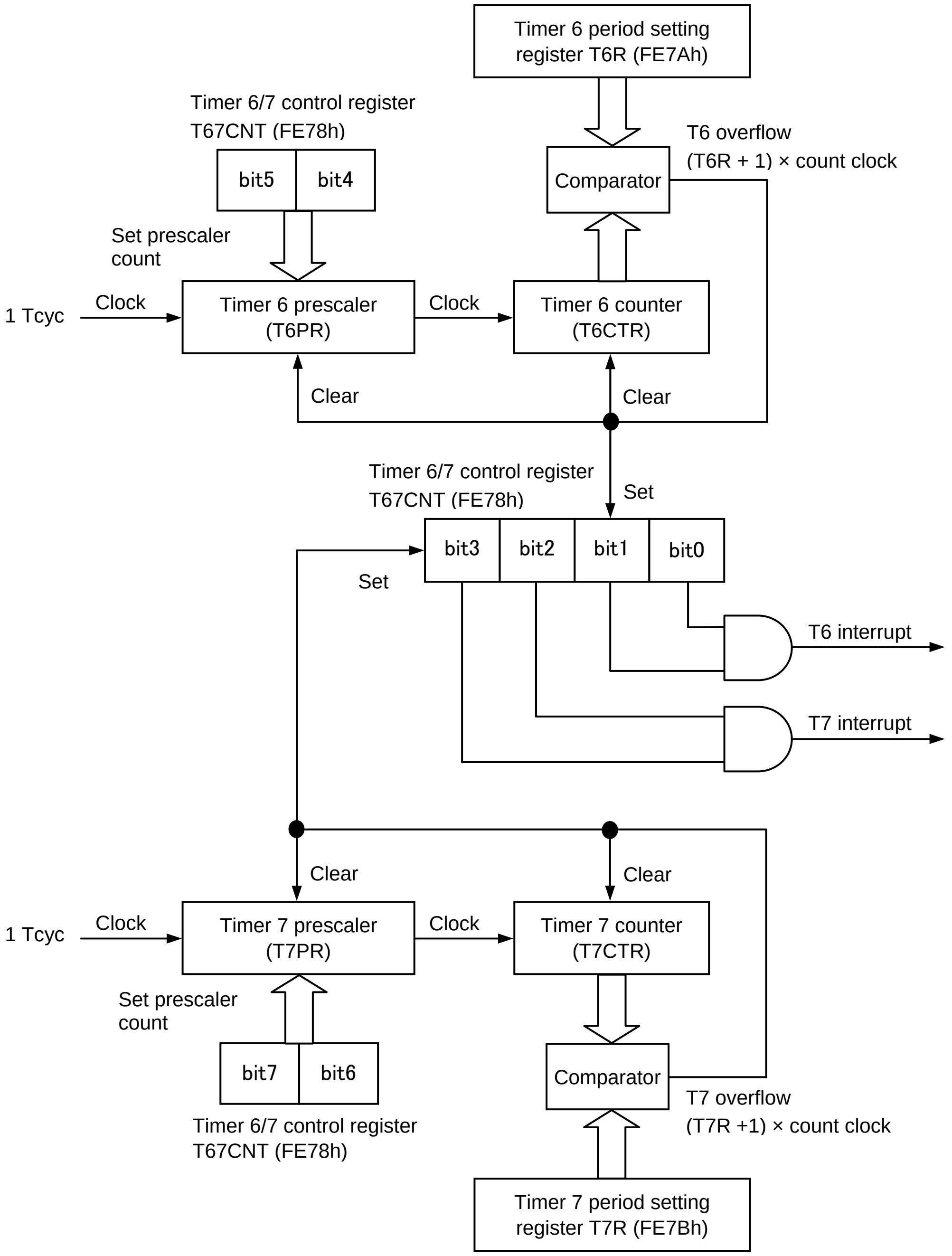


**Figure 3.10.1 Timer 6/7 Operation Block Diagram**

## 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):**

These two 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.

**CLKOEN (bit 3):**

**CKODV2 (bit 2):**

**CKODV1 (bit 1):**

**CKODV0 (bit 0):**

These four 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
3) High-speed mode (when used as a 6-bit base timer)
4) Buzzer output
5) 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. In this case, one of the three clocks (cycle clock, timer/counter 0 prescaler output, and subclock) must be loaded in the input signal select register (ISL) as the base timer count clock.

2) 14-bit binary up-counter

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

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

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

4) Buzzer output function

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

5) Interrupt generation

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

6) HOLD mode operation and HOLD mode release function

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

7) It is necessary to manipulate the following special function registers to control the base timer.

- BTCR, ISL
- P1, P1DDR, P1FCR

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

## 3.11.3 Circuit Configuration

### 3.11.3.1 8-bit binary up-counter

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

### 3.11.3.2 6-bit binary up-counter

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

### 3.11.3.3 Base timer input clock source

1) The clock input to the base timer (fBST) can be selected from among the cycle clock, timer 0 prescaler, and subclock via the input signal select register (ISL).

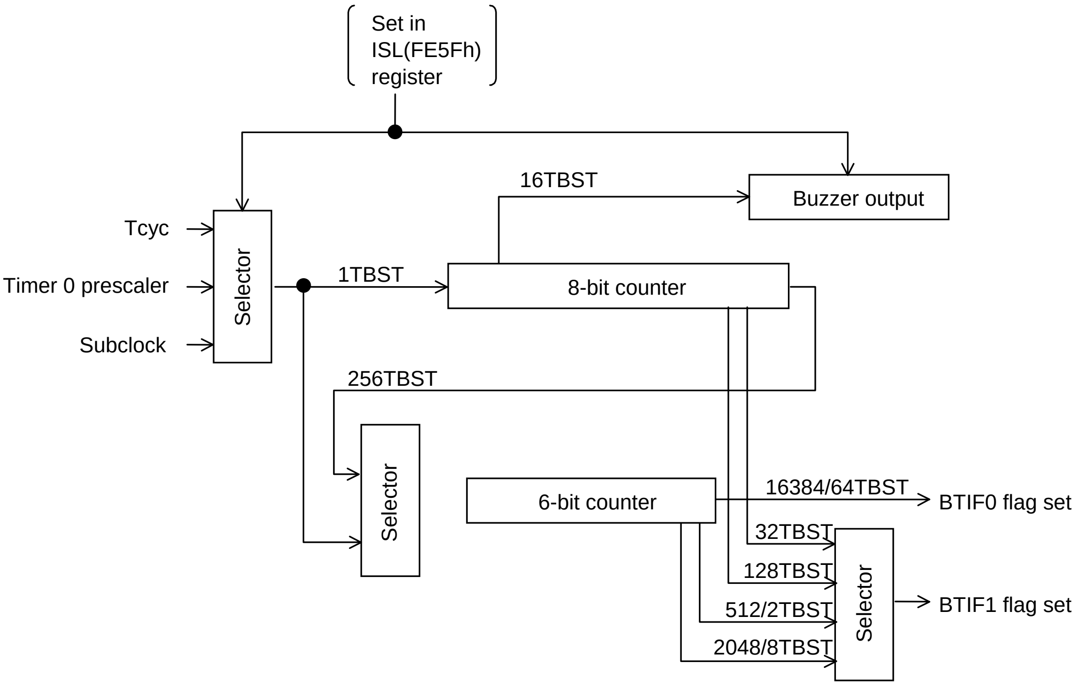


**Figure 3.11.1 Base Timer Block Diagram**

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

**BTON (bit 6): Base timer operation control**

When this bit is set to 0, the base timer stops when the count value reaches 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 |
| 1 | 0 | 0 | 64TBST | 32TBST |
| 0 | 0 | 1 | 16384TBST | 128TBST |
| 1 | 0 | 1 | 64TBST | 128TBST |
| 0 | 1 | 0 | 16384TBST | 512TBST |
| 0 | 1 | 1 | 16384TBST | 2048TBST |
| 1 | 1 | 0 | 64TBST | 2TBST |
| 1 | 1 | 1 | 64TBST | 8TBST |

**BTIF1 (bit 3): Base timer interrupt 1 flag**

This flag is set at the interval 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**

When this bit and BTIF1 are set to 1, X'tal HOLD mode release signal and interrupt request to vector address 001BH are generated.

**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, BTC11, and BTC10.

This flag must be cleared with an instruction.

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

When this bit and BTIF0 are set to 1, X'tal HOLD mode release signal and interrupt request to vector address 001BH are generated.

*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:*

  *Period of cycle clock (Tcyc)＜ Base timer interrupt period ÷ 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 HOLD mode is entered while running the base timer when the cycle clock or subclock is selected as the base timer clock source, the base timer is subject to the influence of unstable oscillations caused by the main clock and subclock when they are started following the release of HOLD mode, resulting in an erroneous count from the base timer. When entering HOLD mode, therefore, it is recommended that the base timer be stopped.*

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

### 3.11.4.2 Input signal select register (ISL)

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

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

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

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

These two 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 |
| 0 | 1 | Cycle clock |
| 1 | 0 | Subclock |
| 1 | 1 | Timer/counter 0 prescaler output |

### BUZON (bit 3): Buzzer output select

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

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

When this bit is set to 0, the buzzer output becomes fixed-high.

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

**NFSEL (bit 2): Noise filter time constant select**

**NFON (bit 1): Noise filter time constant select**

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

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

2) Continuous data transmission/reception

- Transmits and receives bit streams whose length is variable in 1-bit units between 1 and 256 bits. Transfer is carried out in clock synchronization mode. The clock may be an internal or external clock .
  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.

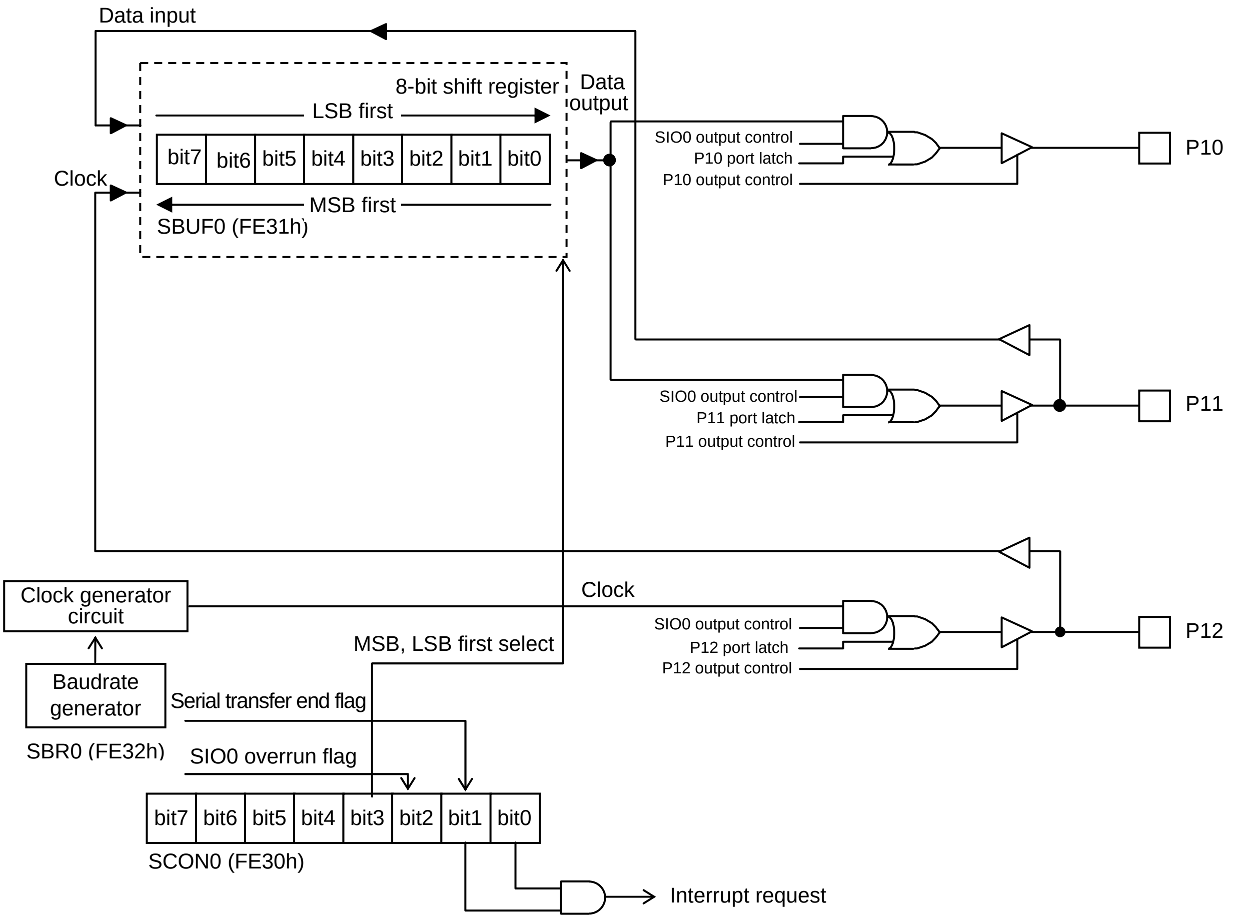


**Figure 3.12.1 SIO0 Synchronous 8-bit Serial I/O Block Diagram (SI0CTR=0)**

## SIO0

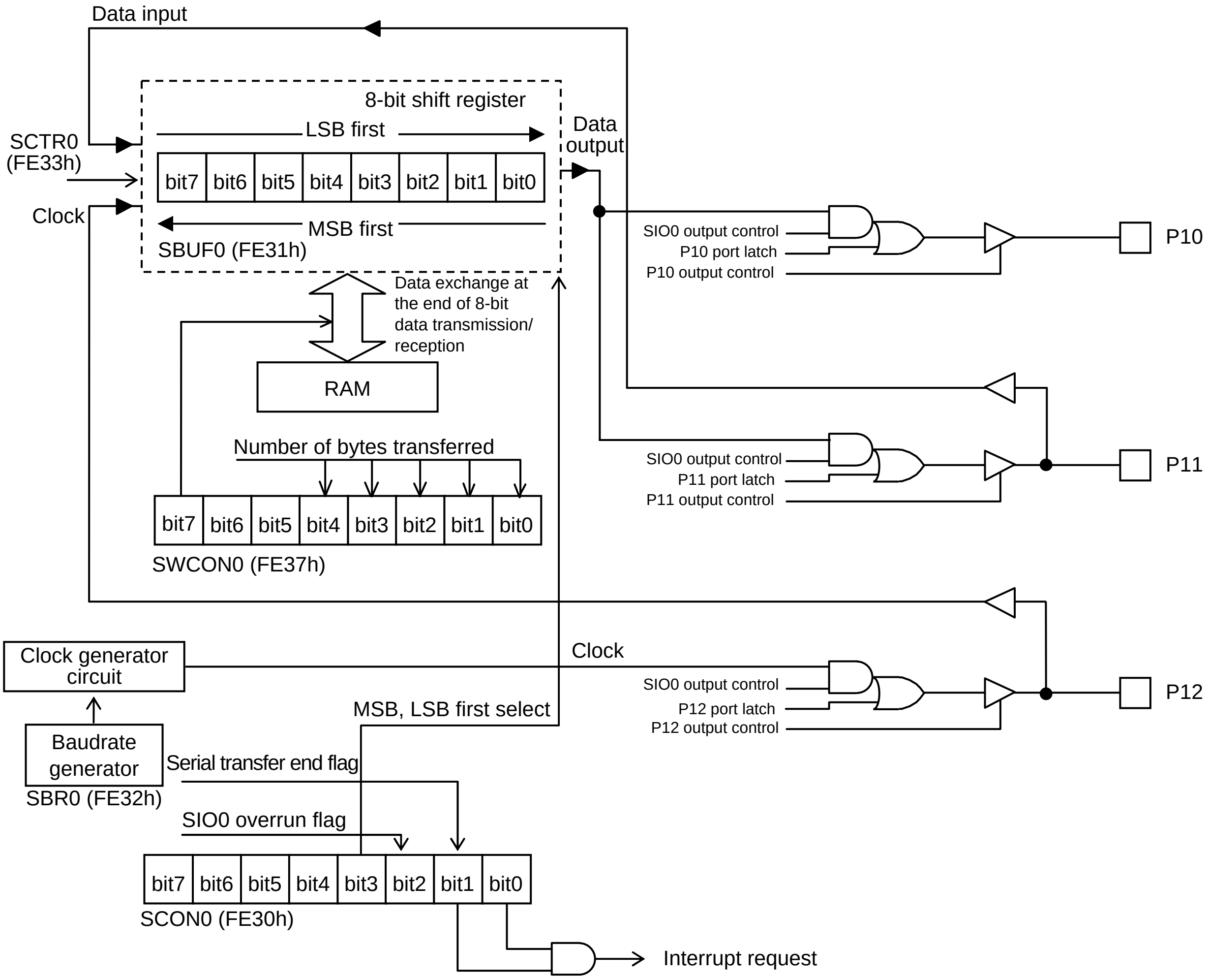


**Figure 3.12.2 SIO0 Continuous Data Transmission/Reception Mode Block Diagram (SI0CTR=1)**

## 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<br>SI1REC=0 | Receive<br>SI1REC=1 | Transfer<br>SI1REC=0 | Receive<br>SI1REC=1 | Transfer<br>SI1REC=0 | Receive<br>SI1REC=1 | Transfer<br>SI1REC=0 | Receive<br>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<br>(Shift data) | 8<br>(All 1’s) | 8<br>(Shift data) | 8<br>(All 1’s) | 8<br>(Shift data) | 8<br>(All 1’s) | 8<br>(Shift data) | 8<br>(All 1’s) |
| Data input | | 8<br>(Input pin) | ← | 8<br>(Input pin) | ← | 8<br>(Input pin) | ← | 8<br>(Input pin) | ← |
| Stop bit | | None | ← | Output (High) | Input (H/L) | Input (H/L) | Output (SBUF1,bit8) | Input (H/L) | Output (L) |
| Clock | | 8 | ← | 9<br>(Internal) | ← | 9 | ← | Low output on falling edge of 8th clock | ← |
| Operation start | | SI1RUN ↑ | ← | 1)<br>SI1RUN ↑<br>2)<br>Start bit detected | Start bit detected | 1)<br>No start bit on falling edge of SI1END when SI1RUN=1<br>2)<br>With start bit on rising edge of SI1RUN when SI1END=0 | 1) on left side | 1) on right side | 1)<br>Clock released on falling edge of SI1END when SI1RUN=1<br>2)<br>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)<br>Instruction<br>2)<br>Start bit detected | Start bit detected | Instruction | Already set | Already set | Start bit detected |
| | Clear | End of processing | ← | End of stop bit | ← | 1)<br>Stop condition detected<br>2)<br>When arbitration lost (Note 1) | ← | 1)<br>Stop condition detected<br>2)<br>Ack=1 detected | ← |
| SI1END (bit 1) | Set | End of processing | ← | End of stop bit | ← | 1)<br>Rising edge of 9th clock<br>2)<br>Stop condition detected | ← | 1)<br>Falling edge of 8th clock<br>2)<br>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)<br>Falling edge of clock detected when SI1RUN=0<br>2)<br>SI1END set conditions met when SI1END=1 | ← | 1)<br>Falling edge of clock detected when SI1RUN=0<br>2)<br>SI1END set conditions met when SI1END=1 | ← | 1)<br>SI1END set conditions met when SI1END=1 | ← | 1)<br>Falling edge of clock detected when SI1RUN=0<br>2)<br>SI1END set conditions met when SI1END=1<br>3)<br>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 | ← |

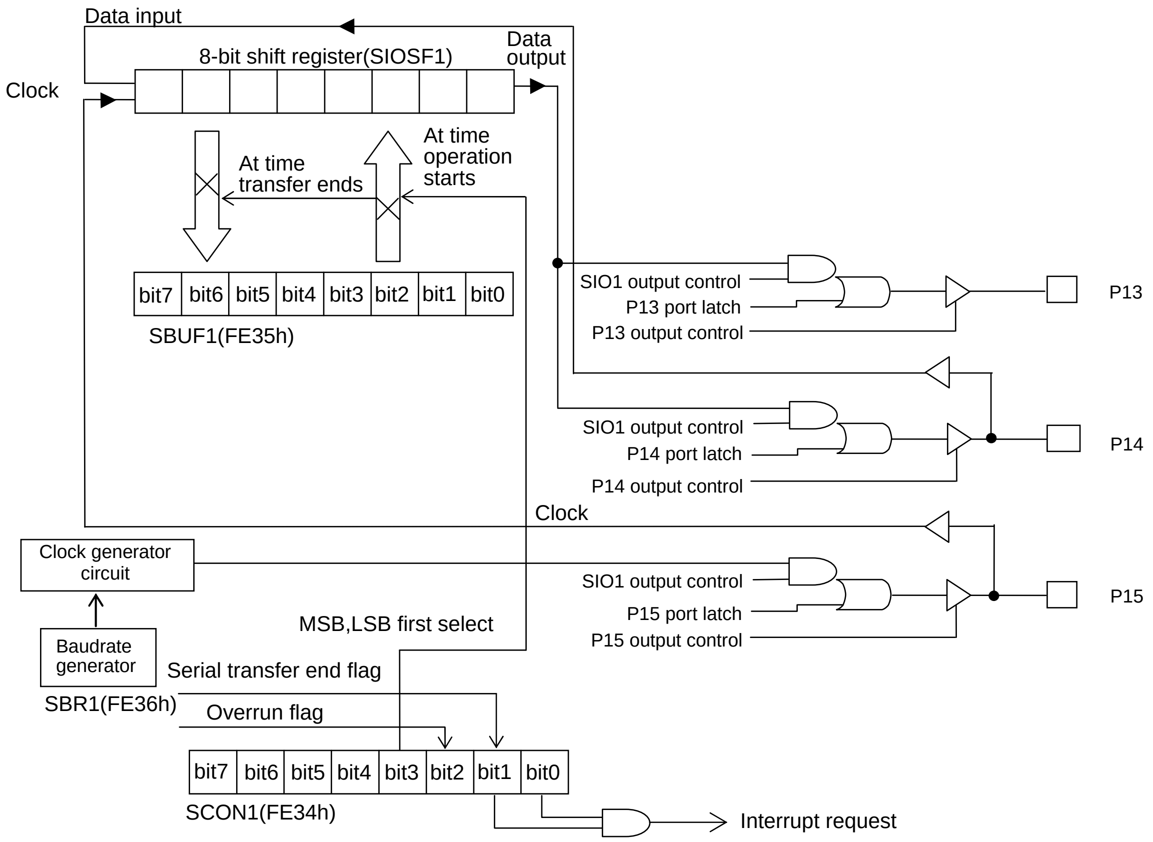


**Figure 3.13.1 SIO1 Mode 0: Synchronous 8-bit Serial I/O Block Diagram (SI1M1=0, SI1M0=0)**

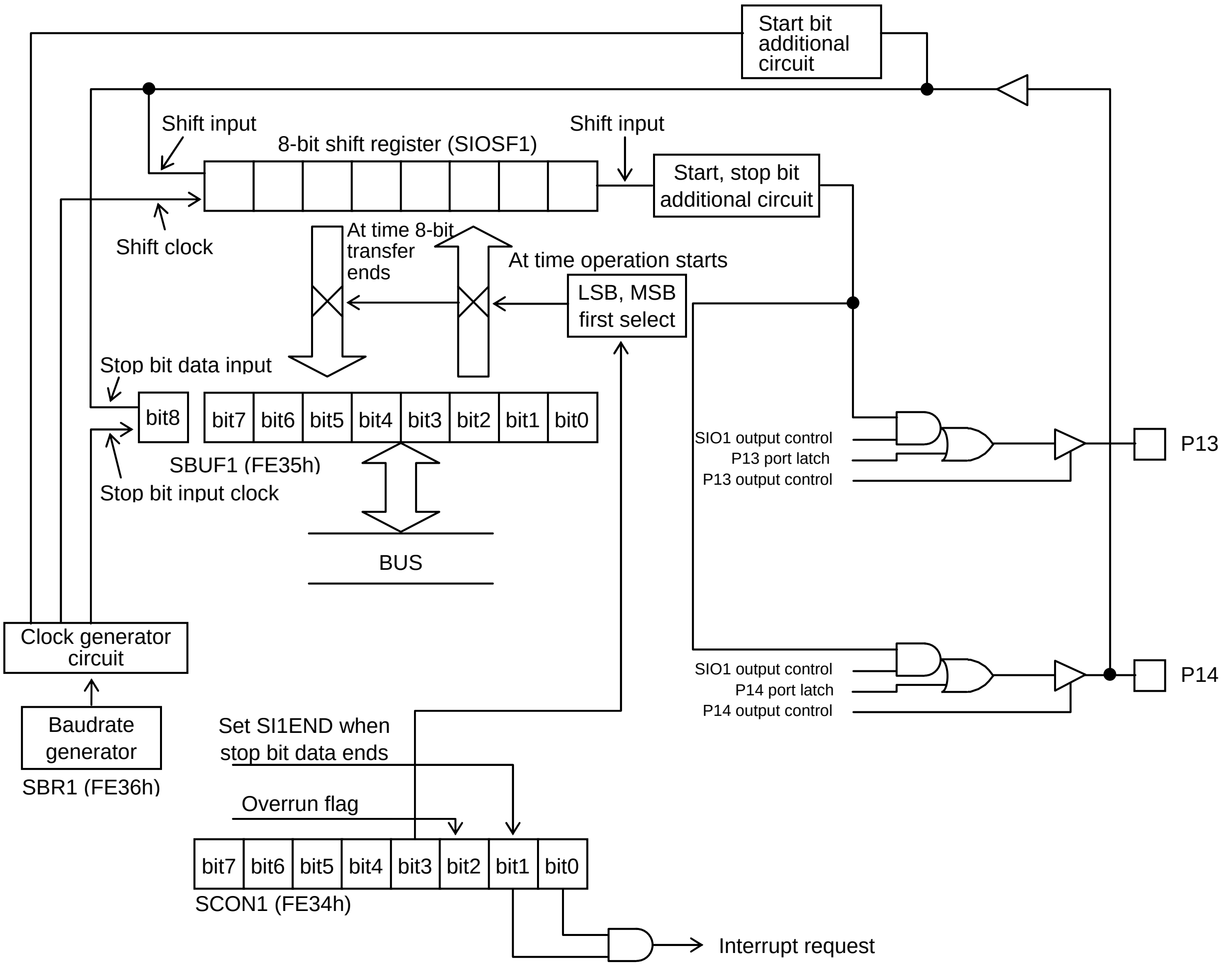


**Figure 3.13.2 SIO1 Mode 1: Asynchronous Serial [UART] Block Diagram (SI1M1=0, SI1M0=1)**

## 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 1024 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 1024 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

**SIO4**

| 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.

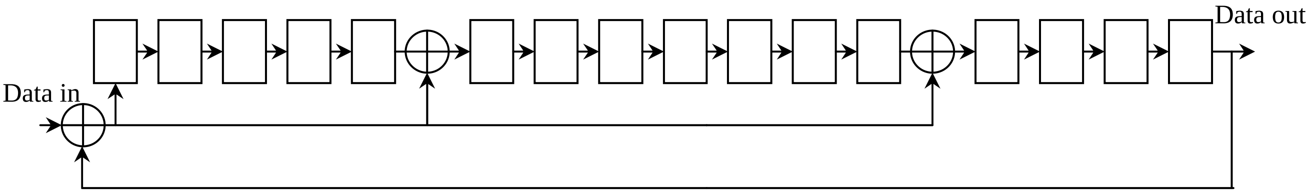


Figure 3.14.1 CRC Encoding/Decoding Circuit

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.

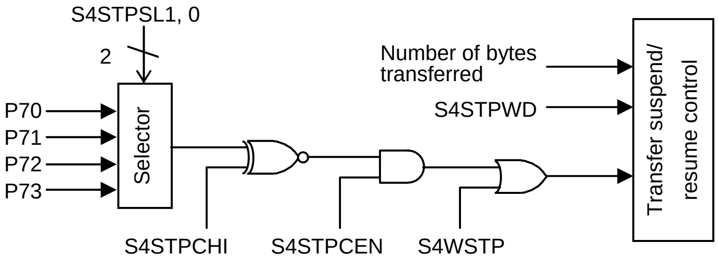
0: Transfer is suspended when the port (P70 to P73) is set to the low level and resumed when the port is set to the high level.

#### 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): Starting RAM address select**

1: On output, the RAM address 00 is selected first, followed by RAM address 01, and so on.
On input, the RAM address 01 is selected first, followed by RAM address 02, and so on.

0: On onput, 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<br>External | 0 | *1<br>— | 1 (*2) | 1<br>0 | — | 0 | *1 | 1 |
| P23 data transmission | None | Internal<br>External | 0 | *1<br>— | 0 (*2) | 1<br>0 | *1 | 1 | — | 0 |
| P23＆P22 data transmission | None | Internal<br>External | 0 | *1<br>— | 0/1 (*2) | 1<br>0 | *1 | 1 | *1 | 1 |
| None | P22 data reception | Internal<br>External | 0 | *1<br>— | 1 | 1<br>0 | — | 0 | — | 0 |
| None | P23 data reception | Internal<br>External | 0 | *1<br>— | 0 | 1<br>0 | — | 0 | — | 0 |
| P22 data transmission | P23 data reception | Internal<br>External | 0 | *1<br>— | 0 | 1<br>0 | — | 0 | *1 | 1 |
| P23 data transmission | P22 data reception | Internal<br>External | 0 | *1<br>— | 1 | 1<br>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 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 starting 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 starting 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<br>[H] | S4ADRL<br>[H] | RAM Start Address<br>[H] |
|---|---|---|
| 00 | 00 | 0000 |
| 00 | 01 | 0001 |
| ≀ | ≀ | ≀ |
| 03 | FF | 03FF |

### 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 (bit 4): Reserved bit

This bit must always be set to 0.

### S4BYTH3 to S4BYTH0 (bits 3 to 0): High-order 4 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.

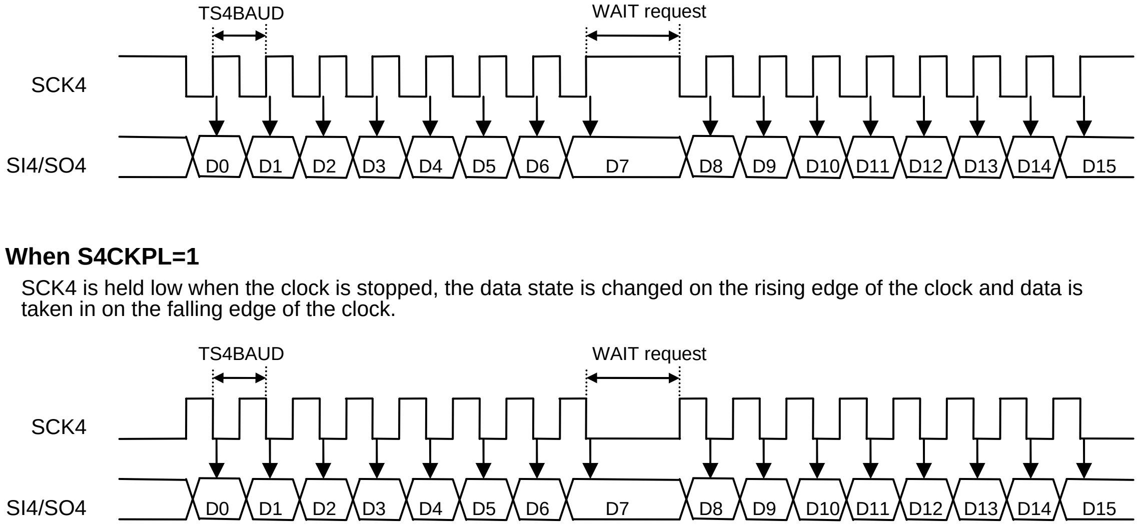
| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEE0 | 0000 0000 | R/W | S4BYTE | S4BYT7 | S4BYT6 | S4BYT5 | S4BYT4 | S4BYT3 | S4BYT2 | S4BYT1 | S4BYT0 |

The low-order 4 bits of S4BYTH and 8 bits of S4BYTE are used to define the number of data bytes to be transferred.

| S4BYTH Low-order 4 Bits [H] | S4BYTE [H] | Transfer Data Byte Count |
|---|---|---|
| 0 | 00 | 1 |
| 0 | 01 | 2 |
| ∫ | ∫ | ∫ |
| 3 | FF | 1024 |

**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

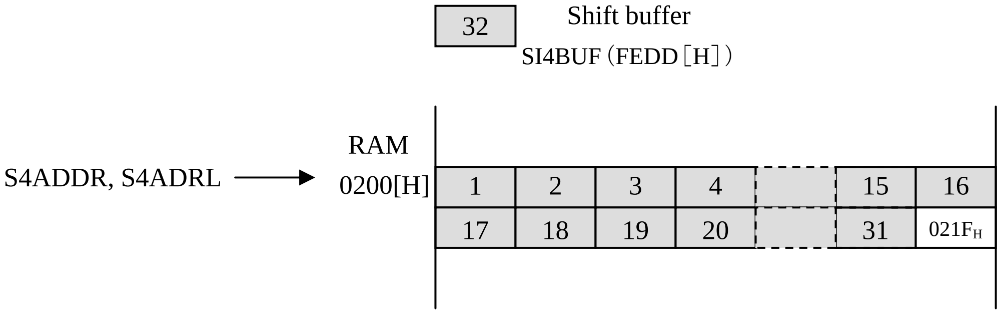
5) Setting the byte count [S4BYTH, S4BYTE]
   - Specify the number of bytes to be received continuously.

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

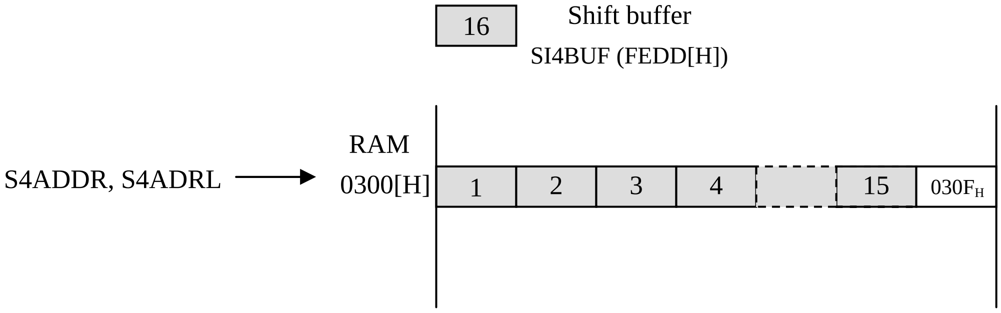
- Received data is stored in RAM sequentially starting at the specified RAM starting address. The last data byte is held in the shift buffer and not transferred to RAM.
- For example, when the byte data stream

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]

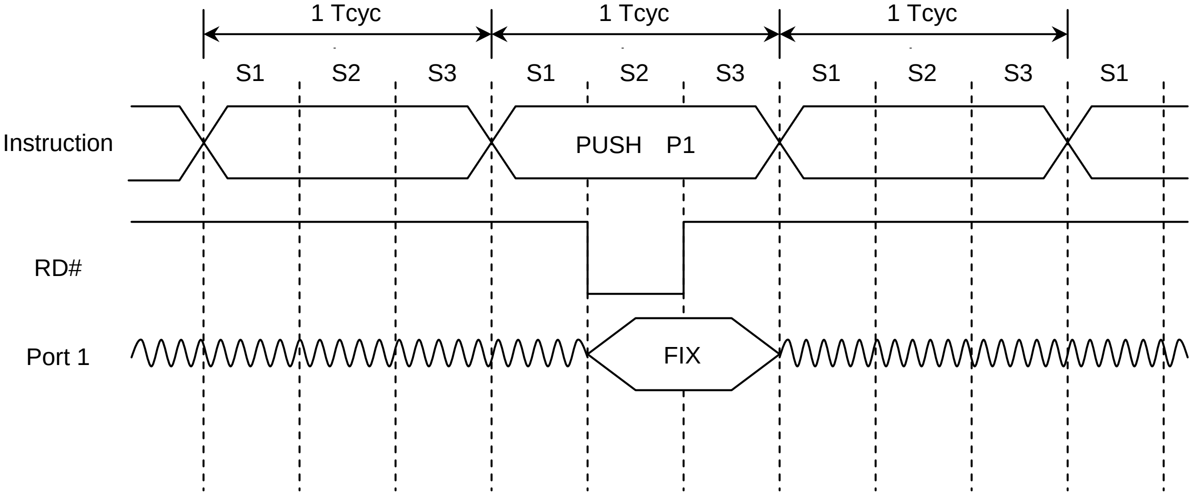
- P1/P0=1 (port 1）

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.

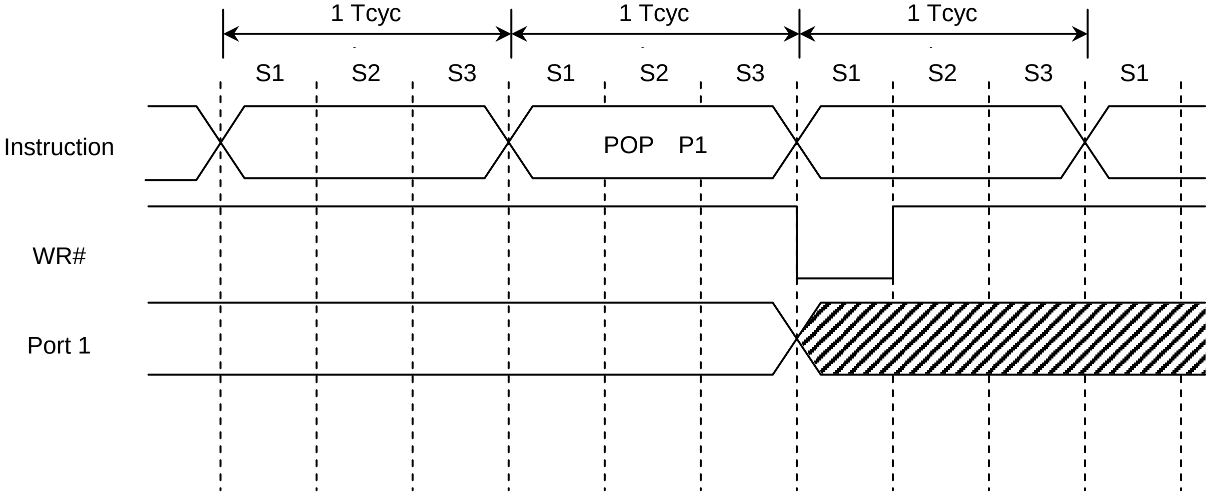
<Read mode timing chart>



<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 Asynchronous Serial Interface 1 (UART1)

## 3.16.1 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
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.

## 3.16.2 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 3.16.4) (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
- P2, P2DDR

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FED0 | 0000 0000 | R/W | UCON0 | UBRSEL | STRDET | RECRUN | STPERR | U0B3 | RBIT8 | RECEND | 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.*

## 3.16.3 Circuit Configuration

### 3.16.3.1 UART1 control register 0 (UCON0) (8-bit register)

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

### 3.16.3.2 UART1 control register 1 (UCON1) (8-bit register)

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

### 3.16.3.3 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).

### 3.16.3.4 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.

### 3.16.3.5 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).

### 3.16.3.6 UART1 receive data register (RBUF) (8-bit register)

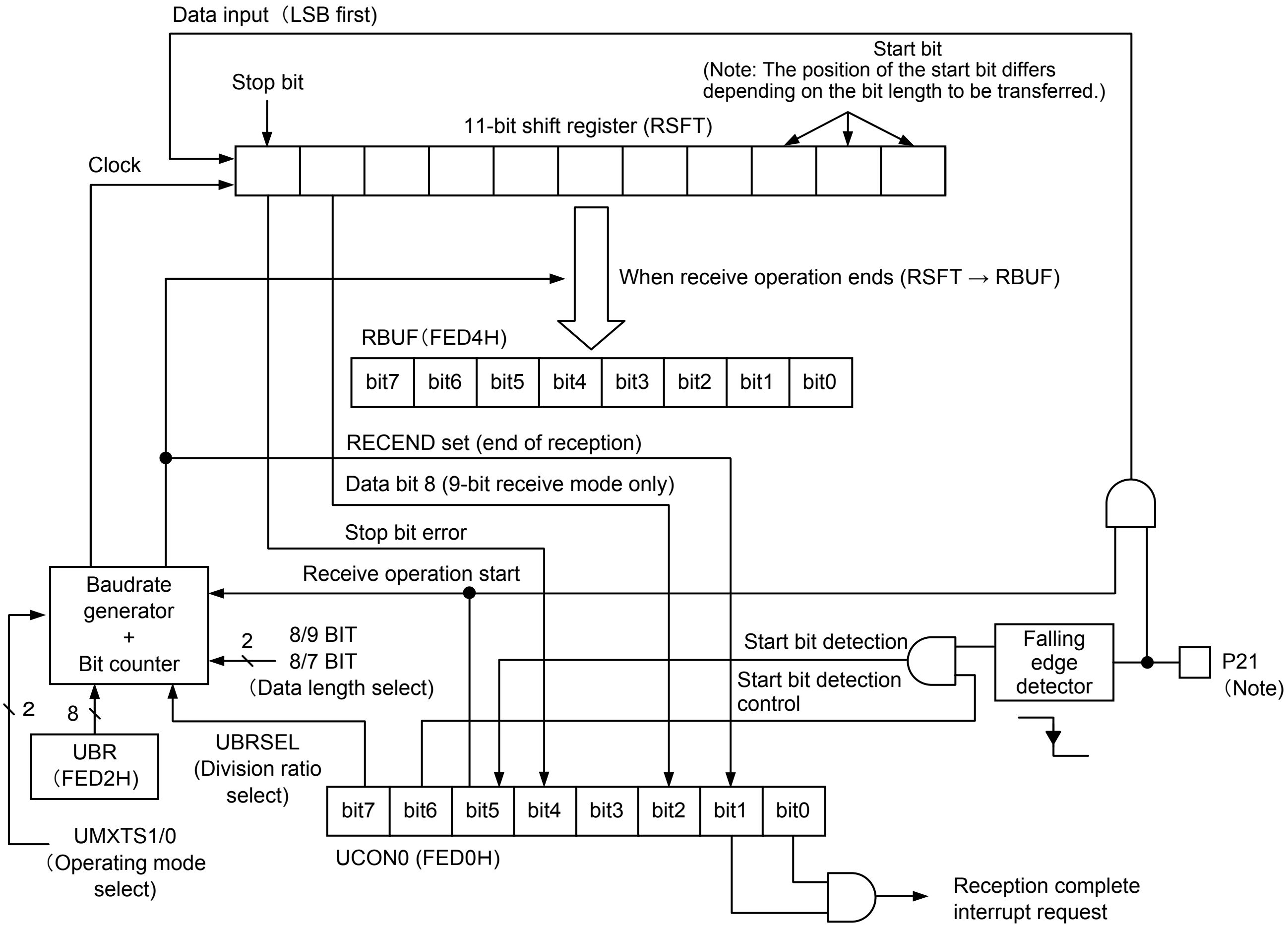
1) This register is an 8-bit register for storing the UART1 receive data.

### 3.16.3.7 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).

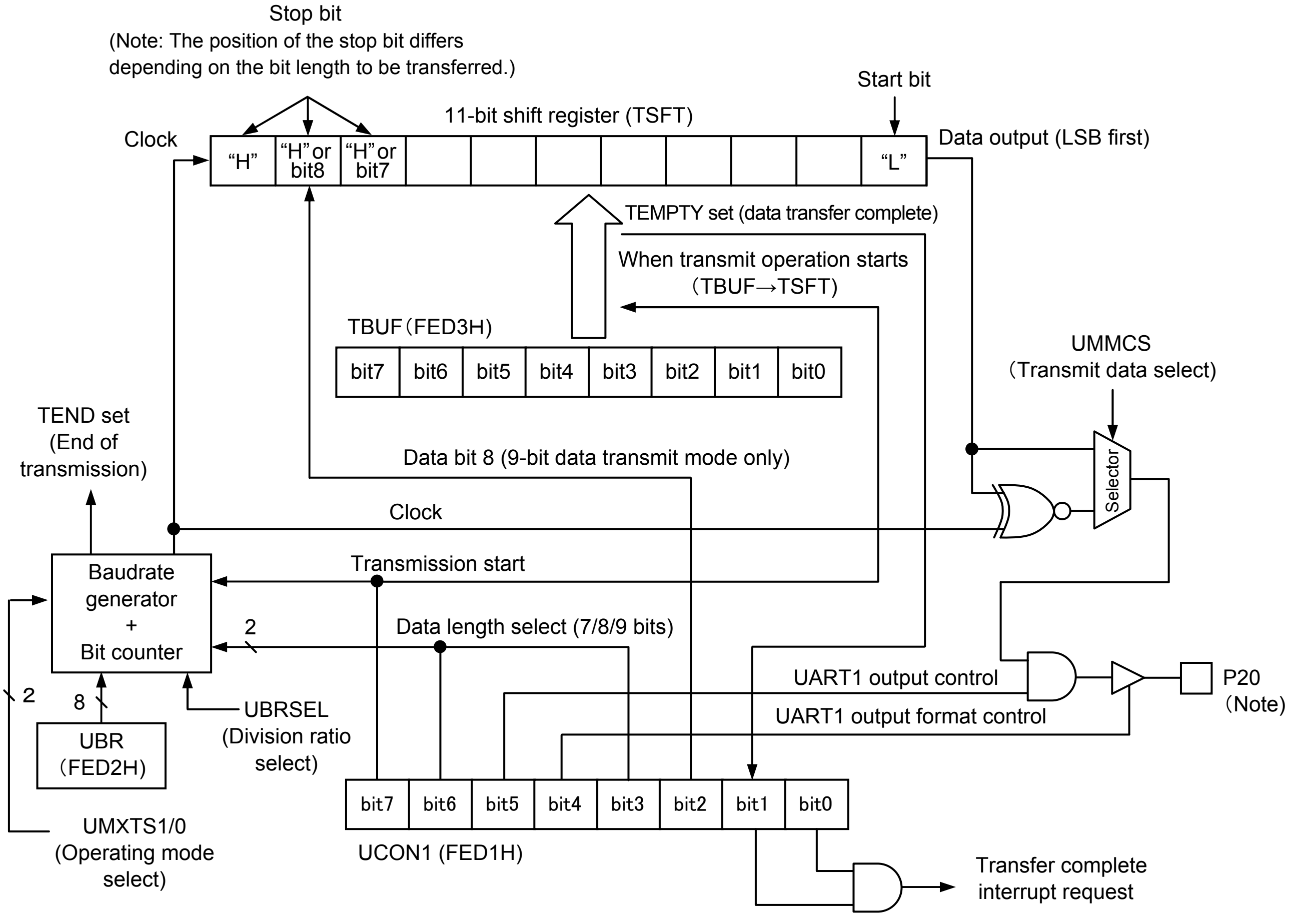
### 3.16.3.8 UART1 mode select register (UMDSL) (8-bit register)

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 P2DDR (at FE49H) 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 3.16.1 UART1 Block Diagram (Reception Mode)**



Stop bit
(Note: The position of the stop bit differs depending on the bit length to be transferred.)
Start bit
11-bit shift register (TSFT)
Clock
"H"
"H" or bit8
"H" or bit7
"L"
Data output (LSB first)
TEMPTY set (data transfer complete)
When transmit operation starts (TBUF→TSFT)
TBUF (FED3H)
bit7
bit6
bit5
bit4
bit3
bit2
bit1
bit0
UMMCS (Transmit data select)
Selector
TEND set (End of transmission)
Data bit 8 (9-bit data transmit mode only)
Clock
Transmission start
Baudrate generator + Bit counter
2
Data length select (7/8/9 bits)
2
8
UBR (FED2H)
UBRSEL (Division ratio select)
UART1 output control
UART1 output format control
P20 (Note)
UMXTS1/0 (Operating mode select)
UCON1 (FED1H)
Transfer complete interrupt request

*Note: Bit 0 of P2DDR (at FE49H) 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 3.16.2 UART1 Block Diagram (Transmission mode)**

## 3.16.4 Related Registers

### 3.16.4.1 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 (P21) is detected when the start bit detection function is enabled (STRDET=1).

<2> This bit is automatically cleared at the end of the receive operation (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

<1> This bit can be used as a general-purpose flag.

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

### RBIT8 (bit 2): UART1 receive data bit 8 storage bit

<1> When the data length is 9-bit long (UCON1:8/9BIT=1), this bit is loaded with bit 8 of the receive data at the end of receive operation.

### 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.

### 3.16.4.2 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.

* 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 | − | 9 bits |

### 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 (P20). (No transmit data is generated if bit 0 of P2DDR (FE49H) is set to 1.)

<2> When this bit is set to 0, no transmit data is placed at the transmit port (P20).

* 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 (P20) is set to CMOS.

<2> When this bit is set to 0, the output type of the transmit port (P20) is set to N-channel open drain.

### 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.

### 3.16.4.3 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 |

### 3.16.4.4 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 |

### 3.16.4.5 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 |

### 3.16.4.6 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.

## 3.16.5 UART1 Continuous Communication Operation Examples

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

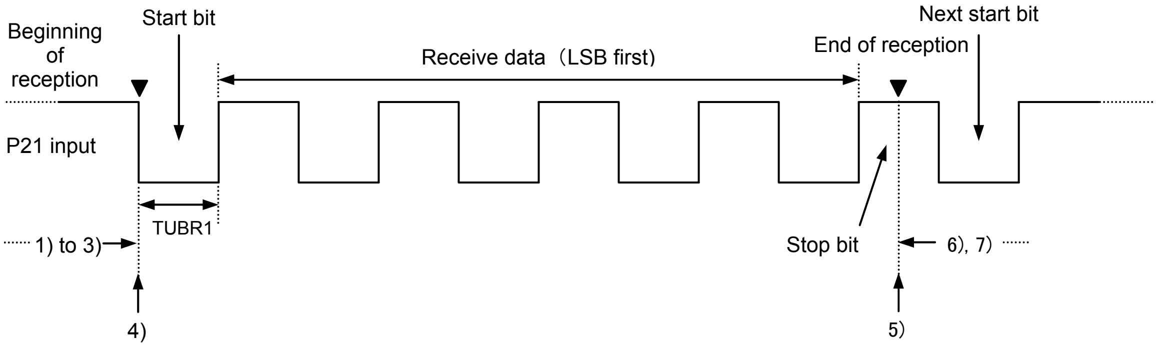


**Figure 3.16.3 Example of Continuous 8-bit Data Reception Mode Operation (Programmable Transfer Mode)**

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 P2DDR:P21DDR to 0 and P2:P21 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 (P21) 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.*

### 3.16.5.2 Continuous 8-bit data transmission mode (transmit data = 55H)

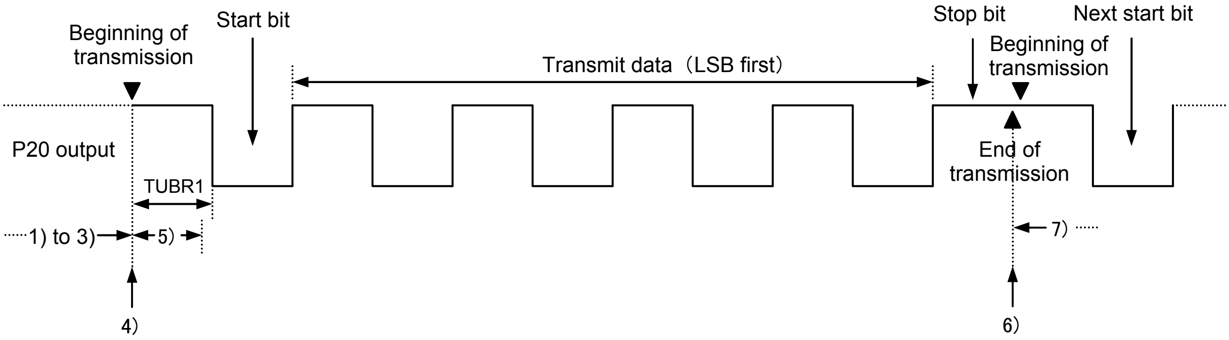


**Figure 3.16.4 Example of Continuous 8-bit Data Transmission Mode Operation (Programmable Transfer Mode)**

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 P2DDR:P20DDR to 0 and P2:P20 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.*

## 3.16.6 Supplementary Notes on UART1

### 3.16.6.1 About transmit data conversion

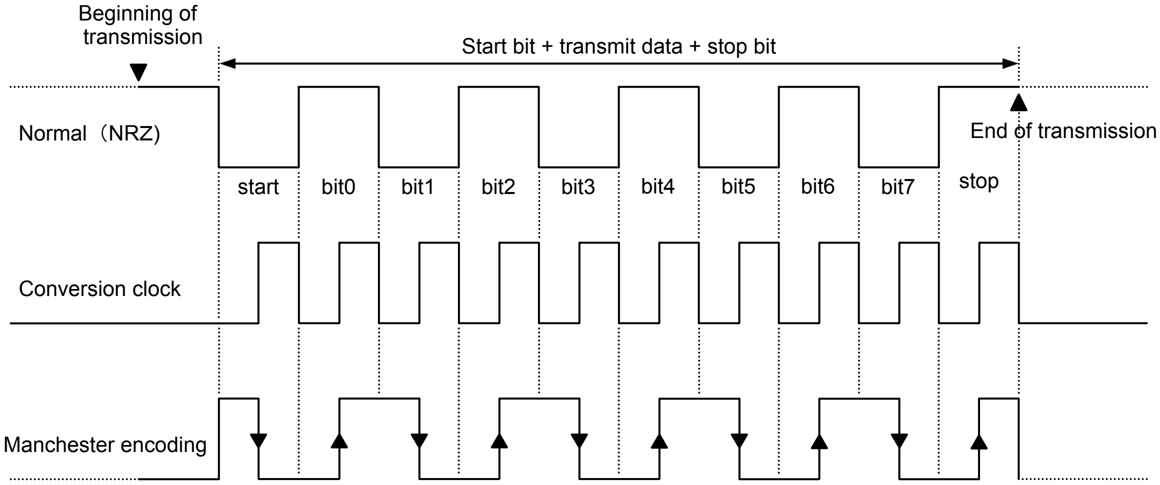


**Figure 3.16.5 Example of Transmit Data Conversion (Transmit Data = 55H)**

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%.

### 3.16.6.2 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.

### 3.16.6.3 UART1 communication port settings

1) Receive port (P21) settings

| Register Data | | Receive Port (P21) State | Internal Pull-up Resistor |
|---|---|---|---|
| P21 | P21DDR | | |
| 0 | 0 | Input | OFF |
| 1 | 0 | Input | ON |

* UART1 cannot receive data normally if P21DDR is set to 1.

2) Transmit port (P20) settings

| Register Data | | | | Transmit Port (P20) State | Internal Pull-up Resistor |
|---|---|---|---|---|---|
| P20 | P20DDR | TDDR | TCMOS | | |
| 0 | 0 | 1 | 1 | CMOS output | OFF |
| 0 | 0 | 1 | 0 | N-channel open drain output | OFF |
| 1 | 0 | 1 | 0 | N-channel open drain output | ON |

* UART1 transmits no data if P20DDR is set to 1.

## 3.16.7 UART1 HALT Mode Operation

### 3.16.7.1 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.

### 3.16.7.2 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.

# 3.17 Smart Card Interface (SCUART)

## 3.17.1 Overview

This series of microcontrollers incorporates an asynchronous serial interface circuit that has the following functions and features:

1) Data bit length: 7 or 8 bits
2) First bit: LSB/MSB first
3) Stop bit length: 1 or 2 bits
4) Parity bit: None/even/odd
5) Appends a parity bit on transmission and checks parity on reception
6) Binary inversion on transmission and reception
7) Checks for receive framing errors (stop bits)
8) Detects receive overrun errors
9) Full duplex communication
10) Capable of setting arbitrary bit rates
11) Smart card interface
    - Detects an error signal (parity error) and retransmit (repeat) operation when transmitting
    - Sends an error signal (parity error) and re-receive (repeat) operation when receiving
    - Capable of selecting direct convention/inverse convention
    - Capable of selecting the number of output clocks per 1 etu

    * etu: Elementary Time Unit (time to transmit or receive 1 bit)

    *The smart card interface (SCUART) supports an interface with IC cards (smart cards) that conform to ISO/IEC7816-3.

## 3.17.2 Functions

1) Data transmission and reception
    - Performs continuous transmission/reception or both transmission and reception at the same time on a frame basis in the preset format (limited to the format that is established before communication is initiated).
    - A transmission begins with a start bit (low level output) and ends with a stop bit (high level output or pulled up to a high level).
    - A reception begins with synchronization of the receiver circuit which is triggered on the detection of a start bit (low level) and ends with the detection of a stop bit (high level).
    - On transmission, the data stored in the transmit data register (SCTBUF) is transferred to the transmit shift register, after which it is transmitted out.
    - On reception, the received data is placed in the receive shift register. The reception terminates when the required number of data bits are received, after which the received data is transferred to the receive data register (SCRBUF).

2) Interrupt generation

    Interrupts are generated at the end of a data transfer from the transmit data register (SCTBUF) to the transmit shift register (transmit data register empty), at the end of transmission, and at the end of reception.

    These interrupt requests occur when the corresponding interrupt request enable bits are set.

3) It is necessary to manipulate the following special function registers to control this circuit.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FECA | 0000 0010 | R/W | SCCNT0 | SCTDDR | SCTSTP | SCTRUN | SCTERR | SCTEND | SCTENDIE | SCTEMPTY | SCTEMPTYIE |
| FECB | 0000 0000 | R/W | SCCNT1 | SCERCE | SCCHRL | SCRRUN | SCROVER | SCRFERR | SCRPERR | SCREND | SCRENDIE |
| FECC | H000 0000 | R/W | SCMOD | - | SCSCK | SCDIV1 | SCDIV0 | SCPODD | SCPEN | SCDIR | SCINV |
| FECD | 0000 0000 | R/W | SCTBUF | SCTBUF7 | SCTBUF6 | SCTBUF5 | SCTBUF4 | SCTBUF3 | SCTBUF2 | SCTBUF1 | SCTBUF0 |
| FECE | 0000 0000 | R | SCRBUF | SCRBUF7 | SCRBUF6 | SCRBUF5 | SCRBUF4 | SCRBUF3 | SCRBUF2 | SCRBUF1 | SCRBUF0 |
| FECF | 0000 0000 | R/W | SCBRG | SCBRG7 | SCBRG6 | SCBRG5 | SCBRG4 | SCBRG3 | SCBRG2 | SCBRG1 | SCBRG0 |

## 3.17.3 Circuit Configuration

The block diagram of the smart card interface (SCUART) is shown in Figure3.17.1

The smart card interface (SCUART) block is shown inside the dotted lines in the figure.

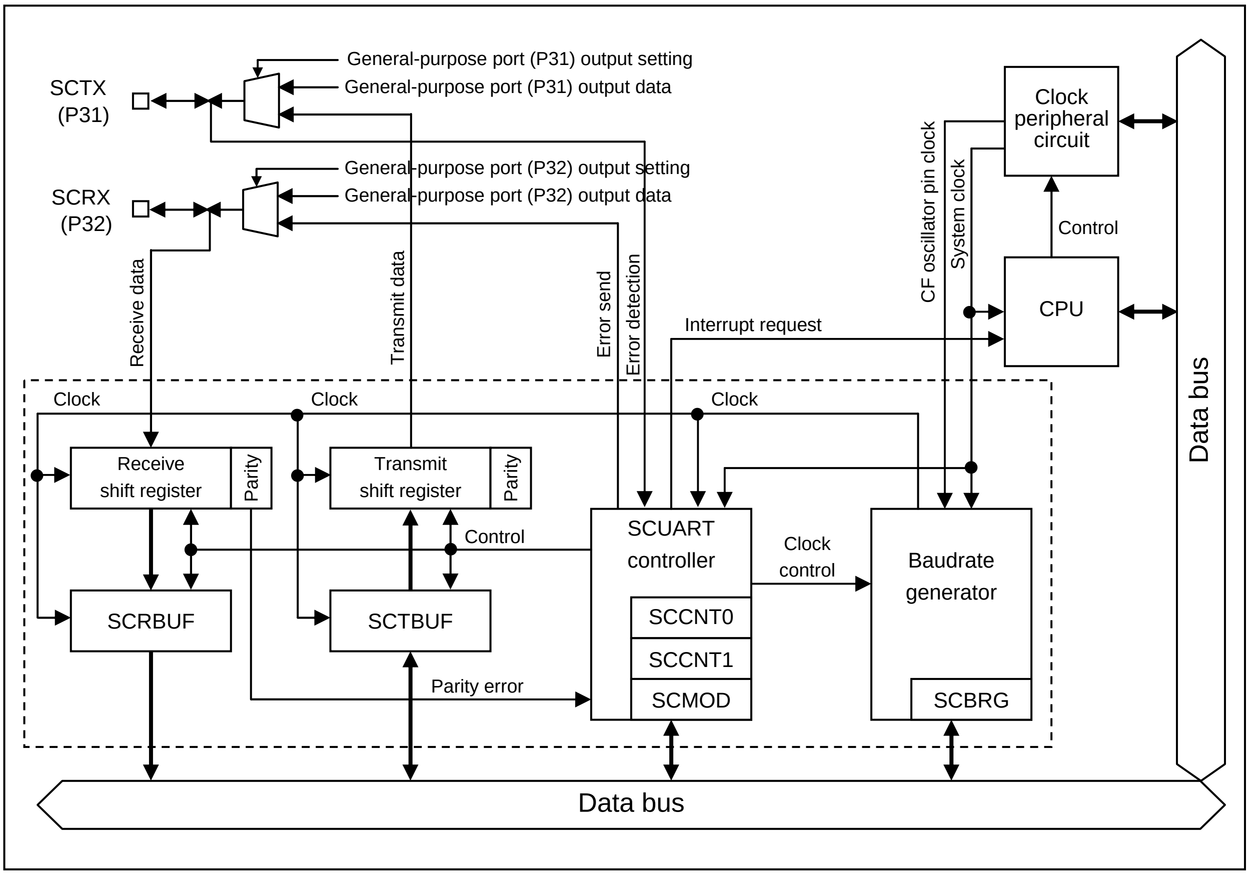


**Figure 3.17.1 Smart Card Interface (SCUART) Block Diagram**

## 3.17.4 Input/Output Pins

The table 3.17.1 shows the input/output pins that are used for full duplex UART communication.

See Subsection 3.17.8.3 for the input/output pins to be used for communication with a smart card.

**Table 3.17.1 Input/Output Pins Used for Full Duplex UART Communication**

| Pin Name | I/O | Function/Setting |
|---|---|---|
| SCTX<br>(P31) | Output | Pin used to transmit data.<br>SCTDDR (FECAH, bit 7) must be set to 1.<br>P31 (FE4CH, bit 1) must be set to 0.<br>* The internal pull-up resistor can be used when P31 (FE4CH, bit 1) is set to 1.<br>P31DDR (FE4DH, bit 1) must be set to 0.<br>The P31 port option must be set to CMOS.<br>*High-level output cannot be generated if the port option is set to N-channel open drain. In this case, the pin is kept open (high impedance) or pulled up to a high level by the internal pull-up resistor. |
| SCRX<br>(P32) | Input | Pin used to receive data.<br>P32 (FE4CH, bit 2) must be set to 0.<br>* The internal pull-up can be used when P32 (FE4CH, bit 2) is set to 1.<br>P32DDR (FE4DH, bit 2) must be set to 0.<br>The P32 port option may be set to either value. |

## 3.17.5 Related Registers

### 3.17.5.1 SCUART control register 0 (SCCNT0)

1) This register is an 8-bit register that controls the transmit/receive operations of the SCUART.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FECA | 0000 0010 | R/W | SCCNT0 | SCTDDR | SCTSTP | SCTRUN | SCTERR | SCTEND | SCTENDIE | SCTEMPTY | SCTEMPTYIE |

**SCTDDR (bit 7): Transmit port (SCTX) control**

0: The bit functions as a general-purpose port (P31).

1: The bit functions as a transmit port (SCTX).

* When setting this bit to 1, also set P31DDR (FE4DH, bit 1) to 0.

**SCTSTP (bit 6):**

- When SCERCE=0, this bit specifies the number of stop bits in transmit mode.

  0: 1 bit

  1: 2 bits

  * See also Figure 3.17.8 for instructions on specifying the number of stop bits.

- When SCERCE=1, this bit specifies the wait time for retransmission to be referenced when an error signal is detected in transmit mode.

  0: 2 etu

  1: 3 etu

### SCTRUN (bit 5): Transmit function master control

0: Stops the function

1: Starts the function

Setting this bit to 1 starts the transmit function and causes the SCUART to wait until the transmission start conditions are established.

Setting this bit to 0 initializes the transmit function.

* This bit must be set to 1 after defining the transmit/receive format. When changing the setting of the transmit/receive format, this bit must be set to 0.

* The data from the transmit data register (SCTBUF) is transmitted if this bit is set when this bit is set to 0, SCTEMPTY is set to 0, and SCTEND is set to 0. This bit serves as the condition for initiating a transmit operation. See paragraph 1) of 3.17.6.2 for details.

### SCTERR (bit 4): Transmit-time error signal detection flag

This bit is set if an error signal is detected during transmission performed with SCERCE set to 1. This bit is not set, however, provided that the number of error signals detected is within the preset repeat count.

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

* This bit is set at the same time SCTEND is set at the end of transmission.

### SCTEND (bit 3): Transmit end flag

This bit is set when a transmission terminates with SCTEMPTY set to 1.

When the transmission terminates with SCTEMPTY set to 0, the bit is not set and transmission of the next data is started. For details, see paragraph 4) of 3.17.6.2.

If the transmission terminates with error signal detection when SCERCE is set to 1, however, the bit is not set and retransmission is carried out regardless of the state of SCTEMPTY provided that the number of errors detected is within the preset repeat count. If the number of errors detected exceeds the repeat count preset, the bit is set and transmit processing is terminated.

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

* No transmission is carried out as long as this bit is set to 1.

* The data from the transmit data register (SCTBUF) is transmitted if this bit is cleared when this bit is set to 1, SCTEMPTY is set to 0, and SCTRUN is set to 1. This bit serves as the condition for initiating a transmit operation. See paragraph 1) of 3.17.6.2 for details.

### SCTENDIE (bit 2): Transmit end interrupt request enable

0: Disables transmit end interrupt requests

1: Enables transmit end interrupt requests

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

### SCTEMPTY (bit 1): Transmit data register empty flag

This bit is set when the data transfer from the transmit data register (SCTBUF) to the transmit shift register terminates.

The bit is cleared when data is written into the transmit data register (SCTBUF).

* This bit is read-only.

* The initial value of this bit is 1.

* This bit serves as the condition for initiating a transmit operation. See paragraph 1) of 3.17.6.2 for details.

**SCTEMPTYIE (bit 0): Transmit data register empty interrupt request enable**

0: Disables transmit data register empty interrupt requests

1: Enables transmit data register empty interrupt requests

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

### 3.17.5.2 SCUART control register 1 (SCCNT1)

1) This register is an 8-bit register that controls the SCUART transmit/receive operations.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FECB | 0000 0000 | R/W | SCCNT1 | SCERCE | SCCHRL | SCRRUN | SCROVER | SCRFERR | SCRPERR | SCREND | SCRENDIE |

**SCERCE (bit 7):**

This bit controls detection and transmission of error signals and repeat (retransmit/re-receive) operations, and modifies some SCUART special function register functions.

0: Disables the function (used for full duplex UART communication)

1: Enables the function (used for communication with a smart card)

* See 3.17.8 for the uses of this bit when set to 1.

**SCCHRL (bit 6):**

- When SCERCE is set to 0, this bit sets the bit length of the transmit/receive data.

  0: 8 bits

  1: 7 bits

  * When this bit is set to 1, bit 6 of the transmit data register (SCTBUF) and receive data register (SCRBUF) are handled as MSB. In this case, bit 7 of the receive data register (SCRBUF) is set to 0 at the end of a receive operation.

  * This setting is common to both transmit and receive formats.

- When SCERCE is set to 1, this bit sets the repeat count.

  The bit defines the number of times transmit retries are to be repeated on detection of an error signal, and the number of times retries to transmit an error signal and retries to re-receive data are to be repeated on detection of a receive parity error. This bit is used to set the repeat count together with SCPEN as shown in Table 3.17.2.

**Table 3.17.2 Repeat Count Settings**

| SCPEN | SCCHRL | Repeat Count |
|---|---|---|
| 0 | X | 0 time |
| 1 | 0 | 3 times |
| 1 | 1 | 4 times |

* This setting is common to the repeat count for data transmissions and the repeat count for error signal transmit and re-receive operations.

* The repeat operation is terminated when error signals are consecutively detected or receive parity errors occur consecutively the number of times equal to the set repeat count plus 1.

* The error count is reset when the processing terminates normally within the preset repeat count or the processing terminates in error after repeating more than the specified number of retries.

### SCRRUN (bit 5): Receive function master control

0: Stops the receive function

1: Starts the receive function

Setting this bit to 1 starts the receive function and causes the SCUART to wait until the initiation of a receive operation is detected (a low level start bit is detected).

Setting this bit to 0 initializes the receive function.

* This bit must be set to 1 after defining the transmit/receive format. When changing the transmit/receive format setting, this bit must be set to 0.

### SCROVER (bit 4): Receive overrun error flag

When the conditions for setting SCREND (end of reception) are established with SCREND set to 1, this bit is set and the receive data register (SCRBUF) is updated with the receive data.

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

* This bit is set at the same time SCREND is set at the end of reception.

### SCRFERR (bit 3): Receive framing error flag

- When SCERCE=0:

  This bit is set when a low level stop bit is detected in receive mode.

  Even when this bit is set, SCREND is set and the receive data register (SCRBUF) is updated with the receive data.

  * Even when SCTSTP is set to 1 and the number of stop bits is set to 2 bits in transmit mode, only the first stop bit is checked in receive mode.

- When SCERCE=1:

  The stop bit is not checked.

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

* This bit is set at the same time SCREND is set at the end of reception.

* Stop bit check: The stop bit is sampled and its high or low level is checked.

### SCRPERR (bit 2): Receive parity error flag

This bit is set when an error is detected during the receive-time parity error check performed with SCPEN set to 1 or SCERCE set to 1. If SCERCE is set to 1, however, this bit is not set, provided that the number of receive parity errors detected is within the preset repeat count.

When SCERCE is set to 1, an error signal is transmitted every time a receive parity error occurs. (An error signal is transmitted even when this bit is set after the number of receive parity errors exceeds the preset repeat count.)

Even when this bit is set, SCREND is set and the receive data register (SCRBUF) is updated with the receive data.

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

* This bit is set at the same time SCREND is set at the end of reception.

* Parity bit check: The check is conducted based on the parity that is generated from the received data and the parity bit that is received.

**SCREND (bit 1): Receive end flag**

This bit is set at the end of a receive operation. The receive data is then transferred from the receive shift register to the receive data register (SCRBUF) and the receive data register (SCRBUF) is updated with the receive data.

This bit is also set when a receive operation ends on conditions under which SCROVER, SCRFERR, or SCRPERR is set. In this case, the receive data register (SCRBUF) is updated with the receive data.

If SCERCE is set to 1, however, this bit is not set even when a receive operation terminates on a receive parity error, provided that the number of receive parity errors detected is within the preset repeat count. In that case, the SCUART waits to receive the data again. The receive data register (SCRBUF) is not updated with the receive data.

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

**SCRENDIE (bit 0): Receive end interrupt request enable**

0: Disables receive end interrupt requests

1: Enables receive end interrupt requests

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

### 3.17.5.3 SCUART mode control register (SCMOD)

1) This register is a 7-bit register that defines the transmit/receive format of the SCUART.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FECC | H000 0000 | R/W | SCMOD | - | SCSCK | SCDIV1 | SCDIV0 | SCPODD | SCPEN | SCDIR | SCINV |

**(Bit 7): This bit does not exist.**

This bit cannot be written. It is always read as 1.

**SCSCK (bit 6): SCUART clock source setting**

0: System clock

1: CF oscillator pin clock

See the section on the “System Clock Generator Function” for details on the system clock.

The frequency of the “CF oscillator pin clock” is identical to that of the CF oscillator pin.

* To use the SCUART with this bit set to 1, set CFSTOP (FE0EH, bit 0) to 0 to activate the CF (main clock) oscillator circuit.

**SCDIV1 (bit 5): Frequency division ratio setting for the SCUART clock**

**SCDIV0 (bit 4): Frequency division ratio setting for the SCUART clock**

Table 3.17.3 lists the frequency division ratio settings for the SCUART clock.

These bits define the frequency division ratio for the clock that is selected by SCSCK. The SCUART runs at a clock frequency that is determined by this setting.

**Table 3.17.3 SCUART Clock Frequency Division Ratio Settings**

| SCDIV1 | SCDIV0 | Frequency Division Ratio |
|---|---|---|
| 0 | 0 | 1/1 |
| 0 | 1 | 1/2 |
| 1 | 0 | 1/4 |
| 1 | 1 | 1/8 |

### SCPODD (bit 3): Parity setting

0: Even parity

1: Odd parity

This bit sets the parity to be used for transmission and reception with SCPEN set to 1 or SCERCE set to 1.

In even parity mode, the total number of “1” bits in the data bits (7 or 8 bits) and the parity bit becomes even.

In odd parity mode, the total number of “1” bits in the data bits (7 or 8 bits) and the parity bit becomes odd.

* When SCERCE=0 and SCCHRL=1 (data bit length: 7 bits), parity is generated and checked on 7 data bits, i.e., bits 0 to 6.

* This setting is common to both transmit and receive formats.

### SCPEN (bit 2):

- When SCERCE=0: Parity control

  0: The parity scheme is not used for transmission and reception.

  1: A parity bit is appended on transmission and parity is checked on reception.

  * When using the SCUART with this bit set to 1, it is necessary to configure SCPODD.

  * This setting is common to both transmit and receive formats.

- When SCERCE=1: Repeat count setting

  This bit and SCCHRL are used to define the repeat count. See Table 3.17.2.

  *This setting is common to the retransmit repeat count, and the error signal transmit and re-receive repeat count.

### SCDIR (bit 1): Transmit/receive data first bit setting

0: LSB first

1: MSB first

For transmission, this bit defines the first bit with which data is to be transmitted from the transmit data register (SCTBUF). For reception, this bit defines the first bit with which the receive data is to be stored in the receive data register (SCRBUF).

* When SCERCE=0 and SCCHRL=1 (data bit length: 7 bits), bit 6 of the transmit data register (SCTBUF) and receive data register (SCRBUF) become the MSB.

* This setting is common to both transmit and receive formats.

### SCINV (bit 0): Binary inversion of the transmit/receive data bits and parity bit control

0: Disables inversion

1: Enables inversion

When this bit is set to 1, the data in the transmit data register (SCTBUF) is binary-inverted for transmission. The receive data is binary-inverted before being transferred to the receive data register (SCRBUF).

The parity bit is also binary-inverted if this bit is set to 1 in the configuration in which parity is to be used for transmission and reception. On transmission, the parity bit derived from the transmit data that is established before it is subjected to binary inversion is binary-inverted and appended to the transmit data as the parity bit. On reception, the parity bit is assumed to have been binary-inverted.

* To suppress the binary inversion of the parity bit when this bit is set to 1, invert the logic setting of SCPODD. This makes it possible to establish an equivalent configuration in which the parity bit is not binary-inverted during both transmission and reception.

* This setting is common to both transmit and receive formats.

### 3.17.5.4 SCUART transmit data register (SCTBUF)

1) This register is an 8-bit register for storing the SCUART transmit data.

2) Data from this register is transferred to the transmit shift register at the beginning of a transmit operation.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FECD | 0000 0000 | R/W | SCTBUF | SCTBUF7 | SCTBUF6 | SCTBUF5 | SCTBUF4 | SCTBUF3 | SCTBUF2 | SCTBUF1 | SCTBUF0 |

* Bit 6 of this register becomes the MSB when SCERCE=0 and SCCHRL=1 (data bit length: 7 bits).

* Since this register can be loaded with data regardless of the state of the SCTEMPTY, if an attempt is made to write data into this register with SCTEMPTY set to 0, the data that is yet to be transmitted (the old data that has been loaded before) will be overwritten.

* When writing data into this register while transmit operation is in progress, make sure that SCTEMPTY is set to 1.

### 3.17.5.5 SCUART receive data register (SCRBUF)

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

2) Data is transferred from the receive shift register to this register when SCREND is set to 1 at the end of a receive operation.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FECE | 0000 0000 | R | SCRBUF | SCRBUF7 | SCRBUF6 | SCRBUF5 | SCRBUF4 | SCRBUF3 | SCRBUF2 | SCRBUF1 | SCRBUF0 |

* When SCERCE=0 and SCCHRL=1 (data bit length: 7 bits), bit 6 of this register becomes the MSB, and bit 7 is set to 0 at the end of a receive operation.

* The receive data should be used after checking all of the SCROVER, SCRFERR, and SCRPERR error flags.

### 3.17.5.6 SCUART baudrate generator (SCBRG)

1) This register is an 8-bit register that defines the SCUART baudrate.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FECF | 0000 0000 | R/W | SCBRG | SCBRG7 | SCBRG6 | SCBRG5 | SCBRG4 | SCBRG3 | SCBRG2 | SCBRG1 | SCBRG0 |

The settings of this register, SCSCK, SCDIV1, and SCDIV0 determine the internal clock for the SCUART which is generated by the baudrate generator, and therefore determine the bit rate for transmission and reception and the transmit/receive time per bit (bit time).

The legitimate value range is from 01H to FFH.

See 3.17.6.3 for the procedure to determine the settings and examples and see also 3.17.8.4 when using the SCUART for communication with a smart card.

* Setting SCBRG to 00H is inhibited.

* Do not change the value of the baudrate while transmission or reception is being performed. When changing the baudrate, make sure that SCTRUN is set to 0 and SCRRUN to 0.

* The bit rate and bit time are the same for both transmission and reception.

* Set this register so that the difference in the bit rate and bit time between this SCUART and the opponent device is at a minimum.

* This setting is common to both transmit and receive formats.

## 3.17.6 Principles of Operation

This subsection describes the usage of the SCUART in full duplex UART communication mode.

It is assumed that SCERCE is set to 0.

See also 3.17.8 when using the SCUART for communication with a smart card.

### 3.17.6.1 Frame format

The SCUART performs data transmission and reception on a frame basis. The configuration of the frame is determined by the characteristics that are summarized in Table 3.17.4. These settings determine the format of the frame.

The frame configurations are shown in Table 3.17.5 and Figure 3.17.2

**Table 3.17.4 Frame Configuration and Format**

| Item | Setting |
|---|---|
| Start bits | - |
| Stop bit length | 1 or 2 bits |
| Data bit length | 7 or 8 bits |
| Parity bit | None/even/odd |
| First bit | LSB/MSB first |
| Data (including parity bit) binary inversion | Yes/No |

**Table 3.17.5 Frame Configurations**

| Configuration | Data Bit Length | Parity Bit | Stop Bit Length |
|---|---|---|---|
| <1> | 7 | No | 1 |
| <2> | 7 | Yes | 1 |
| <3> | 8 | No | 1 |
| <4> | 8 | Yes | 1 |
| <5> | 7 | No | 2 |
| <6> | 7 | Yes | 2 |
| <7> | 8 | No | 2 |
| <8> | 8 | Yes | 2 |

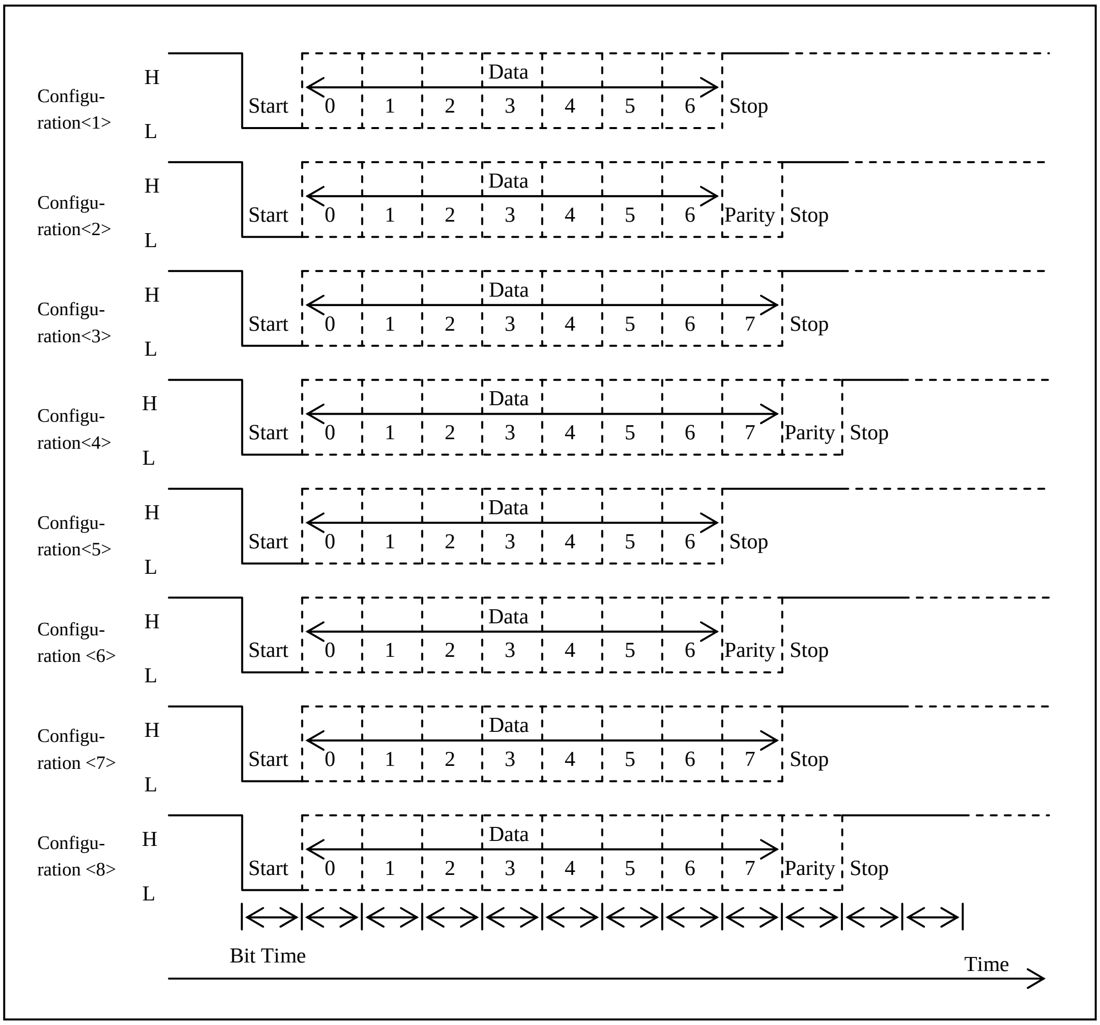


**Figure 3.17.2 Frame Configurations**

1) Parity bit

Sample transmit/receive waveforms generated with a parity bit are shown in Figure 3.17.3.

(Example) Even and odd parity bits in the configuration: data bit length = 8 bits, data = 9CH, stop bit length = 1 bit, LSB first, no binary inversion

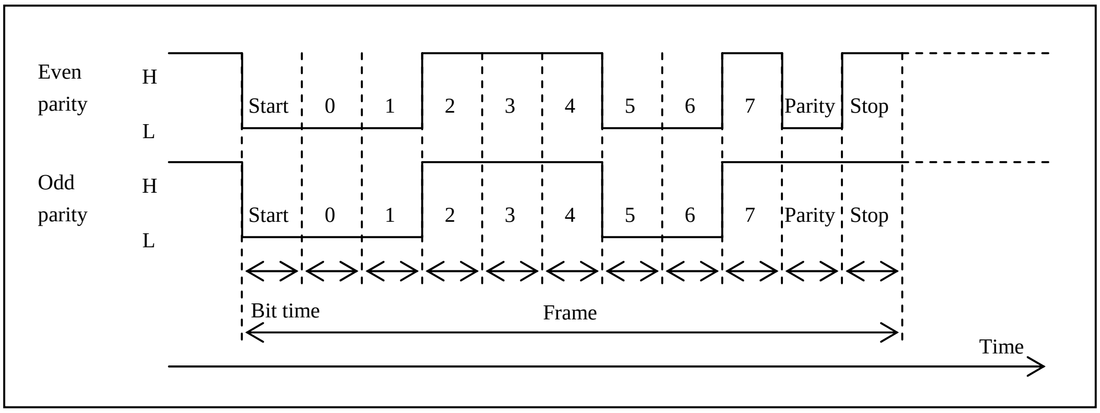


**Figure 3.17.3 Example of Transmit/Receive Waveforms (Parity Used)**

2) First bit

Sample transmit/receive waveforms with first bit settings are shown in Figure 3.17.4.

(Example) LSB first and MSB first modes in the configuration: data bit length = 8 bits,
data = 9CH, stop bit length = 1 bit, even parity, no binary inversion

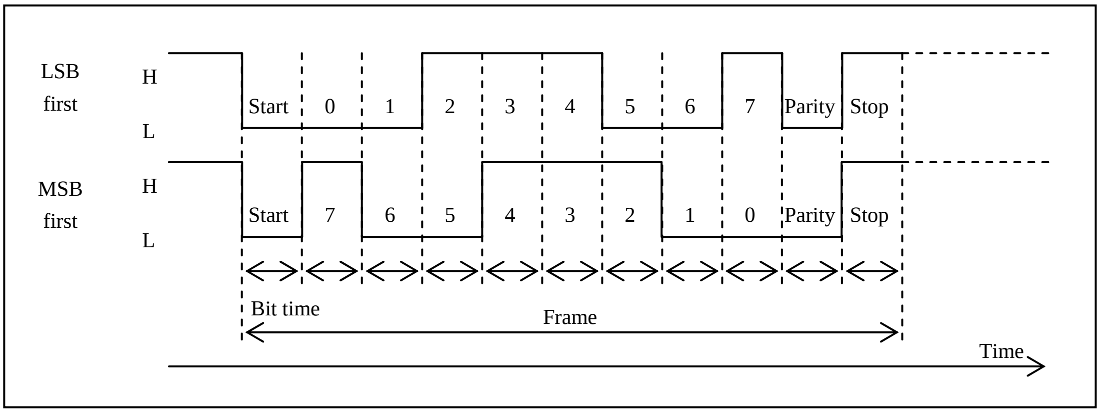


**Figure 3.17.4 Example of Transmit/Receive Waveforms (with First Bit Settings)**

3) Binary inversion

Sample transmit/receive waveforms generated with binary inversion control are shown in Figure 3.17.5.

(Example) Binary inversion enabled and disabled in the configuration: data bit length = 8 bits,
data = 9CH, stop bit length = 1 bit, LSB first, even parity

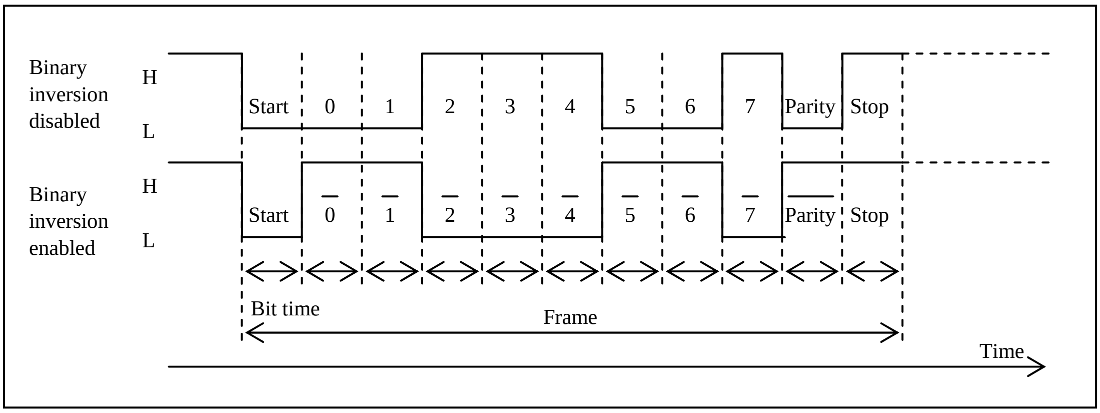


**Figure 3.17.5 Example of Transmit/Receive Waveforms (with Binary Inversion Control)**

### 3.17.6.2 Transmit/receive operations

Transmit and receive operations in the configuration: data bit length= 8 bits, data = 9CH, stop bit length = 1 bit, LSB first, even parity, no binary inversion are described below.

The transmit and receive start- and end-time timings are outlined in Figure 3.17.6.

The data transmit and receive timings are outlined in Figure 3.17.7.

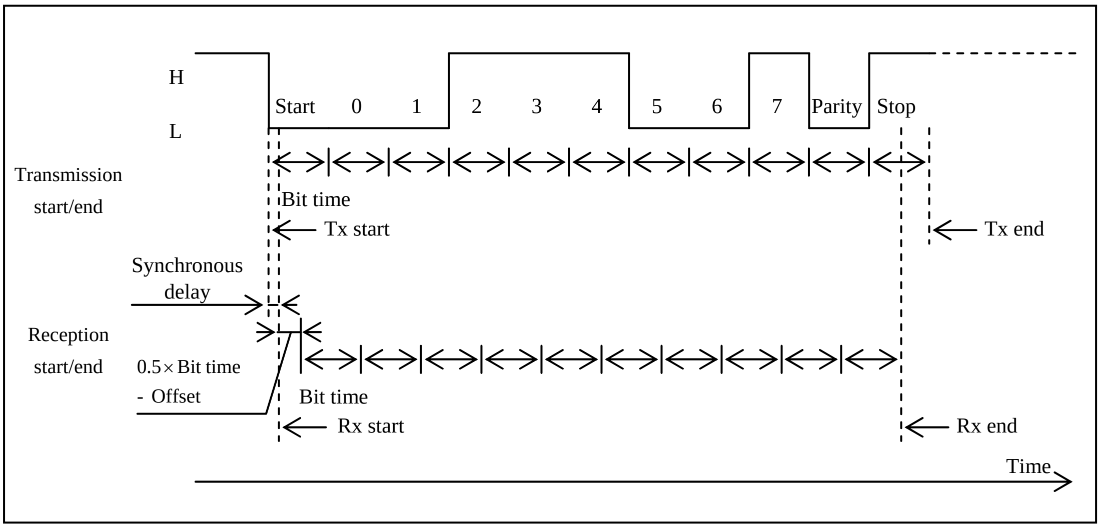


**Figure 3.17.6 Outline of Transmit and Receive Start and End Timings**

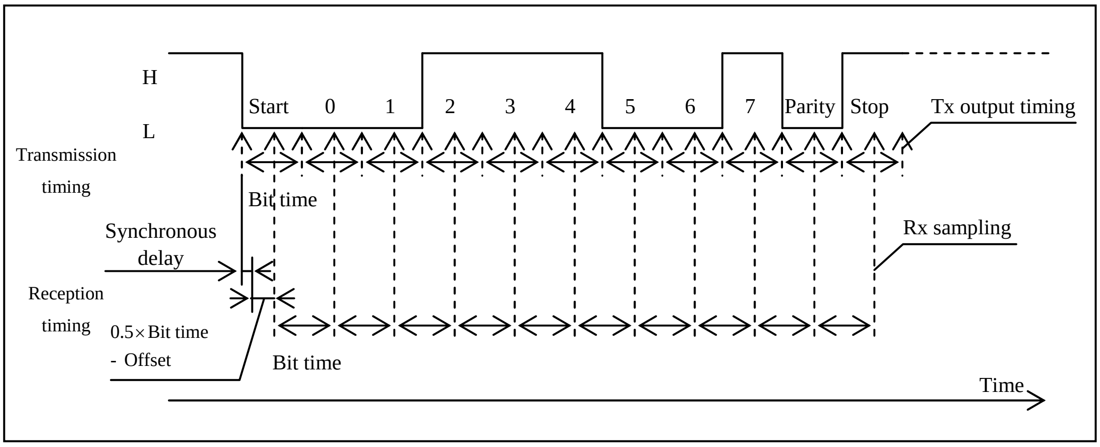


**Figure 3.17.7 Outline of Data Transmit and Receive Timings**

1) Transmission start to end

The SCUART does not initiate a transmit operation if SCTRUN is set to 0 or SCTEND is set to 1. Setting SCTEMTPY to 0 by writing transmit data into the transmit data register (SCTBUF) with SCTEMTPY set to 1 provides a trigger for starting a transmit operation.

To start a transmission, therefore, it is necessary to write the transmit data into the transmit data register (SCTBUF) when SCTRUN is set to 1, SCTEND set to 0, and SCTEMPTY set to 1. Then, SCTEMPTY goes to 0, enabling transmission. The data from the transmit data register (SCTBUF) is copied into the transmit shift register and SCTEMPTY is set to 1, after which transmission is started.

As soon as transmission is started, the SCUART enters the start bit state. Subsequently, it transmits data and parity bits, one bit during each bit-time interval. Finally, the SCUART enters the stop bit state. In one bit time, the SCUART terminates the transmit operation and sets SCTEND to 1.

2) Reception start to end

A receive operation is started when a start bit is detected with SCRRUN set to 1 and the receiver circuit is synchronized. Since the attempt to detect the start bit is performed at the sampling period that is determined by the settings of SCSCK, SCDIV1, and SCDIV0, a synchronization delay of at most this sampling period can occur.

The first receive action (sampling) is performed in “0.5 × bit time – offset” after the receive processing is started.

Subsequently, it receives data and parity bits, one bit during each bit-time interval.

Finally, the stop bit is detected and the receive operation is finished. At the same time, the received data is transferred from the receive shift register to the receive data register (SCRBUF) and SCREND is set to 1.

* For a receive operation to be started, the stop bit state (high level) needs to be established and maintained for a period of at least the bit time or longer before the frame to be received.

*Synchronization delay: $0 \leq \mathrm{delay} \leq \frac{1}{C \times S} [s]$ Expression 3.17.1

*Offset： $\mathrm{offset} = \frac{1}{C \times S} [s]$ Expression 3.17.2

C=SCSCK clock frequency [Hz]

S=SCDIV1, SCDIV0 frequency division value (1/1, 1/2, 1/4, 1/8)

3) Single frame transmission/reception when the stop bit length is set to 2 bits

The outline of a single frame transmission/reception with 2 stop bits is shown in Figure 3.17.8

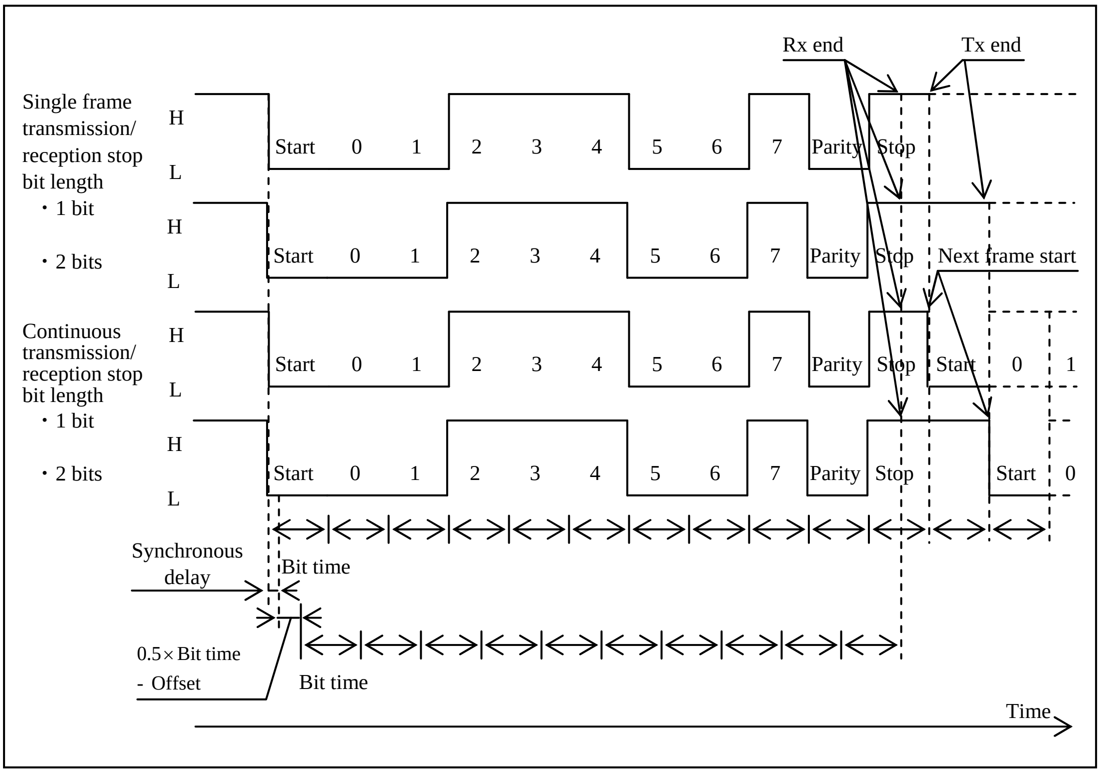


**Figure 3.17.8 Outline of Single and Continuous Transmission and Reception with 2-bit Stop Bit Length**

When the stop bit length is set to 2 bits for a single frame transmission, the transmission ends and SCTEND is set to 1 after a period of 2× bit time elapses in the stop bit state. For reception, the processing is the same as when the stop bit length is set to 1 bit. When the first stop bit is detected, the reception ends and SCREND is set to 1.

4) Continuous transmission

The outline of continuous transmission is shown in Figure 3.17.8.

To start a continuous transmission, check to make sure that SCTEMPTY is set to 1 during the period from when transmission is started until it is ended and SCTEND is set to 1 (while transmitting the preceding frame), then load the transmit data register (SCTBUF) with the next data. This process enables the SCUART to finish the transmission after transmitting the preceding frame and continue with the transmission of the next frame without causing SCTEND to be set to 1. In this case, the transmission of the next frame is started after the lapse of the bit time equivalent to the stop bit length set.

5) Continuous reception

The outline of continuous reception is shown in Figure 3.17.8.

The behavior of the SCUART in continuous receive processing is identical to that in single frame receive processing. The reception of the next frame is started after the reception of the preceding frame is ended and SCREND is set to 1. The SCUART saves the preceding receive data from the receive data register (SCRBUF) until the current receive processing is ended. Subsequently, the SCUART clears SCREND.

* For a receive operation to be started, the stop bit state (high level) needs to be established and maintained for a period of at least the bit time or longer before the frame to be received.

### 3.17.6.3 Transmit/receive bit rate

The transmit/receive bit rate and transmit/receive time per bit (bit time) are determined by the settings of SCSCK, SCDIV1, SCDIV0, and SCBRG.

The bit rate and transmit/receive time per bit (bit time) are calculated using the formulas shown in expressions 3.17.3 and 3.17.4.

$$B = \frac{C \times S}{4 \times (N + 1)} [bps] \qquad \text{Expression 3.17.3}$$

$$T = \frac{1}{B} = \frac{4 \times (N + 1)}{C \times S} [s] \qquad \text{Expression 3.17.4}$$

B = Bit rate [bps]

T = Bit time [s]

C = SCSCK clock frequency [Hz]

S = SCDIV1, SCDIV0 frequency division value (1/1, 1/2, 1/4, 1/8)

N = SCBRG value (1≦N≦255)

The value of SCBRG to be used as the desired bit rate can be calculated using the formula shown in expression 3.17.5.

$$N = \frac{C \times S}{4 \times B_M} - 1 \qquad \text{Expression 3.17.5}$$

$B_M$ = Desired bit rate [bps]

Calculate the value of N for each of the S values 1/1, 1/2, 1/4, and 1/8.

Then set the value of N so as to satisfy the inequality $1 \leqq N \leqq 255$.

Use the formula shown in expression 3.17.6 to identify the error in bit time at the possible bit rate for the bit time at the desired bit rate.

$$Err_T = \left( \frac{4 \times (N + 1) \times B_M}{C \times S} - 1 \right) \times 100 \ [\%] \qquad \text{Expression 3.17.6}$$

$Err_T$ = Error in possible bit time for the desired bit time ［%］

The error at the last bit of the frame can be obtained using the formula shown in expression 3.17.7 for transmission and the formula shown in expression 3.17.8 for reception.

For transmission: $Err_{Ttotal} = Err_{Tlength} [\%]$ Expression 3.17.7

For reception: $Err_{Ttotal} = Err_{Tlength} + Err_{Tdelay} - Err_{Toffset} [\%]$

Expression 3.17.8

$$Err_{Tlength} = Err_T \times L \ [\%] \qquad \text{Expression 3.17.9}$$

$$0 \leq Err_{Tdelay} \leq \frac{B_M}{C \times S} \times 100 \ [\%] \qquad \text{Expression 3.17.10}$$

$$Err_{Toffset} = \frac{B_M}{C \times S} \times 100 \ [\%] \qquad \text{Expression 3.17.11}$$

$Err_{Ttotal}$ = Total error for the last bit of the frame ［%］

$Err_{Tlength}$ = Error for the last bit of the frame that depends on the bit length ［%］

$Err_{Tdelay}$ = Error due to a synchronization delay occurring from the detection of a start bit until the beginning of reception ［%］

* For $Err_{Tdelay}$, take 0 or any greater absolute value of $Err_{Ttotal}$ for $\frac{B_M}{C \times S} \times 100$.

$Err_{Toffset}$ = Error due to the time offset from the beginning of reception until the first sampling time ［%］

L= Bit length of the frame

As shown above, perform calculations on a combination of N values for S values of 1/1, 1/2, 1/4, and 1/8 and determine the value of S and N so that the error represented by $Err_{Ttotal}$ is minimized.

It will become difficult to perform transmit or receive processing if the error represented by $Err_{Ttotal}$ goes over ±50%. Since errors caused by the opponent device and errors associated with clock frequency will also affect communication, use a bit rate such that the error represented by $Err_{Ttotal}$ falls within ±20% or so. Also, conduct extensive evaluations in a practical operating environment.

Examples of bit rate setting for a frame’s bit length of 12 bits (L=12) are shown in Tables 3.17.6 and 3.17.7.

### Table 3.17.6 Examples of Bit Rate Settings (L=12)

| L=12 | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $B_M$[bps] | C=1[MHz] | | | C=1.5[MHz] | | | C=2[MHz] | | | C=3[MHz] | | |
| | S | N | $Err_{Ttotal}$ [%] | S | N | $Err_{Ttotal}$ [%] | S | N | $Err_{Ttotal}$ [%] | S | N | $Err_{Ttotal}$ [%] |
| 134 | 1/8 | 232 | -1.2 | | | | | | | | | |
| 150 | 1/8 | 207 | -2.0 | | | | | | | | | |
| 300 | 1/4 | 207 | -2.0 | 1/8 | 155 | -2.1 | 1/8 | 207 | -2.0 | | | |
| 600 | 1/2 | 207 | -2.0 | 1/4 | 155 | -2.1 | 1/4 | 207 | -2.0 | 1/8 | 155 | -2.1 |
| 1200 | 1/1 | 207 | -2.0 | 1/2 | 155 | -2.1 | 1/2 | 207 | -2.0 | 1/4 | 155 | -2.1 |
| 1800 | 1/1 | 138 | 1.0 | 1/1 | 207 | -2.0 | 1/2 | 138 | 1.0 | 1/2 | 207 | -2.0 |
| 2400 | 1/1 | 103 | -2.2 | 1/1 | 155 | -2.1 | 1/1 | 207 | -2.0 | 1/2 | 155 | -2.1 |
| 4800 | 1/1 | 51 | -2.4 | 1/1 | 77 | -2.2 | 1/1 | 103 | -2.2 | 1/1 | 155 | -2.1 |
| 7200 | 1/1 | 34 | 9.6 | 1/1 | 51 | -2.4 | 1/1 | 68 | -8.0 | 1/1 | 103 | -2.2 |
| 9600 | 1/1 | 25 | -2.9 | 1/1 | 38 | -2.6 | 1/1 | 51 | -2.4 | 1/1 | 77 | -2.2 |
| 14400 | | | | 1/1 | 25 | -2.9 | 1/1 | 34 | 9.6 | 1/1 | 51 | -2.4 |
| 19200 | 1/1 | 12 | -3.8 | | | | 1/1 | 25 | -2.9 | 1/1 | 38 | -2.6 |
| 38400 | | | | | | | 1/1 | 12 | -3.8 | | | |
| 57600 | | | | | | | | | | 1/1 | 12 | -3.8 |
| 115200 | | | | | | | | | | | | |
| 128000 | | | | | | | | | | | | |

* A blank cell indicates settings that are invalid or values that cannot be used because the corresponding error is too great.

### Table 3.17.7 Examples of Bit Rate Settings (L=12)

| L=12 | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $B_M$[bps] | C=4[MHz] | | | C=6[MHz] | | | C=8[MHz] | | | C=12[MHz] | | |
| | S | N | $Err_{Ttotal}$ [%] | S | N | $Err_{Ttotal}$ [%] | S | N | $Err_{Ttotal}$ [%] | S | N | $Err_{Ttotal}$ [%] |
| 134 | | | | | | | | | | | | |
| 150 | | | | | | | | | | | | |
| 300 | | | | | | | | | | | | |
| 600 | 1/8 | 207 | -2.0 | | | | | | | | | |
| 1200 | 1/4 | 207 | -2.0 | 1/8 | 155 | -2.1 | 1/8 | 207 | -2.0 | | | |
| 1800 | 1/4 | 138 | 1.0 | 1/4 | 207 | -2.0 | 1/8 | 138 | 1.0 | 1/8 | 207 | -2.0 |
| 2400 | 1/2 | 207 | -2.0 | 1/4 | 155 | -2.1 | 1/4 | 207 | -2.0 | 1/8 | 155 | -2.1 |
| 4800 | 1/1 | 207 | -2.0 | 1/2 | 155 | -2.1 | 1/2 | 207 | -2.0 | 1/4 | 155 | -2.1 |
| 7200 | 1/1 | 138 | 1.0 | 1/1 | 207 | -2.0 | 1/2 | 138 | 1.0 | 1/2 | 207 | -2.0 |
| 9600 | 1/1 | 103 | -2.2 | 1/1 | 155 | -2.1 | 1/1 | 207 | -2.0 | 1/2 | 155 | -2.1 |
| 14400 | 1/1 | 68 | -8.0 | 1/1 | 103 | -2.2 | 1/1 | 138 | 1.0 | 1/1 | 207 | -2.0 |
| 19200 | 1/1 | 51 | -2.4 | 1/1 | 77 | -2.2 | 1/1 | 103 | -2.2 | 1/1 | 155 | -2.1 |
| 38400 | 1/1 | 25 | -2.9 | 1/1 | 38 | -2.6 | 1/1 | 51 | -2.4 | 1/1 | 77 | -2.2 |
| 57600 | | | | 1/1 | 25 | -2.9 | 1/1 | 34 | 9.6 | 1/1 | 51 | -2.4 |
| 115200 | | | | 1/1 | 12 | -3.8 | | | | 1/1 | 25 | -2.9 |
| 128000 | | | | | | | | | | | | |

* A blank cell indicates settings that are invalid or values that cannot be used because the corresponding error is too great.

### 3.17.6.4 Examples of transmit/receive operations

Examples of transmit and receive operations are explained below using flowcharts.

In each example, the initial values that are listed in Table 3.17.8 are assumed. The transmit and receive formats should be set up as listed in Table 3.17.9.

**Table 3.17.8 Initial Values**

- H- and L-level interrupt requests are enabled.
  - IE7 (FE08H, bit 7)=1
- Interrupt level setting for interrupt vector addresses 00033H and 0003BH
  - IP33 (FE09H, bit 4)=0 or 1
  - IP3B (FE09H, bit 5)=0 or 1
- System clock frequency
  - 6 [MHz]
- SCTX (P31) and SCRX (P32) port settings
  - P31 (FE4CH, bit 1)=0
  - P32 (FE4CH, bit 2)=0
  - P31DDR (FE4DH, bit 1)=0
  - P32DDR (FE4DH, bit 2)=0
  - The port option of P31 and P32 is set to CMOS.

**Table 3.17.9 Transmit/Receive Formats**

| Item | Setting |
|---|---|
| Bit rate | 4800[bps] |
| Data bit length | 8 bits |
| Stop bit length | 1 bit |
| First bit | LSB first |
| Binary inversion | Disabled |
| Parity | Used (even parity) |
| Clock source | System clock |
| SCUART clock frequency division ratio | 1/2 |
| Baudrate generator | 155 (9BH) |

1) Single frame transmission

In this example, only 1 frame is transmitted.

Figure 3.17.9 shows an example of a flowchart and Figure 3.17.10 shows an example of a transmit operation.

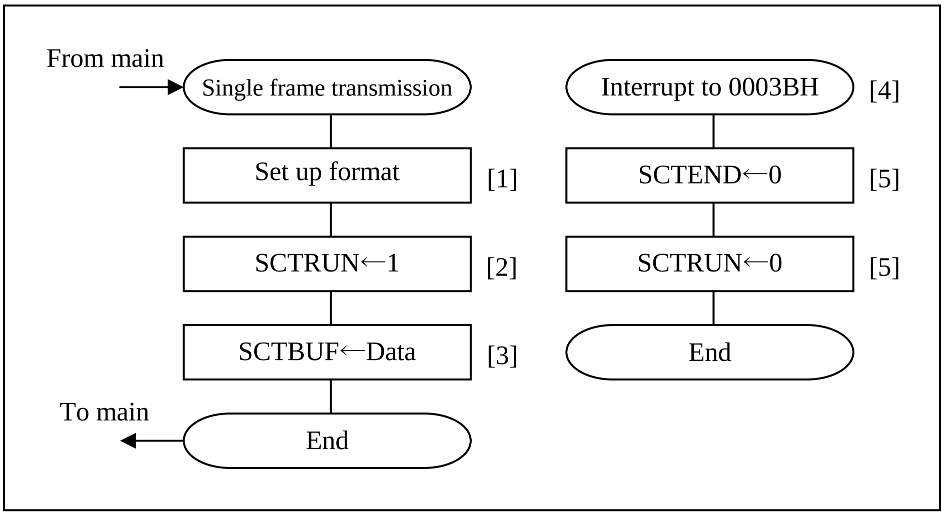


**Figure 3.17.9 Example of Single Frame Transmission Flowchart**

[1] Make the following settings:
- SCCNT0=84H (Enable transmit end interrupt request.)
- SCCNT1=00H
- SCMOD=14H
- SCBRG=9BH

[2] Start the transmit function.

[3] Write the transmit data into SCTBUF, and transmission starts.

[4] When the transmission terminates, an interrupt is generated.

[5] Stop the transmit function and perform its termination processing.

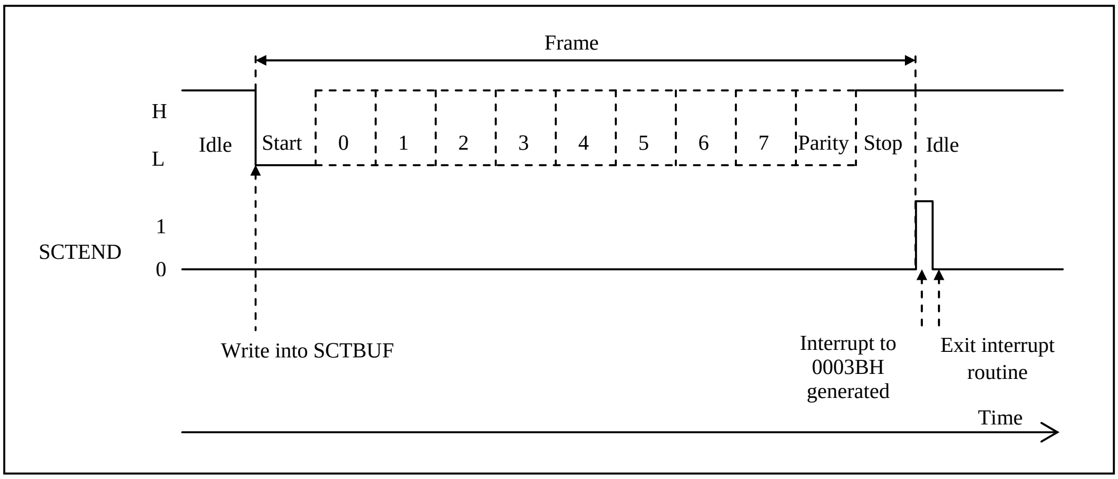


**Figure 3.17.10 Example of Single Frame Transmit operation**

2) Continuous transmission

In this example, 3 frames are transmitted consecutively.

Figure 3.17.11 shows an example of a flowchart and Figure 3.17.12 shows an example of a continuous transmit operation.

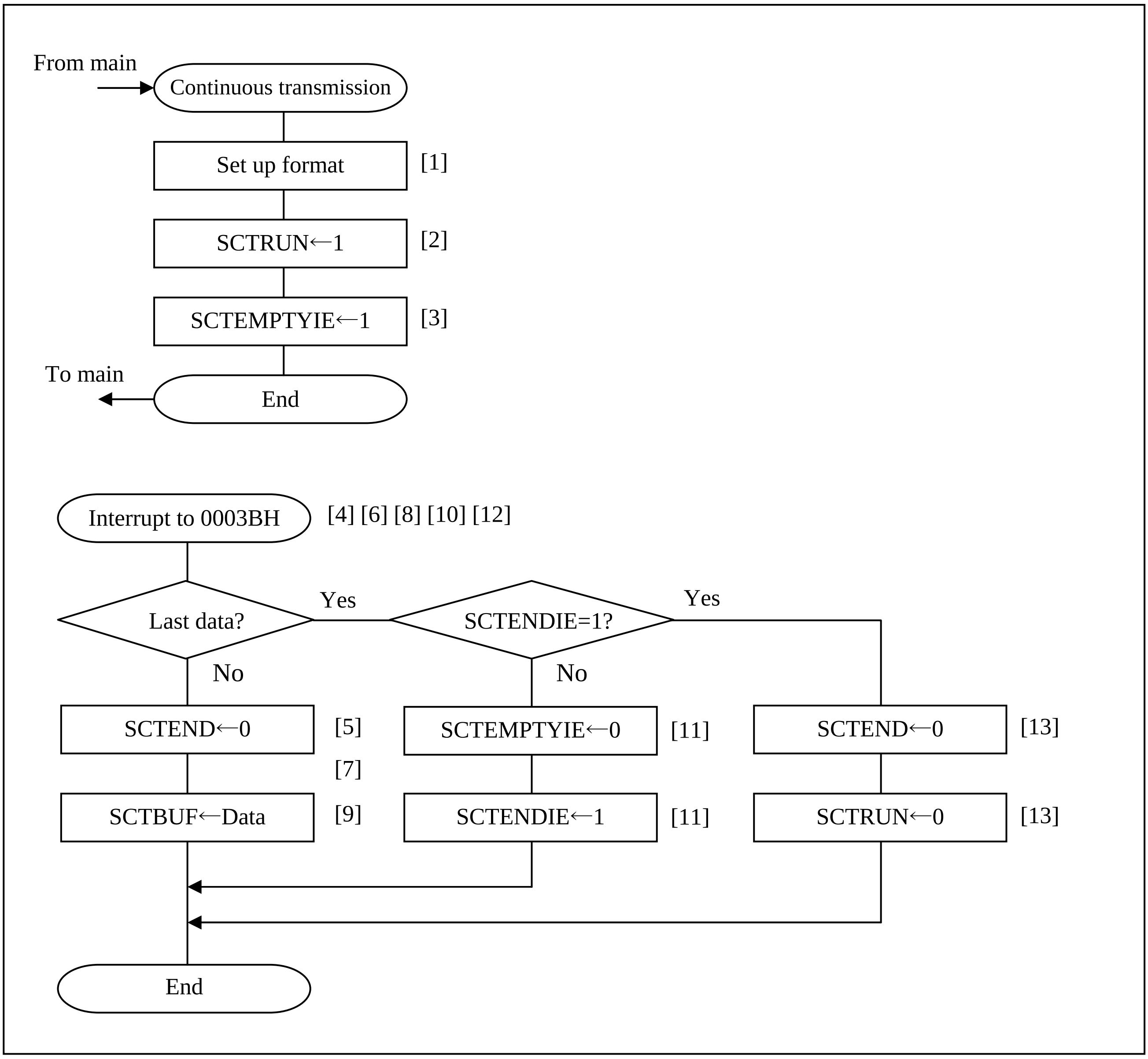


**Figure 3.17.11 Example of Continuous Transmission Flowchart**

[1] Make the following settings:

- SCCNT0=80H
- SCCNT1=00H
- SCMOD=14H
- SCBRG=9BH

[2] Start the transmit function.
(In this example, SCTEMPTY=1 is assumed.)

[3] Enable a transmit data register empty interrupt request.

[4] A transmit data register empty interrupt occurs.

[5] Clear the transmit end flag and write the first transmit data into SCTBUF, and transmission of the first frame starts.

[6] A transmit data register empty interrupt occurs.

[7] Clear the transmit end flag and write the second transmit data into SCTBUF.

[8] When the transmission of the second frame starts, a transmit data register empty interrupt occurs.

[9] Clear the transmit end flag and write the third transmit data into SCTBUF.

[10] When the transmission of the third frame starts, a transmit data register empty interrupt occurs.

[11] Disable a transmit data register empty interrupt request and enable a transmit end interrupt request.

[12] When the transmission of the third frame is terminated, an interrupt occurs.

[13] Stop the transmit function and perform termination processing.

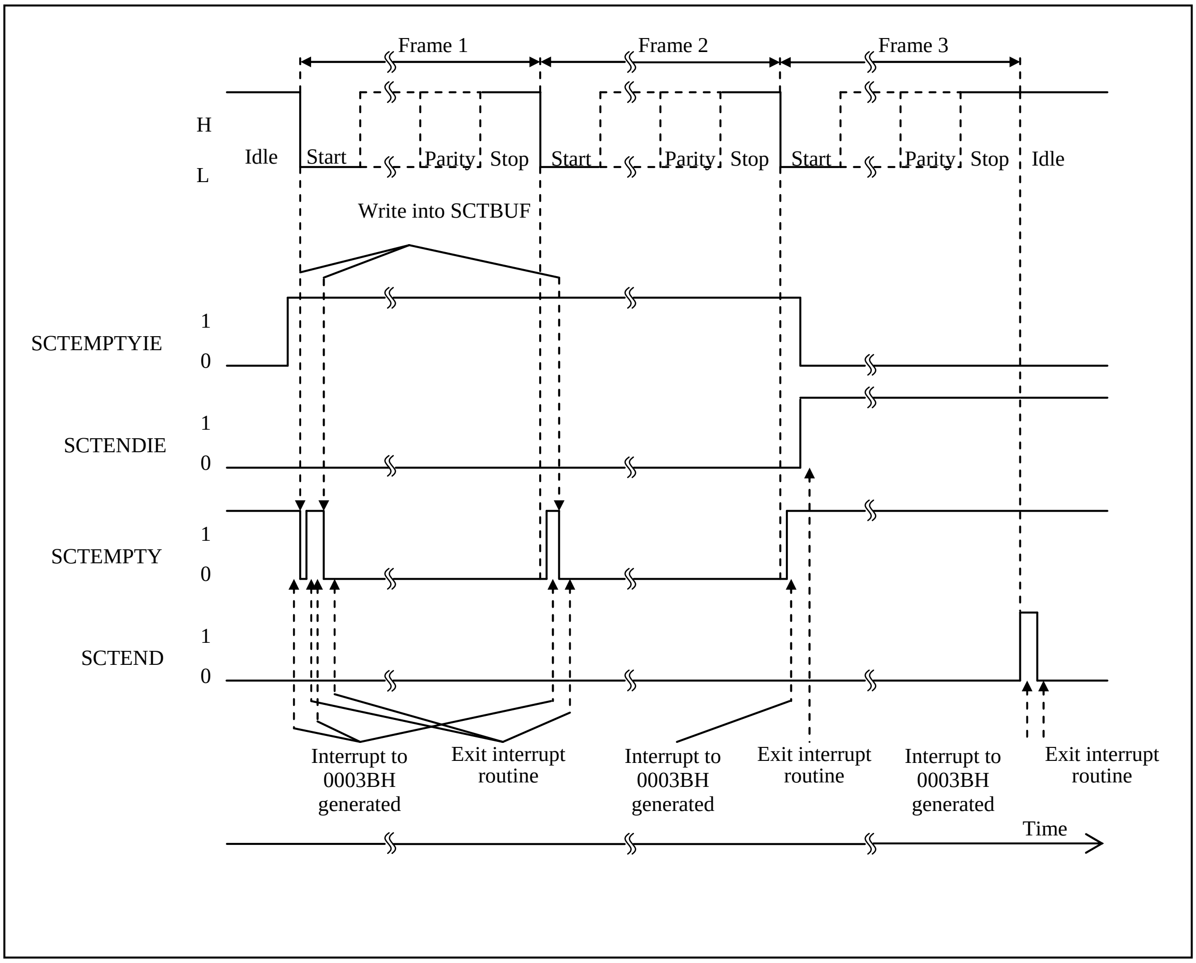


**Figure 3.17.12 Example of Continuous Transmit operation**

3) Single frame reception

In this example, only 1 frame is received.

Figure 3.17.13 shows an example of a flowchart and Figure 3.17.14 shows an example of a receive operation.

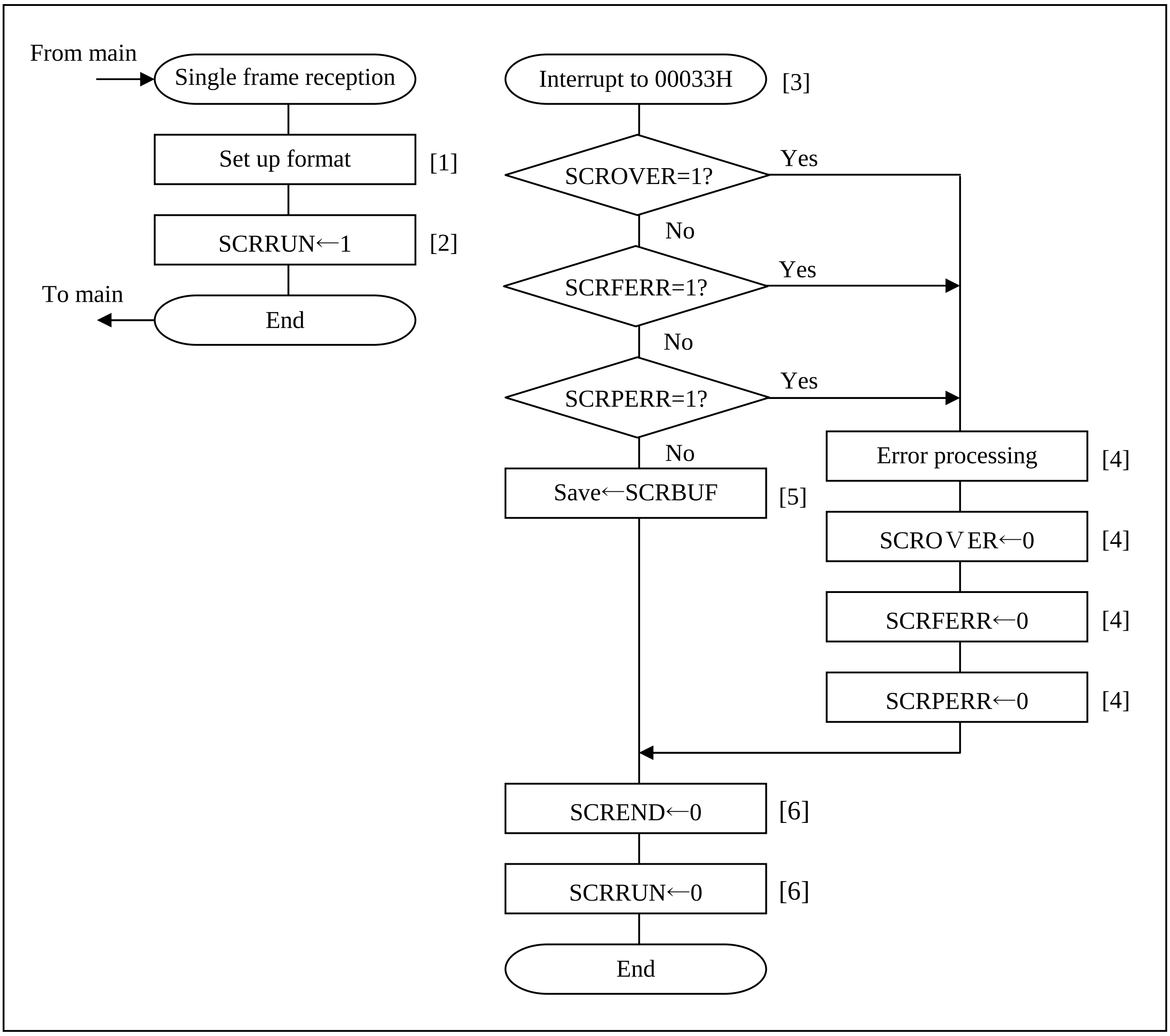


**Figure 3.17.13 Example of Single Frame Reception Flowchart**

[1] Make the following settings:

- SCCNT0=80H
- SCCNT1=01H (Enable receive end interrupt requests.)
- SCMOD=14H
- SCBRG=9BH

[2] Start the receive function.

[3] When the reception terminates, an interrupt occurs.

[4] Check the error flags and, if an error is found, execute the necessary error processing and clear the pertinent error flag. Table 3.17.10 shows a list of error flags related to the receive operation.

[5] Save the received data from SCRBUF.

[6] Stop the receive function and perform termination processing.

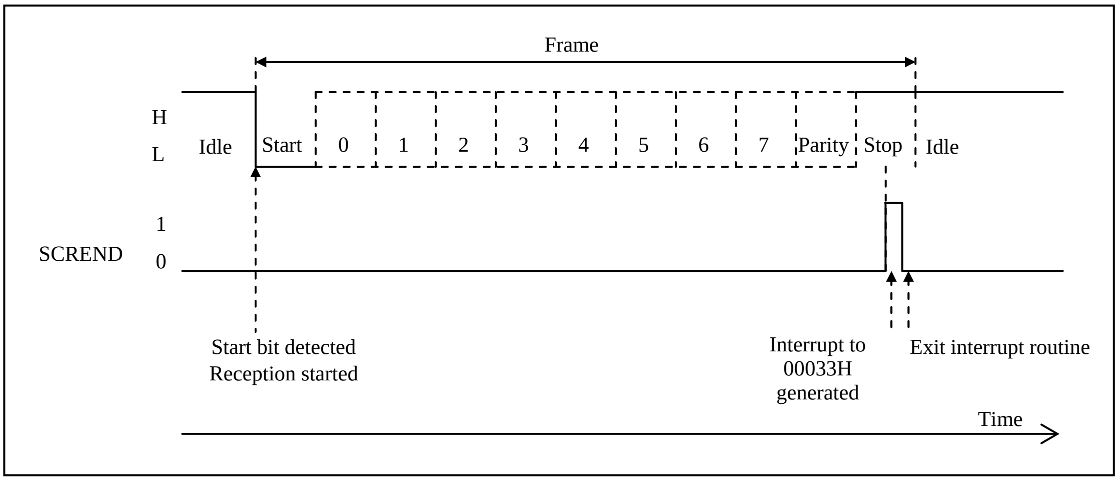


**Figure 3.17.14 Example of Single Frame Receive Operation**

**Table 3.17.10 List of Error Flags Associated with the Receive Operation**

| Receive Error Name | Flag Name | Flag Setting Condition | Flag Setting Timing | Receive Data |
|---|---|---|---|---|
| Overrun error | SCROVER | SCREND is set on termination of reception with SCREND set to 1. | At the same time SCREND is set on termination of reception. | The receive data register (SCRBUF) is updated with receive data. |
| Framing error | SCRFERR | A low level is detected while checking the stop bit. | | |
| Parity error | SCRPERR | An error occurs while checking the parity. | | |

## 3.17.7 SCUART HALT Mode Operation

### 3.17.7.1 Transmission

The transmitter circuit remains active even in HALT mode.

Consequently, when the microcontroller is placed in HALT mode after a transmit operation is started (in the middle of transmission), that transmission is carried out to its end. When the microcontroller is placed in HALT mode after the next transmit data is loaded in continuous transmission mode, the transmission of the next frame is carried out to its end.

Transmission retries due to errors enabled by setting SCERCE to 1 are continued until transmission retries terminate normally before the count reaches the preset repeat count or until none of them terminate normally and the preset repeat count is exhausted. If the next transmit data is loaded for continuous transmission when retransmissions have terminated normally before the preset repeat count is reached, the transmission of that data is carried out to its end.

HALT mode can be exited via a transmit data register empty or a transmit end interrupt.

### 3.17.7.2 Reception

The receiver circuit remains active even in HALT mode.

Consequently, when the microcontroller is placed in HALT mode in the middle of reception, that reception is carried out to its end. Furthermore, a receive operation is started every time a start bit is detected.

This also applies to reception retries due to errors that are enabled by setting SCERCE to 1.

HALT mode can be exited via a receive end interrupt.

## 3.17.8 Using the SCUART for Communication with a Smart Card

This subsection describes the use of the SCUART for communication with a smart card.

### 3.17.8.1 Overview

This asynchronous serial interface also serves as an interface for IC cards (smart cards) that conform to ISO/IEC7816-3.

It is provided with the functions that are necessary for supporting protocol type T = 0.

- Detects error signals (parity errors) during transmission and retries transmit (repeat) operation

  Detects error signals that are transmitted on parity errors on the receiver side and retries transmission.
- Transmits error signals (parity errors) during reception and retries receive (repeat) operation

  Transmits error signals when parity errors are detected in the receive data and retries reception.

This function is enabled by setting SCERCE to 1.

The SCUART does not provide the above-mentioned functions for protocol type T=1. In this case the SCUART must be used with SCERCE set to 0.

### 3.17.8.2 Configuration

1) When using the SCUART for protocol type T=0

   Set SCERCE to 1 to use the SCUART for protocol type T=0. When SCERCE is set to 1, part of the function settings of the SCUART special function registers are modified so that the configuration items for the smart card interface can be set to your preference and part of the functions are automatically configured. Table 3.17.11 shows a list of registers whose function settings are changed when SCERCE is set to 1 and Table 3.17.12 shows a list of functions that are automatically configured when SCERCE is set to 1.

   Perform settings according to the T=0 format while referring to 3.17.5.

**Table 3.17.11 List of Registers Whose Functions are Modified When SCERCE is Set to 1**

| Address | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|
| FECA | SCCNT0 | | SCTSTP | | SCTERR | SCTEND | | | |
| FECB | SCCNT1 | | SCCHRL | | | SCRFERR | SCRPERR | SCREND | |
| FECC | SCMOD | | | | | | SCPEN | | |

**Table 3.17.12 List of Functions That are Automatically Configured When SCERCE is Set to 1**

| |
|---|
| ・A frame is made up of 10 bits.<br>(start bit, data bits (8 bits), parity bit) |
| ・A parity bit is appended on transmission and parity check is performed on reception.<br>(The parity (even or odd) needs to be defined.) |
| ・No receive framing errors are detected.<br>(The stop bit is not checked on reception.) |
| ・Detection of transmission-time error signals and sending of reception-time error signals are enabled. |
| ・Repeated retransmission and re-reception on errors are enabled.<br>(The repeat count and retransmission wait time must be defined.) |

2) When using the SCUART for protocol type T=1

Set SCERCE to 0 to use the SCUART for protocol type T=1. Make settings according to the T=1 format while referring to 3.17.5. When SCERCE is set to 0, the receive framing error flag (stop bit checking) function is enabled but it is not required for communication with smart cards. When referring to 3.17.5, read the stop bit length as the guard time, 1- and 2-bit bit lengths as 1 etu and 2 etu, respectively. The stop bit state is "high" as pulled up by an external pull-up resistor if the connection and configuration that are described in 3.17.8.3 are used.

### 3.17.8.3 Pin connection example

Figure 3.17.15 shows the example of the SCUART pin connection configuration for communication with a smart card. The corresponding settings are summarized in Table 3.17.13.

Using these settings, the SCTX and SCRX pins can only be set to low output; they cannot generate high-level output. When generating a high-level output, the pin is placed in a high-impedance (open) state and set high by the externally connected pull-up resistor.

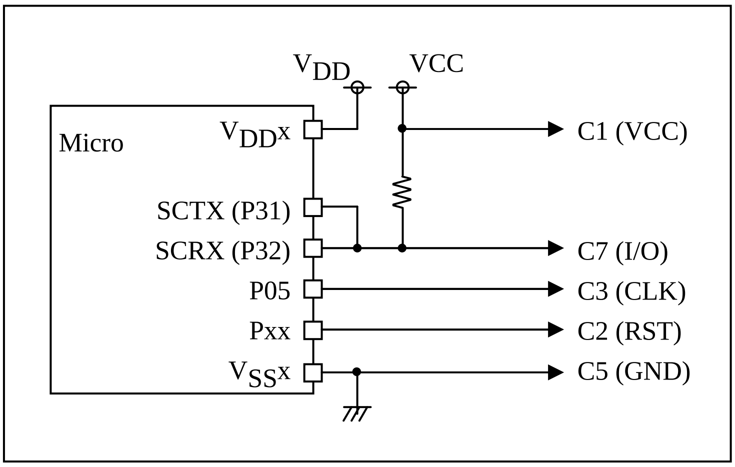


**Figure 3.17.15 Example of Pin Configuration for Connection with a Smart Card**

**Table 3.17.13 Example of Pin Settings for Connection with a Smart Card**

| Pin Name | I/O | Function and Setting |
|---|---|---|
| SCTX (P31) | Input/ Output | Used to transmit data and detect error signals.<br>SCTDDR (FECAH, bit 7) must be set to 1.<br>P31 (FE4CH, bit 1) must be set to 0.<br>P31DDR (FE4DH, bit 1) must be set to 0.<br>The port option of P31 must be set to N-channel open drain. |
| SCRX (P32) | Input/ Output | Used to receive data and transmit error signals.<br>P32 (FE4CH, bit 2) must be set to 0.<br>P32DDR (FE4DH, bit 2) must be set to 0.<br>The port option of P32 can be set to either N-channel open drain or CMOS. |
| P05 | Output | Transmits a selected clock.<br>See the page on port 0.<br>The port option of P05 must be set to CMOS. |
| Pxx | Output | General-purpose ports must be used.<br>See the page on the pertinent port.<br>The port option of Pxx must be set to CMOS. |

#### 3.17.8.4 Card clock

Tables 3.17.14 and 3.17.15 show the sample communication speed settings for communication with a card using the P05 clock output function for supplying clocks to the card. 1 etu in this case can be calculated using the expression 3.17.12.

Since communication with the card is done asynchronously even when the SCUART is run while supplying clocks to the card from P05, errors can occur during mutual communications. The valid transmission/reception time per bit (bit time) for this interface can be obtained using the formula that is given by expression 3.17.4. The difference between that bit time and the definition of 1 etu which is calculated using expression 3.17.12 constitutes the error. Another error includes the error that is caused by the synchronization delay that develops at the beginning of reception as shown in Figure 3.17.6.

Tables 3.17.14 and 3.17.15 show the overall errors at the last communication bit in virtual communication at 13 etu (L=13). In the tables, blank cells indicate settings that cannot be used because they produce errors that are too large. Shaded cells indicate that the error is extremely large. When making practical use of the SCUART, conduct extensive evaluations in a practical operating environment.

$$１\ \mathrm{etu} = \frac{F}{D} \times \frac{1}{\text{P05 output clock (card clock) frequency [Hz]}} [\mathrm{s}] \qquad \text{Expression 3.17.12}$$

etu (elementary time unit): Transmit/receive time per bit (bit time)

F (clock rate conversion integer)

D (baudrate adjustment integer)

**Table 3.17.14 Example of Communication Speed Settings for Communication with a Card**

| Frequency Ratio<br>P05 Output Clock:<br>SCUART Clock Source | F | D | SCDIVx<br>Division Ratio<br>Value S | SCBRG<br>Value N | Err$_{Ttotal}$ [%]<br>(L=13) |
|---|---|---|---|---|---|
| 1: 8 | 372 | 1 | 1/4 | 185 | -0.1 |
| | | 2 | 1/2 | 185 | -0.1 |
| | | 4 | 1/1 | 185 | -0.1 |
| | | 8 | 1/1 | 92 | -0.3 |
| | | 12 | 1/1 | 61 | -0.4 |
| | | 16 | 1/1 | 46 | 14 |
| | | 20 | 1/1 | 36 | -7.7 |
| | | 32 | 1/1 | 22 | -15.1 |
| | 512 | 1 | 1/4 | 255 | -0.1 |
| | | 2 | 1/2 | 255 | -0.1 |
| | | 4 | 1/1 | 255 | -0.1 |
| | | 8 | 1/1 | 127 | -0.2 |
| | | 12 | 1/1 | 84 | -5.4 |
| | | 16 | 1/1 | 63 | -0.4 |
| | | 20 | 1/1 | 50 | -5.6 |
| | | 32 | 1/1 | 31 | -0.8 |
| 1: 4 | 372 | 1 | 1/2 | 185 | -0.1 |
| | | 2 | 1/1 | 185 | -0.1 |
| | | 4 | 1/1 | 92 | -0.3 |
| | | 8 | 1/1 | 46 | 14 |
| | | 12 | 1/1 | 30 | -0.8 |
| | | 16 | 1/1 | 22 | -15.1 |
| | | 20 | | | |
| | | 32 | | | |
| | 512 | 1 | 1/2 | 255 | -0.1 |
| | | 2 | 1/1 | 255 | -0.1 |
| | | 4 | 1/1 | 127 | -0.2 |
| | | 8 | 1/1 | 63 | -0.4 |
| | | 12 | 1/1 | 42 | 10.2 |
| | | 16 | 1/1 | 31 | -0.8 |
| | | 20 | 1/2 | 12 | 20.3 |
| | | 32 | 1/1 | 15 | -1.6 |

**Table 3.17.15 Example of Communication Speed Settings for Communication with a Card**

| Frequency Ratio<br>P05 Output Clock:<br>SCUART Clock Source | F | D | SCDIVx<br>Division Ratio<br>Value S | SCBRG<br>Value N | Err$_{Ttotal}$ [%]<br>(L=13) |
|---|---|---|---|---|---|
| 1: 2 | 372 | 1 | 1/1 | 185 | -0.1 |
| | | 2 | 1/1 | 92 | -0.3 |
| | | 4 | 1/1 | 46 | 14 |
| | | 8 | 1/1 | 22 | -15.1 |
| | | 12 | | | |
| | | 16 | | | |
| | | 20 | | | |
| | | 32 | | | |
| | 512 | 1 | 1/1 | 255 | -0.1 |
| | | 2 | 1/1 | 127 | -0.2 |
| | | 4 | 1/1 | 63 | -0.4 |
| | | 8 | 1/1 | 31 | -0.8 |
| | | 12 | 1/1 | 20 | -21.5 |
| | | 16 | 1/1 | 15 | -1.6 |
| | | 20 | 1/1 | 12 | 20.3 |
| | | 32 | 1/1 | 7 | -3.1 |
| 1: 1 | 372 | 1 | 1/1 | 92 | -0.3 |
| | | 2 | 1/1 | 46 | 14 |
| | | 4 | 1/1 | 22 | -15.1 |
| | | 8 | | | |
| | | 12 | | | |
| | | 16 | | | |
| | | 20 | | | |
| | | 32 | | | |
| | 512 | 1 | 1/1 | 127 | -0.2 |
| | | 2 | 1/1 | 63 | -0.4 |
| | | 4 | 1/1 | 31 | -0.8 |
| | | 8 | 1/1 | 15 | -1.6 |
| | | 12 | | | |
| | | 16 | 1/1 | 7 | -3.1 |
| | | 20 | | | |
| | | 32 | 1/1 | 3 | -6.3 |

### 3.17.8.5 Frame format

The transmit/ receive frame format to be used with SCERCE set to 1 differs from the format which is to be used with SCERCE set to 0.

The data transmission/reception timing when used with SCERCE set to 1 is the same as the timing when used with SCERCE set to 0 as shown in Figure 3.17.7.

A description of the frame format that is to be used with SCERCE set to 1 follows.

The configuration of the frame is determined by the contents of Table 3.17.16. These settings determine the format of the frame. The settings for the "parity bit," "first bit," and "data binary inversion" listed in Table 3.17.16 are the same as those for the frame format that is to be used with SCERCE set to 0.

**Table 3.17.16 Frame Configuration and Format to Be Used with SCERCE Set to 1**

| Item | Setting |
|---|---|
| Start bit | - |
| Stop bit | No stop bit |
| Data bit length | 8 bits fixed |
| Parity bit | Even/odd |
| First bit | LSB/MSB first |
| Data (including parity bit) binary inversion | Yes/No |
| Error-time retransmission/re-reception repeat count | 0/3/4 times |
| Wait period until retransmission after detection of an error signal | 2 or 3 etu |
| Period for sending an error signal on a receive parity error | 1 etu fixed |

1) When no error signal is present

- Transmission

Figure 3.17.16 shows an example of a frame configuration in a case in which no error signal is present for the data that is transmitted. The SCUART attempts to detect an error signal 1 etu after transmitting data (after the parity bit). Since no error signal is found, the SCUART terminates the transmit operation in another 1 etu.

- Reception

If the received data contains no error, the SCUART transmits no error signal and terminates the receive operation in 10.5 etu after the beginning of the receive operation as shown in Figure 3.17.16.

* There is a synchronization delay shown in Figure 3.17.6 at the beginning of the receive operation.

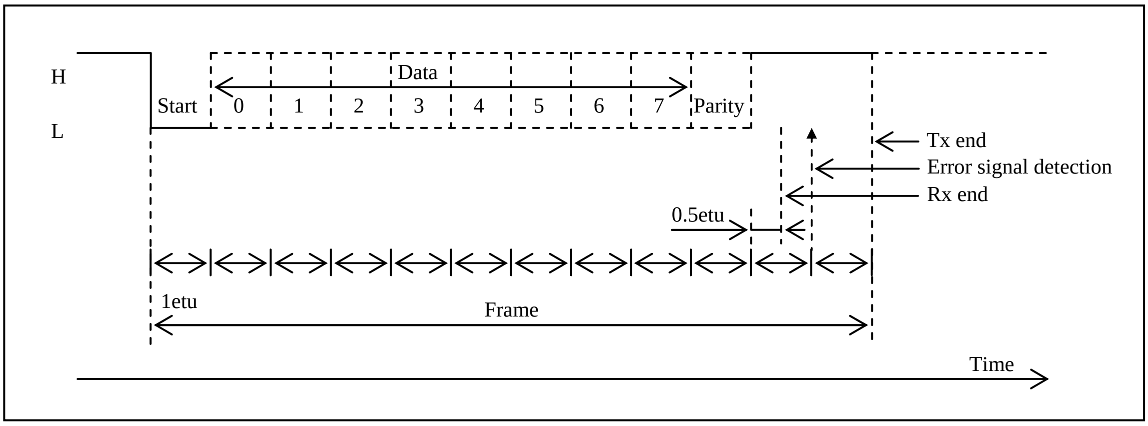


**Figure 3.17.16 Example of Frame Configuration When No Error Signal is Present**

2) When an error signal is present

- Transmission

Figure 3.17.17 shows an example of a frame configuration in a case in which an error signal is present for the data that is transmitted. Since there is an error signal that is detected 1 etu after the SCUART transmitted data (after the parity bit), the subsequent behavior of the SCUART differs depending on the settings of the “error-time retransmission/re-reception repeat count” and the “wait period until retransmission after detection of an error signal” shown in Table 3.17.16. In the example shown in Figure 3.17.17, the “error-time retransmission/re-reception repeat count” is set to 0. In this case, the SCUART tries no retransmission and terminates the transmit operation 1 etu after it detects an error signal. In transmission, any error signal is transmitted from the card side and the wait time is not necessarily 1 etu.

- Reception

If the received data contains an error, the SCUART transmits a 1-etu period error signal 10.5 etu after the beginning of the receive operation as shown in Figure 3.17.17. The subsequent behavior of the SCUART differs depending on the settings of the “error-time retransmission/ re-reception repeat count” shown in Table 3.17.16. In the example shown in Figure 3.17.17, the “error-time retransmission/re-reception repeat count” is set to 0. In this case, the SCUART tries no re-receptions and terminates the receive operation after transmitting a 1-etu period error signal.

* There is a synchronization delay shown in Figure 3.17.6 at the beginning of the receive operation.

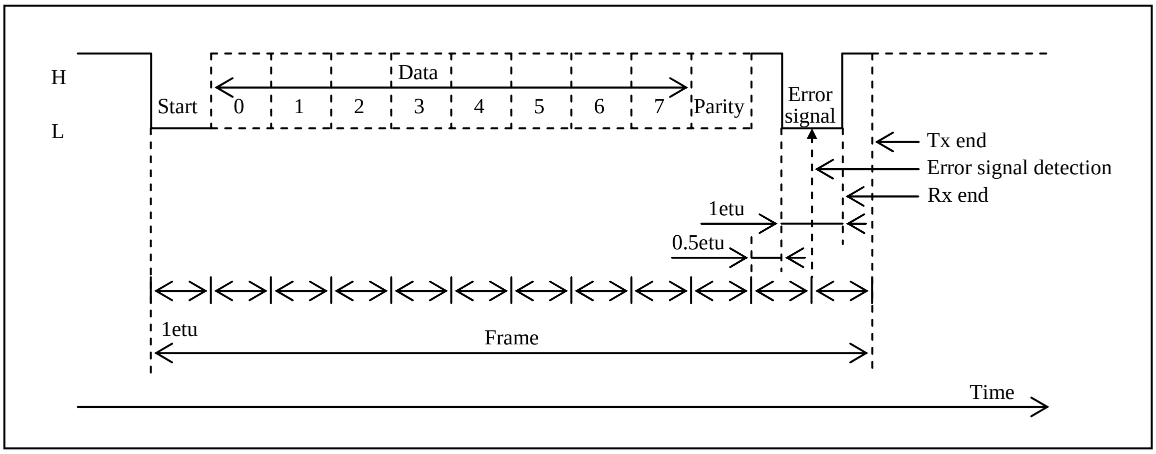


**Figure 3.17.17 Example of Frame Configuration When an Error Signal is Present (Repeat count being set to 0 or in the last repeat cycle when repeat count is set to 3 or 4)**

3) Repeat operation on error

Figure 3.17.18 shows a diagram of an image of the repeat operation that is performed according to the retransmission/re-reception repeat count settings. The figure illustrates a case in which a maximum number of repeat operations are performed according to the set repeat count setting. When the SCUART terminates a repeat operation normally within the set repeat count, the retransmission/re-reception cycle is terminated. The frame configuration at that time is shown in Figure 3.17.16.

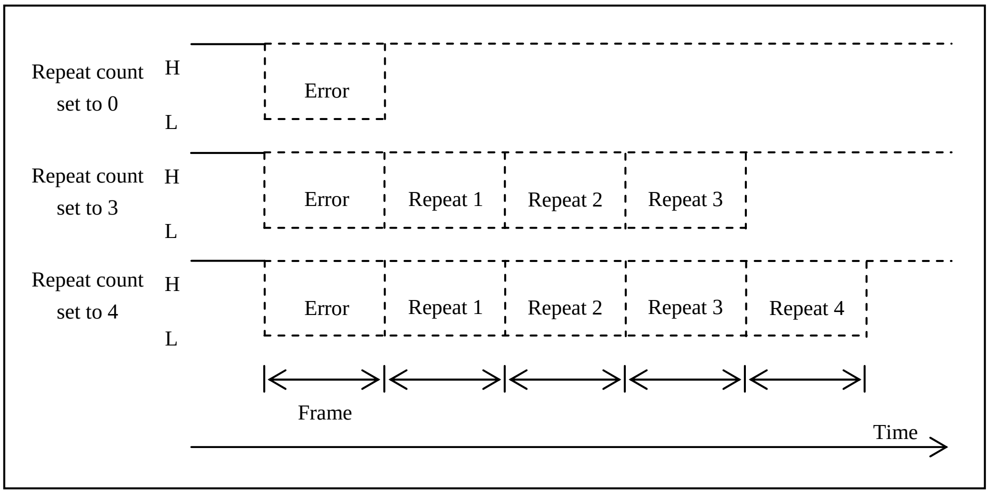


**Figure 3.17.18 Maximum Number of Repeat Operations Specified by Repeat Count Setting**

4) Wait period until retransmission in repeat operation

The frame configuration to be used for transmission when the error-time retransmit/re-receive repeat count is set to 3 or 4 varies depending on the setting of the wait period until retransmission after detection of an error signal. As shown in Figure 3.17.19, the SCUART starts retransmission after the lapse of the preset wait period until retransmission after detection of an error signal. The error signal shown in Figure 3.17.19 is transmitted by the card side and is not necessarily 1 etu. The SCUART retries the number of consecutive transmissions equal to the preset repeat count. The frame configuration established when the last repeated operation ends in error is shown in Figure 3.17.17.

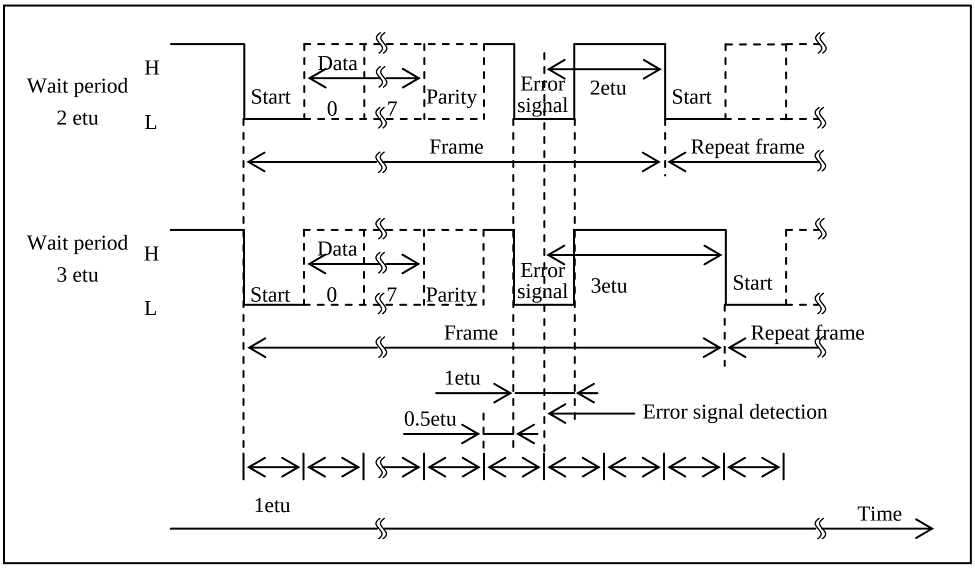


**Figure 3.17.19 Example of Wait Period until Retransmit Setting and Corresponding Frame Configuration**

5) Continuous transmission with SCERCE set to 1

The SCUART can perform continuous transmission when SCERCE is set to 1 in the same way as when SCERCE is set to 0 (see paragraph 4) in 3.17.6.2). In continuous transmission with SCERCE set to 1, the interval between the parity bit of the current frame and the start bit of the next frame is 2 etu as shown in Figure 3.17.20. Consequently, the frame between the beginning of the start bit of the current frame to the beginning of the start bit of the next frame is 12 etu. This does not hold, however, if an error signal is present, as in that case repeat operations are involved as controlled by the frame format settings.

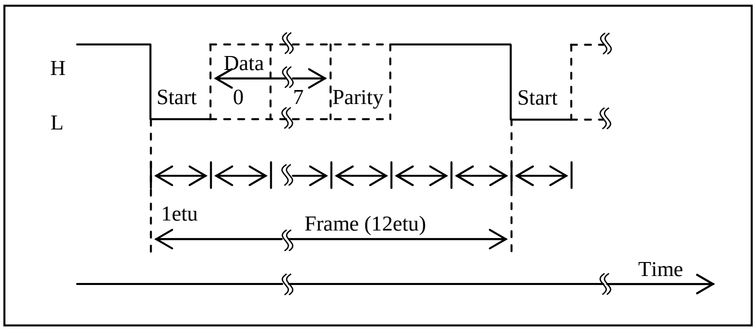


**Figure 3.17.20 Example of Continuous Transmission with SCERCE Set to 1**

### 3.17.8.6 Direct convention/inverse convention

Table 3.17.17 summarizes the settings for direct and inverse conventions and Figure 3.17.21 shows an example of the corresponding waveforms. The sample waveforms shown in Figure 3.17.21 are used for the card side to specify the communication format during the initial communication between the SCUART and the card. The card transmits 3BH for direct convention and 3FH for inverse convention. The settings listed in Table 3.17.17 must be performed according to the format that is specified.

**Table 3.17.17 Direct and Inverse Convention Settings**

| Item | Direct Convention | Inverse Convention |
|---|---|---|
| Parity bit | Even | Even |
| First bit | LSB first | MSB first |
| Data (including parity bit) binary inversion | No | Yes |

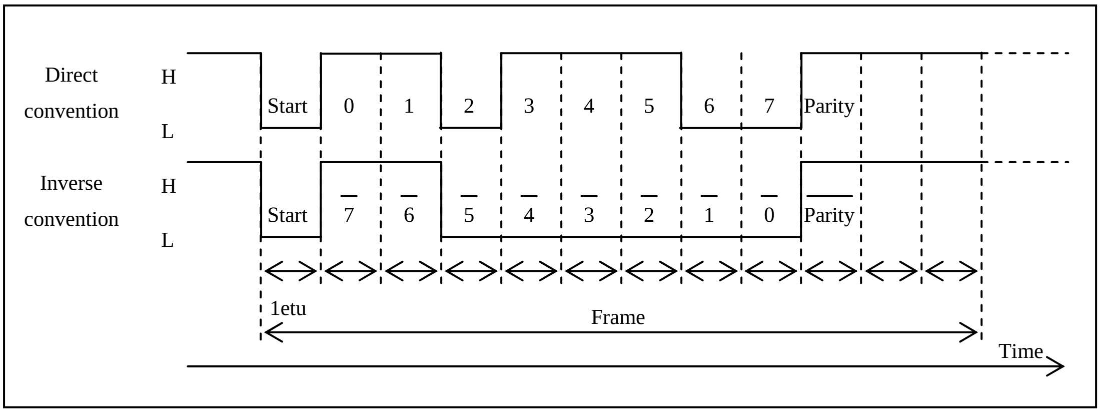


**Figure 3.17.21 Example of Waveforms for Direct Convention and Inverse Convention**

# 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 high-current P-channel driver output and PWM1 as high-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<br>FE24, bit 2 | PWM0L3<br>FE20, bit 7 | PWM0L2<br>FE20, bit 6 | PWM0L1, 0<br>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<br>FE24, bit 3 | PWM1L3<br>FE22, bit 7 | PWM1L2<br>FE22, bit 6 | PWM1L1, 0<br>FE22, bits 5, 4 |
|---|---|---|---|---|
| Hi-Z | 0 | — | 0 | — |
| Low | 0 | 0 | 1 | 0, 0 |
| High | 0 | 1 | 1 | 0, 0 |

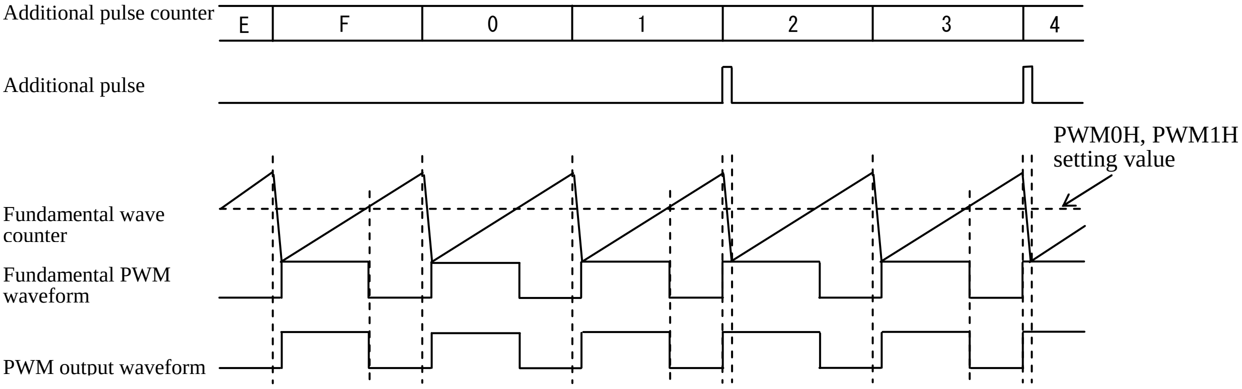
### 3.18.4.5 PWM1 compare register H (PWM1H) (8-bit register)

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

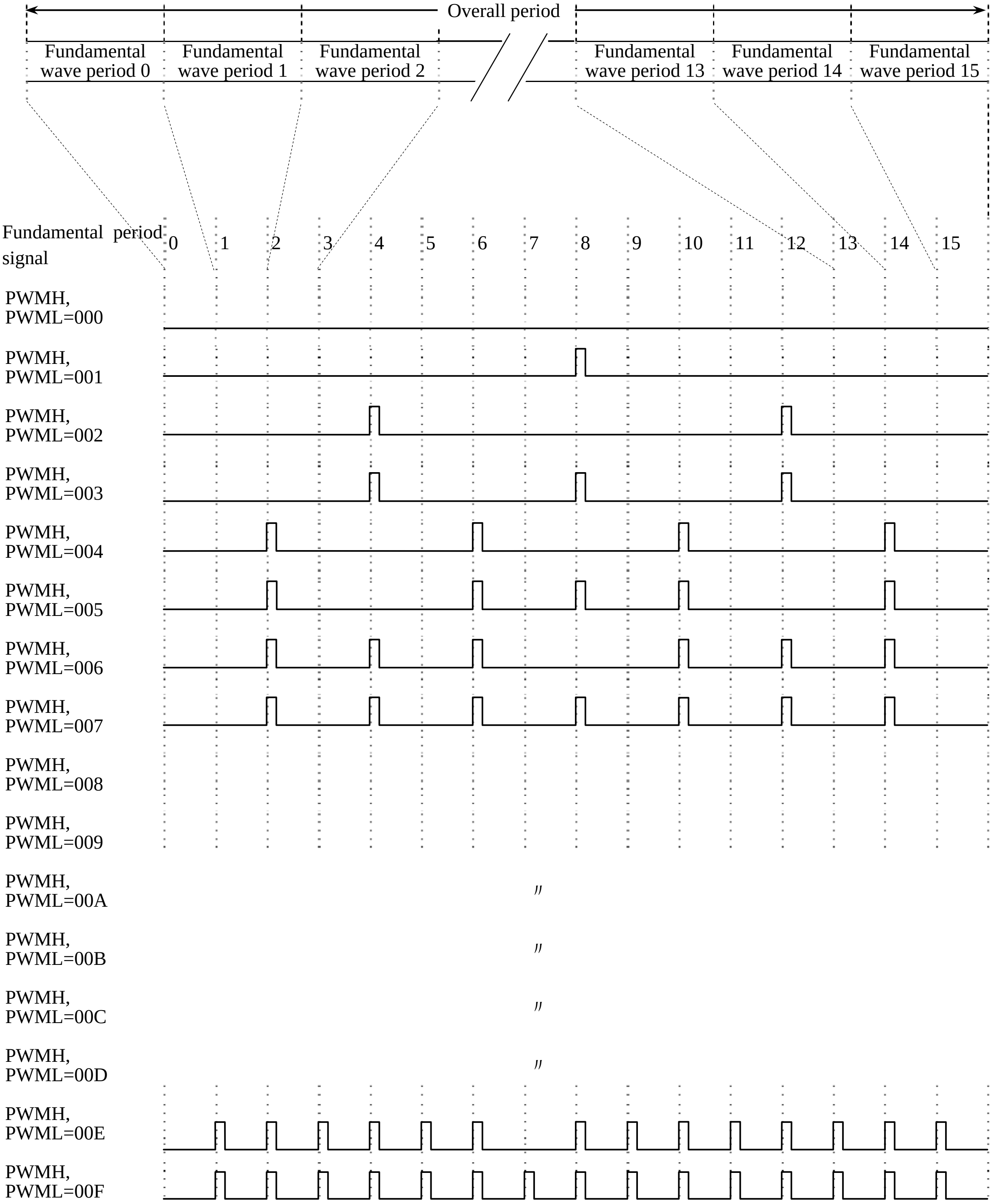
2) When bits 7 to 4 of PWM1L are all fixed at 0, PWM1 can serve as period-programmable 8-bit PWM that is controlled by PWM1H.

| 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 |

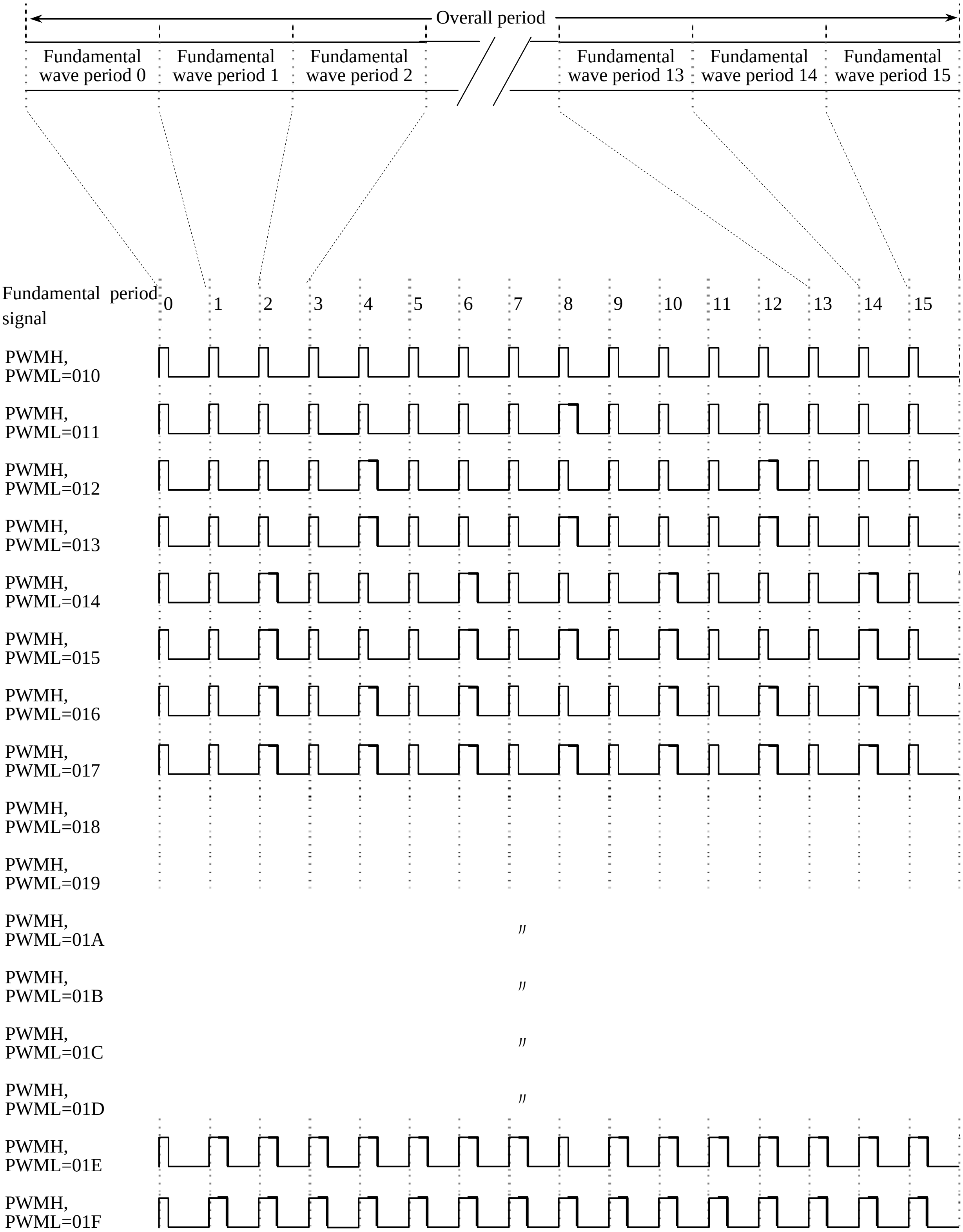


- The 12-bit PWM has the following waveform structure:
  - The overall period consists of 16 fundamental wave periods.
  - A fundamental wave period is represented by an 8-bit PWM. (PWM compare register H) (PWMH)
  - 4 bits are used to designate the fundamental wave period to which additional pulses 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]



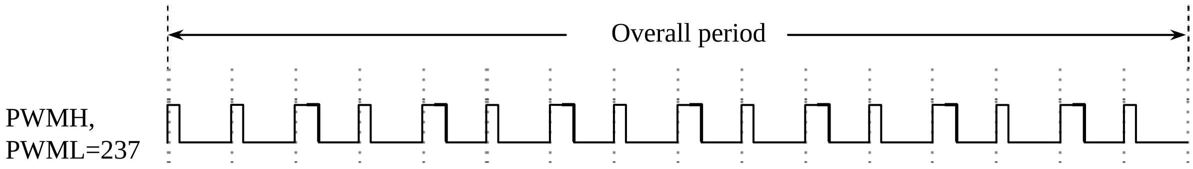
- The fundamental wave period is variable within the range of $\frac{(16 \text{ to } 256)}{3}$ Tcyc.
  Fundamental wave period = (Value represented by PWM0C7 to PWM0C4 + 1) × $\frac{16}{3}$ Tcyc
  - The overall period can be changed by changing the fundamental wave period.
  - The overall period is made up of 16 fundamental wave periods.

**Examples:**

- Wave comparison when the 12-bit PWM contains 237[H].

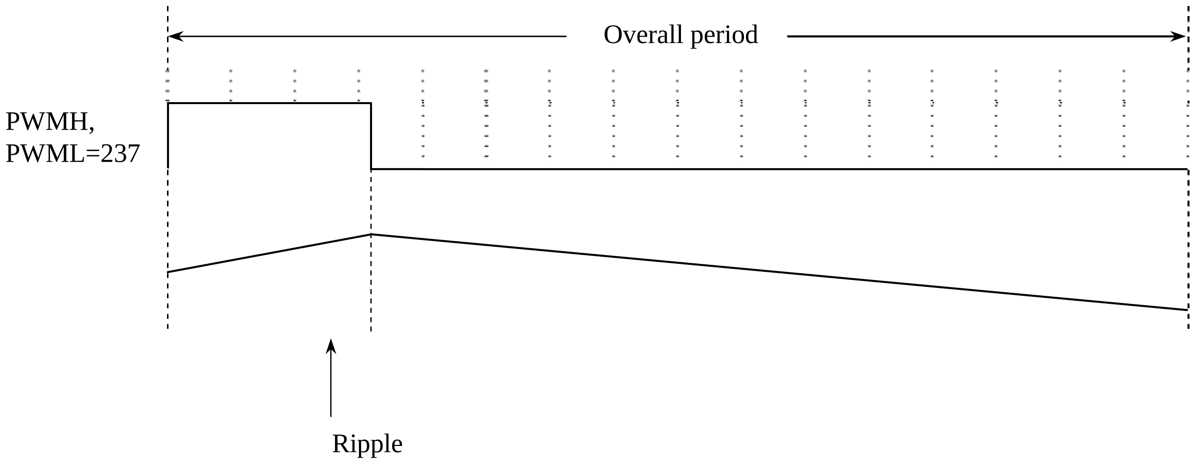
  12-bit register configuration → (PWMH), (PWML) = 237[H]

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) 20-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) 20-channel analog input

   The signal to be converted is selected using the AD control register (ADCRC) and high-current driver control register (BUFCNT) from 20 types of analog signals that are supplied from pins P0, P2, PWM0, PWM1, 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, P2INH, BUFCNT

| 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 |
| FE67 | 0000 0000 | R/W | P2INH | P27INH | P26INH | P25INH | P24INH | P23INH | P22INH | P21INH | P20INH |
| FEFF | 0000 0000 | R/W | BUFCNT | AD CHSEL4 | BUFCNT6 | BUFP1 | BUFP0 | BUFCNT3 | BUFN2 | BUFN1 | BUFN0 |

## 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 20 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 |
| FEFF | 0000 0000 | R/W | BUFCNT | ADCHSEL4 | BUFCNT6 | BUFP1 | BUFP0 | BUFCNT3 | BUFN2 | BUFN1 | BUFN0 |

**ADCHSEL4 (BUFCNT, bit 7):**
**ADCHSEL3 (ADCRC, bit 7):**
**ADCHSEL2 (ADCRC, bit 6):**
**ADCHSEL1 (ADCRC, bit 5):**
**ADCHSEL0 (ADCRC, bit 4)**

**AD conversion input signal (AN0 to AN19) select**

These 5 bits, bit 7 of the high-current driver control register (BUFCNT:FEFF) and bits 7 to 4 of the AD control register (ADCRC), are used to select the signal to be subject to AD conversion.

| AD CHSEL4 | AD CHSEL3 | AD CHSEL2 | AD CHSEL1 | AD CHSEL0 | Signal Input Pin |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | P00/AN0 |
| 0 | 0 | 0 | 0 | 1 | P01/AN1 |
| 0 | 0 | 0 | 1 | 0 | P02/AN2 |
| 0 | 0 | 0 | 1 | 1 | P03/AN3 |
| 0 | 0 | 1 | 0 | 0 | P04/AN4 |
| 0 | 0 | 1 | 0 | 1 | P05/AN5 |
| 0 | 0 | 1 | 1 | 0 | P06/AN6 |
| 0 | 0 | 1 | 1 | 1 | P07/AN7 |
| 0 | 1 | 0 | 0 | 0 | PWM1/AN8 |
| 0 | 1 | 0 | 0 | 1 | PWM0/AN9 |
| 0 | 1 | 0 | 1 | 0 | XT1/AN10 |
| 0 | 1 | 0 | 1 | 1 | XT2/AN11 |
| 0 | 1 | 1 | 0 | 0 | P20/AN12 |
| 0 | 1 | 1 | 0 | 1 | P21/AN13 |
| 0 | 1 | 1 | 1 | 0 | P22/AN15 |
| 0 | 1 | 1 | 1 | 1 | P23/AN15 |
| 1 | 0 | 0 | 0 | 0 | P24/AN16 |
| 1 | 0 | 0 | 0 | 1 | P25/AN17 |
| 1 | 0 | 0 | 1 | 0 | P26/AN18 |
| 1 | 0 | 0 | 1 | 1 | P27/AN19 |

### 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 ADCHSEL4 to ADCHSEL0 to a value between “10100” and “11111.”*
- *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 ADCHSEL4 (bit 7) of the high current driver control register (BUFCNT) to 0, 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.
   - When port 2 (AN12 to AN19) is to be used as analog input ports for an AD converter, set the corresponding bits of P2INH (FE67) to 1. The digital input is disabled, and the through current flowing in the digital input circuit can be prevented.

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, PWM1/AN8, PWM0/AN9, XT1/AN10, XT2/AN11, and P20/AN12 to P27/AN19. 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 USB Interface

## 3.20.1 Overview

This series of microcontrollers incorporates a USB (Universal Serial Bus) function control circuit that has the following features:

1) Conforms to USB Specification version 2.0.
2) Supports the-full speed (12 Mbps) specifications.
3) Supports control transfer, bulk transfer, interrupt transfer, and isochronous transfer modes.

## 3.20.2 Functions

1) USB function control
   - Up to 7 endpoints can be configured (including the control endpoint EP0).
   - The endpoint buffer for data transmission and reception (64 bytes maximum) is mapped into RAM.
   - The USB data sampling clock (48 MHz) is supplied from a clock that is generated by the internal PLL.

2) Interrupt generation

   An interrupt request is generated at the end of a USB transaction or on detection of a bus reset signal, a suspend state, an SOF packet, or a resume signal if the corresponding interrupt request enable bit is set.

3) It is necessary to manipulate the following special function registers to control the USB interface.
   - USBDIV, PLLCNT
   - USCTRL, USPORT, USBINT, EP0INT, EP1INT, EP2INT, EP3INT, EP4INT, EP5INT, EP6INT, FRAMEL, FRAMEH, USBADR, EPINFO, EP0STA, EP0MP, EP0RX, EP0TX, EP1STA, EP2STA, EP3STA, EP4STA, EP5STA, EP6STA, EP1CNT, EP1RX, EP2CNT, EP2RX, EP3CNT, EP3RX, EP4CNT, EP4RX, EP5CNT, EP5RX, EP6CNT, EP6RX, EPBMOD, TESTR0, TESTR1
   - P3, P3DDR

## USB

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE04 | 0000 0000 | R/W | USBDIV | CF48ON | DVCKON | DVCKDR | UREFSEL | CF12OFF | UDVSEL2 | UDVSEL1 | UDVSEL0 |
| FE0D | 0000 0000 | R/W | PLLCNT | SELREF2 | SELREF1 | SELREF0 | PLLTEST | VCOSTP | CMPSTP | PLLCNT1 | PLLCNT0 |
| FE80 | 0000 0000 | R/W | USCTRL | USBON | USBRUN | VD3OEN | VD3KIL | IDLFG | IDLEN | DPIEZ | DMIEZ |
| FE81 | 0000 0000 | R/W | USPORT | DDRSOF | VD3OE2 | USBSIO | SUSPND | DDRDP | DDRDM | PORTDP | PORTDM |
| FE82 | 0000 0000 | R/W | USBINT | BRSFG | BRSEN | BACFG | BACEN | SOFFG | SOFEN | USBINT1 | ENPEN |
| FE83 | 0000 0000 | R/W | EP0INT | AK0FG | AK0EN | NK0FG | NK0EN | ER0FG | ER0EN | ST0FG | ST0EN |
| FE84 | 0000 0000 | R/W | EP1INT | AK1FG | AK1EN | NK1FG | NK1EN | ER1FG | ER1EN | ST1FG | ST1EN |
| FE85 | 0000 0000 | R/W | EP2INT | AK2FG | AK2EN | NK2FG | NK2EN | ER2FG | ER2EN | ST2FG | ST2EN |
| FE86 | 0000 0000 | R/W | EP3INT | AK3FG | AK3EN | NK3FG | NK3EN | ER3FG | ER3EN | ST3FG | ST3EN |
| FE87 | 0000 0000 | R/W | EP4INT | AK4FG | AK4EN | NK4FG | NK4EN | ER4FG | ER4EN | ST4FG | ST4EN |
| FE88 | 0000 0000 | R/W | EP5INT | AK5FG | AK5EN | NK5FG | NK5EN | ER5FG | ER5EN | ST5FG | ST5EN |
| FE89 | 0000 0000 | R/W | EP6INT | AK6FG | AK6EN | NK6FG | NK6EN | ER6FG | ER6EN | ST6FG | ST6EN |
| FE8A | 0000 0000 | R | FRAMEL | FRM07 | FRM06 | FRM05 | FRM04 | FRM03 | FRM02 | FRM01 | FRM00 |
| FE8B | HHHH H000 | R | FRAMEH | - | - | - | - | - | FRM10 | FRM09 | FRM08 |
| FE8C | 0000 0000 | R/W | USBADR | ADREN | ADDR6 | ADDR5 | ADDR4 | ADDR3 | ADDR2 | ADDR1 | ADDR0 |
| FE8D | 0000 0000 | R/W | EPINFO | EPNO3 | EPNO2 | EPNO1 | EPNO0 | TKN1 | TKN0 | CTKN1 | CTKN0 |
| FE8E | 0000 0000 | R/W | EP0STA | E0EN | E0TGL | E0OVR | E0STL | E0ACK | E0CSU | E0CST | E0CRW |
| FE8F | H000 0000 | R/W | EP0MP | - | E0MP6 | E0MP5 | E0MP4 | E0MP3 | E0MP2 | E0MP1 | E0MP0 |
| FE90 | H000 0000 | R/W | EP0RX | - | E0RX6 | E0RX5 | E0RX4 | E0RX3 | E0RX2 | E0RX1 | E0RX0 |
| FE91 | H000 0000 | R/W | EP0TX | - | E0TX6 | E0TX5 | E0TX4 | E0TX3 | E0TX2 | E0TX1 | E0TX0 |
| FE92 | 0000 0000 | R/W | EP1STA | E1EN | E1TGL | E1OVR | E1STL | E1ACK | E1DIR | E1ISO | E1BNK |
| FE93 | 0000 0000 | R/W | EP2STA | E2EN | E2TGL | E2OVR | E2STL | E2ACK | E2DIR | E2ISO | E2BNK |
| FE94 | 0000 0000 | R/W | EP3STA | E3EN | E3TGL | E3OVR | E3STL | E3ACK | E3DIR | E3ISO | E3BNK |
| FE95 | 0000 0000 | R/W | EP4STA | E4EN | E4TGL | E4OVR | E4STL | E4ACK | E4DIR | E4ISO | E4BNK |
| FE96 | 0000 0000 | R/W | EP5STA | E5EN | E5TGL | E5OVR | E5STL | E5ACK | E5DIR | E5ISO | E5BNK |
| FE97 | 0000 0000 | R/W | EP6STA | E6EN | E6TGL | E6OVR | E6STL | E6ACK | E6DIR | E6ISO | E6BNK |
| FE98 | H000 0000 | R/W | EP1CNT | - | E1CN6 | E1CN5 | E1CN4 | E1CN3 | E1CN2 | E1CN1 | E1CN0 |
| FE99 | H000 0000 | R/W | EP1RX | - | E1RX6 | E1RX5 | E1RX4 | E1RX3 | E1RX2 | E1RX1 | E1RX0 |
| FE9A | H000 0000 | R/W | EP2CNT | - | E2CN6 | E2CN5 | E2CN4 | E2CN3 | E2CN2 | E2CN1 | E2CN0 |
| FE9B | H000 0000 | R/W | EP2RX | - | E2RX6 | E2RX5 | E2RX4 | E2RX3 | E2RX2 | E2RX1 | E2RX0 |
| FE9C | H000 0000 | R/W | EP3CNT | - | E3CN6 | E3CN5 | E3CN4 | E3CN3 | E3CN2 | E3CN1 | E3CN0 |
| FE9D | H000 0000 | R/W | EP3RX | - | E3RX6 | E3RX5 | E3RX4 | E3RX3 | E3RX2 | E3RX1 | E3RX0 |
| FE9E | H000 0000 | R/W | EP4CNT | - | E4CN6 | E4CN5 | E4CN4 | E4CN3 | E4CN2 | E4CN1 | E4CN0 |
| FE9F | H000 0000 | R/W | EP4RX | - | E4RX6 | E4RX5 | E4RX4 | E4RX3 | E4RX2 | E4RX1 | E4RX0 |
| FEA0 | H000 0000 | R/W | EP5CNT | - | E5CN6 | E5CN5 | E5CN4 | E5CN3 | E5CN2 | E5CN1 | E5CN0 |
| FEA2 | H000 0000 | R/W | EP5RX | - | E5RX6 | E5RX5 | E5RX4 | E5RX3 | E5RX2 | E5RX1 | E5RX0 |
| FEA6 | H000 0000 | R/W | EP6CNT | - | E6CN6 | E6CN5 | E6CN4 | E6CN3 | E6CN2 | E6CN1 | E6CN0 |
| FEA8 | H000 0000 | R/W | EP6RX | - | E6RX6 | E6RX5 | E6RX4 | E6RX3 | E6RX2 | E6RX1 | E6RX0 |
| FEAB | HHHH H000 | R/W | EPBMOD | - | - | - | - | - | EPBM2 | EPBM1 | EPBM0 |
| FEAC | 0000 0000 | R/W | TESTR0 | DPLTST | CMPTST | CMPKIL | TTXCLK | TTXREQ | TADR2 | TADR1 | P71NDL |
| FEAD | 0000 0000 | R/W | TESTR1 | TDAT7 | TDAT6 | TDAT5 | TDAT4 | TDAT3 | TDAT2 | TDAT1 | TDAT0 |

## 3.20.3 Circuit Configuration

The USB interface control circuit consists of the functional blocks shown below.

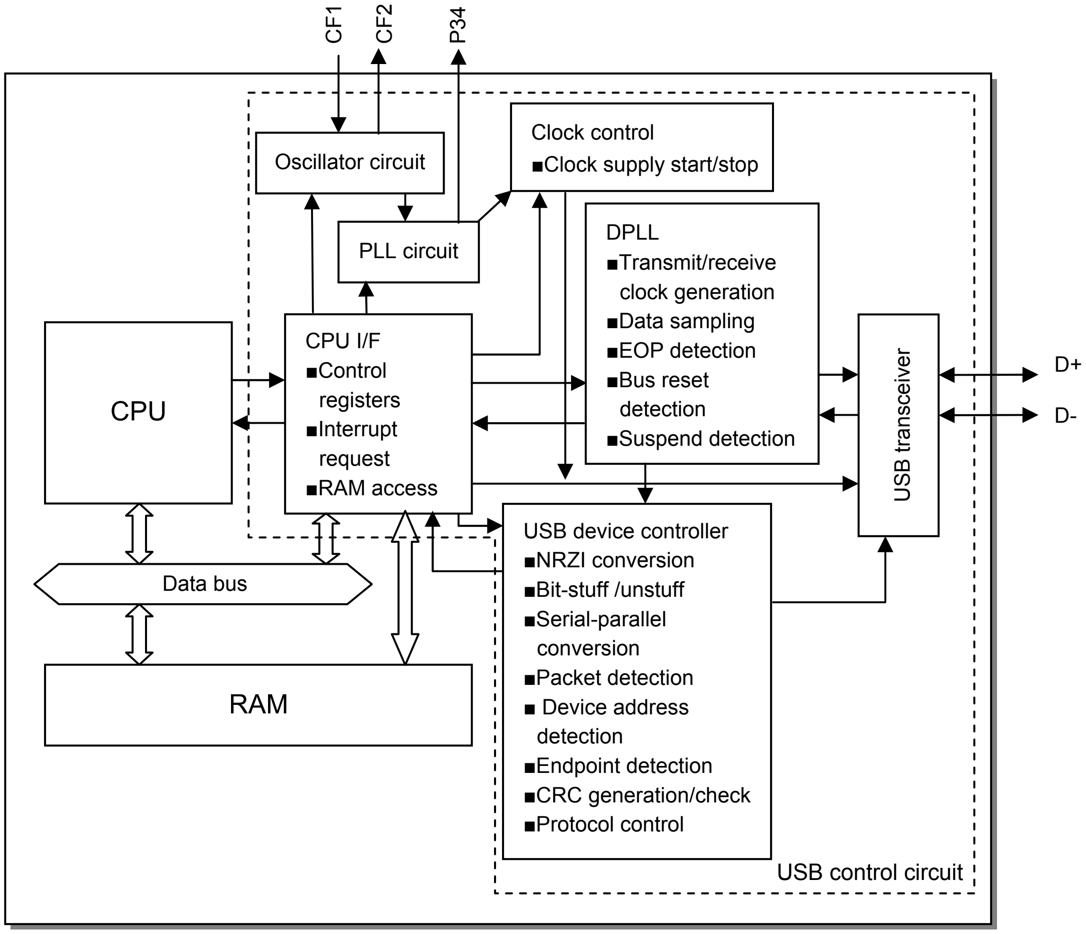


**Figure 3.20.1 USB Interface Control Circuit Block Diagram**

### 3.20.3.1 USB interface PLL circuit

The internal PLL generates the USB data sampling clock (48 MHz).

### 3.20.3.2 Clock control circuit

This circuit starts and stops supply of the clock signal to the DPLL and USB device controller.

### 3.20.3.3 DPLL circuit

1) Generates the transmit/receive clock (12 MHz).
2) Samples receive data.
3) Detects EOP.
4) Detects the bus reset signal.
5) Detects the suspend signal.

### 3.20.3.4 USB device controller circuit

1) Performs NRZI encoding/decoding.
2) Performs bit stuffing/unstuffing.
3) Performs serial-to-parallel conversion.
4) Detects packets.

5) Detects the device address and endpoint.

6) Generates and checks CRC.

7) Performs protocol control.

### 3.20.3.5 CPU interface circuit

1) Has registers for controlling the operation of the USB interface 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 (end point buffer).

### 3.20.3.6 Endpoint configuration

1) Up to 7 endpoints can be configured.

2) The address mapping in RAM of the data transmit/receive endpoint buffer can be selected by configuring the endpoint buffer mode register (EPBMOD).

3) The table below lists the transfer types that each endpoint supports, data buffer sizes, and the data buffer RAM address mapping (modes 0 to 7).

**Table 3.20.1 Endpoints and Supported Transfer Types**

| Endpoint | | EP0 | EP1 | EP2 | EP3 | EP4 | EP5 | EP6 |
|---|---|---|---|---|---|---|---|---|
| Transfer type | Control | ○ | – | – | – | – | – | – |
| | Bulk | – | ○ | ○ | ○ | ○ | ○ | ○ |
| | Interrupt | – | ○ | ○ | ○ | ○ | ○ | ○ |
| | Isochronous | – | ○ | ○ | ○ | ○ | ○ | ○ |

**Table 3.20.2 Endpoint Configuration (Mode 0)**

| | Setting | Max. Size (in bytes) | RAM Address |
|---|---|---|---|
| EP0 | Receive | 64 | 0200H－023FH |
| | Transmit | | 0240H－027FH |
| EP1 | Bank 0 | 64 | 0280H－02BFH |
| | Bank 1 | | 02C0H－02FFH |
| EP2 | Bank 0 | 64 | 0300H－033FH |
| | Bank 1 | | 0340H－037FH |
| EP3 | Bank 0 | 64 | 0380H－03BFH |
| | Bank 1 | | 03C0H－03FFH |
| EP4 | Bank 0 | 64 | 0380H－03BFH |
| | Bank 1 | | 03C0H－03FFH |
| EP5 | Bank 0 | 64 | 0300H－033FH |
| | Bank 1 | | 0340H－037FH |
| EP6 | Bank 0 | 64 | 0280H－02BFH |
| | Bank 1 | | 02C0H－02FFH |

* The RAM addresses of EP1 and EP6, EP2 and EP5, and EP3 and EP4 overlap with each other. respectively.

Table 3.20.3　Endpoint Configuration (Mode 1)

| | Setting | Max. Size (in bytes) | RAM Address |
|---|---|---|---|
| EP0 | Receive | 64 | 0380H－03BFH |
| | Transmit | | 03C0H－03FFH |
| EP1 | Bank 0 | 64 | 0300H－033FH |
| | Bank 1 | | 0340H－037FH |
| EP2 | Bank 0 | 64 | 0280H－02BFH |
| | Bank 1 | | 02C0H－02FFH |
| EP3 | Bank 0 | 16 | 0260H－026FH |
| | Bank 1 | | 0270H－027FH |
| EP4 | Bank 0 | 16 | 0240H－024FH |
| | Bank 1 | | 0250H－025FH |
| EP5 | Bank 0 | 16 | 0220H－022FH |
| | Bank 1 | | 0230H－023FH |
| EP6 | Bank 0 | 16 | 0200H－020FH |
| | Bank 1 | | 0210H－021FH |

Table 3.20.4　Endpoint Configuration (Mode 2)

| | Setting | Max. Size (in bytes) | RAM Address |
|---|---|---|---|
| EP0 | Receive | 64 | 0380H－03BFH |
| | Transmit | | 03C0H－03FFH |
| EP1 | Bank 0 | 32 | 0340H－035FH |
| | Bank 1 | | 0360H－037FH |
| EP2 | Bank 0 | 32 | 0300H－031FH |
| | Bank 1 | | 0320H－033FH |
| EP3 | Bank 0 | 32 | 02C0H－02DFH |
| | Bank 1 | | 02E0H－02FFH |
| EP4 | Bank 0 | 32 | 0280H－029FH |
| | Bank 1 | | 02A0H－02BFH |
| EP5 | Bank 0 | 32 | 0240H－025FH |
| | Bank 1 | | 0260H－027FH |
| EP6 | Bank 0 | 32 | 0200H－021FH |
| | Bank 1 | | 0220H－023FH |

Table 3.20.5　Endpoint Configuration (Mode 3)

| | Setting | Max. Size (in bytes) | RAM Address |
|---|---|---|---|
| EP0 | Receive | 32 | 03C0H－03DFH |
| | Transmit | | 03E0H－03FFH |
| EP1 | Bank 0 | 32 | 0380H－039FH |
| | Bank 1 | | 03A0H－03BFH |
| EP2 | Bank 0 | 32 | 0340H－035FH |
| | Bank 1 | | 0360H－037FH |
| EP3 | Bank 0 | 32 | 0300H－031FH |
| | Bank 1 | | 0320H－033FH |
| EP4 | Bank 0 | 32 | 02C0H－02DFH |
| | Bank 1 | | 02E0H－02FFH |
| EP5 | Bank 0 | 32 | 0280H－029FH |
| | Bank 1 | | 02A0H－02BFH |
| EP6 | Bank 0 | 32 | 0240H－025FH |
| | Bank 1 | | 0260H－027FH |

Table 3.20.6 Endpoint Configuration (Mode 4)

| | Setting | Max. Size (in bytes) | RAM Address |
|---|---|---|---|
| EP0 | Receive | 32 | 03C0H－03DFH |
| | Transmit | | 03E0H－03FFH |
| EP1 | Bank 0 | 32 | 0380H－039FH |
| | Bank 1 | | 03A0H－03BFH |
| EP2 | Bank 0 | 32 | 0340H－035FH |
| | Bank 1 | | 0360H－037FH |
| EP3 | Bank 0 | 32 | 0300H－031FH |
| | Bank 1 | | 0320H－033FH |
| EP4 | Bank 0 | 16 | 02E0H－02EFH |
| | Bank 1 | | 02F0H－02FFH |
| EP5 | Bank 0 | 16 | 02C0H－02CFH |
| | Bank 1 | | 02D0H－02DFH |
| EP6 | Bank 0 | 16 | 02A0H－02AFH |
| | Bank 1 | | 02B0H－02BFH |

Table 3.20.7 Endpoint Configuration (Mode 5)

| | Setting | Max. Size (in bytes) | RAM Address |
|---|---|---|---|
| EP0 | Receive | 32 | 03C0H－03DFH |
| | Transmit | | 03E0H－03FFH |
| EP1 | Bank 0 | 16 | 03A0H－03AFH |
| | Bank 1 | | 03B0H－03BFH |
| EP2 | Bank 0 | 16 | 0380H－038FH |
| | Bank 1 | | 0390H－039FH |
| EP3 | Bank 0 | 16 | 0360H－036FH |
| | Bank 1 | | 0370H－037FH |
| EP4 | Bank 0 | 16 | 0340H－034FH |
| | Bank 1 | | 0350H－035FH |
| EP5 | Bank 0 | 16 | 0320H－032FH |
| | Bank 1 | | 0330H－033FH |
| EP6 | Bank 0 | 16 | 0300H－030FH |
| | Bank 1 | | 0310H－031FH |

Table 3.20.8 Endpoint Configuration (Mode 6)

| | Setting | Max. Size (in bytes) | RAM Address |
|---|---|---|---|
| EP0 | Receive | 16 | 03E0H－03EFH |
| | Transmit | | 03F0H－03FFH |
| EP1 | Bank 0 | 16 | 03C0H－03CFH |
| | Bank 1 | | 03D0H－03DFH |
| EP2 | Bank 0 | 16 | 03A0H－03AFH |
| | Bank 1 | | 03B0H－03BFH |
| EP3 | Bank 0 | 16 | 0380H－038FH |
| | Bank 1 | | 0390H－039FH |
| EP4 | Bank 0 | 8 | 0370H－0377H |
| | Bank 1 | | 0378H－037FH |
| EP5 | Bank 0 | 8 | 0360H－0367H |
| | Bank 1 | | 0368H－036FH |
| EP6 | Bank 0 | 8 | 0350H－0357H |
| | Bank 1 | | 0358H－035FH |

Table 3.20.9 Endpoint Configuration (Mode 7)

| | Setting | Max. Size (in bytes) | RAM Address |
|---|---|---|---|
| EP0 | Receive | 8 | 03F0H－03F7H |
| | Transmit | | 03F8H－03FFH |
| EP1 | Bank 0 | 8 | 03E8H－03EFH |
| | Bank 1 | | 03B8H－03BFH |
| EP2 | Bank 0 | 8 | 03E0H－03E7H |
| | Bank 1 | | 03B0H－03B7H |
| EP3 | Bank 0 | 8 | 03D8H－03DFH |
| | Bank 1 | | 03A8H－03AFH |
| EP4 | Bank 0 | 8 | 03D0H－03D7H |
| | Bank 1 | | 03A0H－03A7H |
| EP5 | Bank 0 | 8 | 03C8H－03CFH |
| | Bank 1 | | 0398H－039FH |
| EP6 | Bank 0 | 8 | 03C0H－03C7H |
| | Bank 1 | | 0390H－0397H |

### Table 3.20.10 Endpoint Buffer RAM Mapping

| RAM Address | EPBMOD | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0x200 | EP0Rx | EP6BK0 | EP6BK0 | ▨ | ▨ | ▨ | ▨ | ▨ |
| 0x210 | | EP6BK1 | | ▨ | ▨ | ▨ | ▨ | ▨ |
| 0x220 | | EP5BK0 | EP6BK1 | ▨ | ▨ | ▨ | ▨ | ▨ |
| 0x230 | | EP5BK1 | | ▨ | ▨ | ▨ | ▨ | ▨ |
| 0x240 | EP0Tx | EP4BK0 | EP5BK0 | EP6BK0 | ▨ | ▨ | ▨ | ▨ |
| 0x250 | | EP4BK1 | | | ▨ | ▨ | ▨ | ▨ |
| 0x260 | | EP3BK0 | EP5BK1 | EP6BK1 | ▨ | ▨ | ▨ | ▨ |
| 0x270 | | EP3BK1 | | | ▨ | ▨ | ▨ | ▨ |
| 0x280 | EP1BK0<br>EP6BK0 | EP2BK0 | EP4BK0 | EP5BK0 | ▨ | ▨ | ▨ | ▨ |
| 0x290 | | | | | ▨ | ▨ | ▨ | ▨ |
| 0x2A0 | | | EP4BK1 | EP5BK1 | EP6BK0 | ▨ | ▨ | ▨ |
| 0x2B0 | | | | | EP6BK1 | ▨ | ▨ | ▨ |
| 0x2C0 | EP1BK1<br>EP6BK1 | EP2BK1 | EP3BK0 | EP4BK0 | EP5BK0 | ▨ | ▨ | ▨ |
| 0x2D0 | | | | | EP5BK1 | ▨ | ▨ | ▨ |
| 0x2E0 | | | EP3BK1 | EP4BK1 | EP4BK0 | ▨ | ▨ | ▨ |
| 0x2F0 | | | | | EP4BK1 | ▨ | ▨ | ▨ |
| 0x300 | EP2BK0<br>EP5BK0 | EP1BK0 | EP2BK0 | EP3BK0 | EP3BK0 | EP6BK0 | ▨ | ▨ |
| 0x310 | | | | | | EP6BK1 | ▨ | ▨ |
| 0x320 | | | EP2BK1 | EP3BK1 | EP3BK1 | EP5BK0 | ▨ | ▨ |
| 0x330 | | | | | | EP5BK1 | ▨ | ▨ |
| 0x340 | EP2BK1<br>EP5BK1 | EP1BK1 | EP1BK0 | EP2BK0 | EP2BK0 | EP4BK0 | ▨ | ▨ |
| 0x350 | | | | | | EP4BK1 | EP6BK0<br>EP6BK1 | ▨ |
| 0x360 | | | EP1BK1 | EP2BK1 | EP2BK1 | EP3BK0 | EP5BK0<br>EP5BK1 | ▨ |
| 0x370 | | | | | | EP3BK1 | EP4BK0<br>EP4BK1 | ▨ |
| 0x380 | EP3BK0<br>EP4BK0 | EP0Rx | EP0Rx | EP1BK0 | EP1BK0 | EP2BK0 | EP3BK0 | ▨ |
| 0x390 | | | | | | EP2BK1 | EP3BK1 | EP6BK1<br>EP5BK1 |
| 0x3A0 | | | | EP1BK1 | EP1BK1 | EP1BK0 | EP2BK0 | EP4BK1<br>EP3BK1 |
| 0x3B0 | | | | | | EP1BK1 | EP2BK1 | EP2BK1<br>EP1BK1 |
| 0x3C0 | EP3BK1<br>EP4BK1 | EP0Tx | EP0Tx | EP0Rx | EP0Rx | EP0Rx | EP1BK0 | EP6BK0<br>EP5BK0 |
| 0x3D0 | | | | | | | EP1BK1 | EP4BK0<br>EP3BK0 |
| 0x3E0 | | | | EP0Tx | EP0Tx | EP0Tx | EP0Rx | EP2BK0<br>EP1BK0 |
| 0x3F0 | | | | | | | EP0Tx | EP0Rx<br>EP0Tx |

### 3.20.3.7 Related I/O pins

1) The table below lists the I/O pins that are related to the USB interface control circuit.

**Table 3.20.11 USB Related I/O Pins**

| Pin Name | I/O | Description |
|---|---|---|
| D+ | I/O | USB serial data I/O pin |
| D- | I/O | USB serial data I/O pin |
| P34 | I/O | Internal PLL filter circuit connection pin |
| P70 | O | D+ pull-up control pin |
| P27 | O | D+ pull-up control pin |

2) The USB port peripheral circuit is shown in the figure below.

3) The pull-up resistor (1.5 kΩ) at the D+ pin must be connected to the P70 or P27 pin so that it can be turned on and off according to the presence or absence of Vbus.

The on/off control of this register must be accomplished through bit 5 (VD3OEN) of USCTRL (FE80H) or bit 6 (VD3OE2) of USPORT (FE81H).

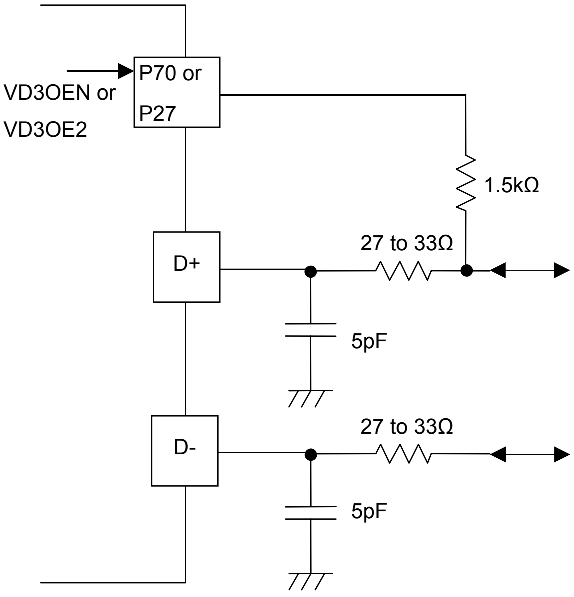


**Figure 3.20.2 USB Port Peripheral Circuit**

4) To generate the USB 48 MHz clock using the internal PLL circuit, it is necessary to connect the following external filter circuit to the P34/UFILT pin.

Set bit 4 (P34DDR) of P3DDR (FE4DH) and bit 4 (P34) of P3 (FE4CH) to 0.

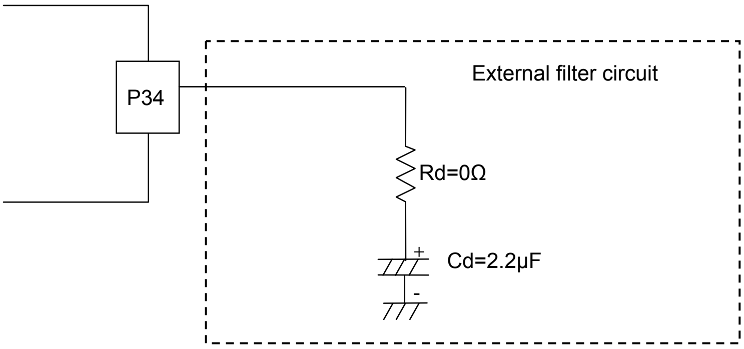


**Figure 3.20.3 Internal-PLL External Filter Circuit for the USB Interface**

## 3.20.4 Related Registers

### 3.20.4.1 USB frequency-divided clock control register (USBDIV)

1) This register is used to select the frequency of the frequency-divided clock that is derived from the 48 MHz clock for the USB.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE04 | 0000 0000 | R/W | USBDIV | CF48ON | DVCKON | DVCKDR | UREFSEL | CF12OFF | UDVSEL2 | UDVSEL1 | UDVSEL0 |

**CF48ON (bit 7): Reserved bit.**

This bit must always be set to 0.

**DVCKON (bit 6): Frequency-divided clock select flag**

<1> This bit must be set to 1 to select the frequency-divided clock that is generated by frequency-dividing the USB 48 MHz clock for the main clock. If the OCR register is set to select the main clock (CLKCB4 set to 1) in this configuration, the frequency-divided clock is supplied as the system clock.

<2> Set this bit to 0 to select the CF clock as the main clock.

<3> When changing the setting of this bit, make sure that the system clock selection of the OCR register is set to a clock other than the main clock (CLKCB4 = 0).

**DVCKDR (bit 5): Frequency-divided clock external output control flag**

<1> This bit must be set to 1 to transmit the frequency-divided clock that is generated by frequency-dividing the USB 48 MHz clock 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 frequency-divided clock is transmitted from P73.

<2> Set this bit to 0 if the frequency-divided clock is not transmitted to any external device.

**UREFSEL (bit 4): PLL reference clock select**

<1> When this bit is set to 0, the clock that is obtained by multiplying the frequency-divided (by 8 or 12) CF main clock (8 or 12 MHz) by 48 is supplied as the USB 48 MHz clock.

<2> When this bit is set to 1, the clock that is obtained by multiplying the CF main clock (8 MHz or 12 MHz) by 6 or 4 is supplied as the USB 48 MHz clock.

**CF12OFF (bit 3): Reserved bit.**

This bit must always be set to 0.

**UDVSEL2 (bit 2): Frequency-divided clock frequency select**

**UDVSEL1 (bit 1): Frequency-divided clock frequency select**

**UDVSEL0 (bit 0): Frequency-divided clock frequency select**

<1> These bits select the divided clock frequency of the 48 MHz clock for the USB.

<2> Set the frequency-divided clock to 8 to 12 MHz to run the USB interface control circuit by supplying the frequency-divided clock as the system clock.

<3> To change the setting of the 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 3.20.12 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) |

### 3.20.4.2 USB PLL control register (PLLCNT)

1) This register is an 8-bit register that controls the operation of the PLL oscillator circuit for the USB.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE0D | 0000 0000 | R/W | PLLCNT | SELREF2 | SELREF1 | SELREF0 | PLLTEST | VCOSTP | CMPSTP | PLLCNT1 | PLLCNT0 |

**SELREF2 (bit 7): PLL reference clock frequency select**

**SELREF1 (bit 6): PLL reference clock frequency select**

**SELREF0 (bit 5): PLL reference clock frequency select**

<1> These bits select the oscillation frequency of the main clock connected to the CF pins (CF1 and CF2).

<2> Either 8 or 12 MHz can be selected if "ENABLE" is selected as the user option "Main clock 8 MHz selection." If "DISABLE" is selected, only 12 MHz is selected.

Table 3.20.13 Main Clock Oscillation Frequencies

| Main Clock 8 MHz 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 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 the 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 3.20.3 to the P34/UFILT pin.

<2> Set these bits to 1 when the PLL is not to be used.

**PLLCNT1 (bit 1): Reserved bit.**

**PLLCNT0 (bit 0): Reserved bit.**

### 3.20.4.3 USB operation control register (USCTRL)

1) This register is used to control clock supply start/stop.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE80 | 0000 0000 | R/W | USCTRL | USBON | USBRUN | VD3OEN | VD3KIL | IDLFG | IDLEN | DPIEZ | DMIEZ |

#### USBON (bit 7): USB clock control flag

#### USBRUN (bit 6): USB clock control flag

<1> These bits control clock supply start/stop.

**Table 3.20.14 Clock Control**

| USBON | USBRUN | USB operation |
|---|---|---|
| 0 | — | Enable to detect only USB bus active (*1).<br>Disable the other USB functions. |
| 1 | 0 | Enable to detect only USB bus reset and USB bus active.<br>Disable the other USB functions. |
| 1 | 1 | Enable the entire USB block. |

*1: Bus state other than bus idle (J state), i.e., K state or SE0

#### VD3OEN (bit 5): D+ pull-up control flag

<1> Setting this bit to 1 causes a high D+ pull-up level to be output from the P70 pin. In this case, bit 4 (P70DDR) and bit 0 (P70) of P7 (FE5CH) must be set to 0.

<2> Setting this bit to 0 inhibits the high D+ pull-up level from being output from the P70 pin. In this case, the P70 pin is held in the Hi-Z state if bit 4 (P70DDR) and bit 0 (P70) of P7 (FE5CH) are set to 0.

#### VD3KIL (bit 4): Reserved bit.

This bit must always be set to 0.

#### IDLFG (bit 3): Suspend detection flag

<1> This flag is set if a suspend state (staying in bus idle state for 3 ms or longer) is detected.

<2> This bit remains set until the suspend state is released.

<3> This flag must be cleared with an instruction.

#### IDLEN (bit 2): Suspend interrupt request enable flag

<1> An interrupt request to vector address 0033H is generated when this bit and IDLFG are set to 1.

#### DPIEZ (bit 1): D+ pin input enable flag

<1> A 1 in this bit disables reading in of the data on the D+ pin.

<2> A 0 in this bit enables reading in of the data on the D+ pin.

<3> Set this bit to 0 to perform USB communication.

#### DMIEZ (bit 0): D- pin input enable flag

<1> A 1 in this bit disables reading in of the data on the D- pin.

<2> A 0 in this bit enables reading in of the data on the D- pin

<3> Set this bit to 0 to perform USB communication.

### 3.20.4.4 USB port control register (USPORT)

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE81 | 0000 0000 | R/W | USPORT | DDRSOF | VD3OE2 | USBSIO | SUSPND | DDRDP | DDRDM | PORTDP | PORTDM |

#### DDRSOF (bit 7): SOF detection pulse output control

<1> Setting DDRSOF to 1, bit 6 (P72DDR) of P7 (FE5CH) to 1 and bit 2 (P72) to 0 causes an SOF detection pulse (approx. 80 ns) to be output from the P72 pin on reception of an SOF

#### VD3OE2 (bit 6):D+ pull-up control flag

<1> Setting this bit to 1 causes a high D+ pull-up level to be output from the P27 pin. In this case, bit 7 (P27) of P2 (FE48H) and bit 7 (P27DDR) of P2DDR (FE49H) must be set to 0.

<2> Setting this bit to 0 inhibits a high D+ pull-up level from being output from the P27 pin. In this case, the P27 pin is held in the Hi-Z state if bit 7 (P27) of P2 (FE48H) and bit 7 (P27DDR) of P2DDR (FE49H) are set to 0.

#### USBSIO (bit 5): Reserved bit.

This bit must always be set to 0.

#### SUSPND (bit 4): USB transceiver operation control flag

<1> When this bit is set to 1, the USB transceiver is placed in the suspend state. It is possible to detect bus active even in this state.

<2> When this bit is set to 0, the USB transceiver is in the normal operating mode.

#### DDRDP (bit 3): D+ pin general-purpose output control

<1> When this bit is set to 1, PORTDP data is output from the D+ pin.

<2> This bit must be set to 0 when performing USB communication.

#### DDRDM (bit 2): D- pin general-purpose output control

<1> When this bit is set to 1, PORTDM data is output from the D- pin.

<2> This bit must be set to 0 when performing USB communication.

#### PORTDP (bit 1): D+ port data latch

<1> When DDRDP is set to 1, data in this bit is output from the D+ pin.

<2> When this bit is read with an instruction, the data on the D+ pin is read in. If USPORT 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 D+.

#### PORTDM (bit 0): D- port data latch

<1> When DDRDM is set to 1, data in this bit is output from the D- pin.

<2> When this bit is read with an instruction, the data on the D- pin is read in. If USPORT 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 D- pin.

### 3.20.4.5 USB interrupt control register (USBINT)

1) This register controls the USB interrupts.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE82 | 0000 0000 | R/W | USBINT | BRSFG | BRSEN | BACFG | BACEN | SOFFG | SOFEN | USBINT1 | ENPEN |

#### BRSFG (bit 7): Bus reset detection flag

<1> This flag is set when a USB bus reset state (SE0 state continuing for 2.5 μs or longer) is detected.

<2> The flag remains set until the bus reset state is released.

<3> This flag must be cleared with an instruction.

#### BRSEN (bit 6): Bus reset interrupt request enable flag

<1> An interrupt request to vector address 0033H is generated when this bit and BRSFG are set to 1.

#### BACFG (bit 5): Bus active detection flag

<1> This flag is set when a USB bus active state (bus state other than bus idle) is detected.

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

#### BACEN (bit 4): Bus active interrupt request enable flag

<1> An interrupt request to vector address 0013H is generated when this bit and BACFG are set to 1.

#### SOFFG (bit 3): SOF detection flag

<1> This flag is set when an SOF is detected.

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

#### SOFEN (bit 2): SOF interrupt request enable flag

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

#### USBINT1 (bit 1): Reserved bit.

This bit must always be set to 0.

#### ENPEN (bit 0): Endpoint interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit is set to 1 and either one of interrupt sources EP0INT, EP1INT, EP2INT, EP3INT, EP4INT, EP5INT, and EP6INT is established.

### 3.20.4.6 EP0 interrupt control register (EP0INT)

1) This register controls the endpoint 0 interrupts.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE83 | 0000 0000 | R/W | EP0INT | AK0FG | AK0EN | NK0FG | NK0EN | ER0FG | ER0EN | ST0FG | ST0EN |

#### AK0FG (bit 7): EP0 ACK end flag

<1> This flag is set to 1 when the endpoint 0 transaction terminates with an ACK. See the column entitled "End Flag" in Table 3.20.16.

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

#### AK0EN (bit 6): EP0 ACK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, AK0FG, and ENPEN are all set to 1.

### NK0FG (bit 5): EP0 NAK end flag

<1> This bit is set to 1 if the endpoint 0 transaction terminates with a NAK. See the column entitled "End Flag" in Table 3.20.16.

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

### NK0EN (bit 4): EP0 NAK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, NK0FG, and ENPEN are all set to 1.

### ER0FG (bit 3): EP0 error end flag

<1> This bit is set to 1 if the endpoint 0 transaction terminates with an error. See the column entitled "End Flag" in Table 3.20.16.

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

### ER0EN (bit 2): EP0 error interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ER0FG, and ENPEN are all set to 1.

### ST0FG (bit 1): EP0 STALL end flag

<1> This bit is set to 1 if the endpoint 0 transaction terminates with a STALL. See the column entitled "End Flag" in Table 3.20.16.

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

### ST0EN (bit 0): EP0 STALL interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ST0FG, and ENPEN are all set to 1.

## 3.20.4.7 EP1 interrupt control register (EP1INT)

1) This register controls the endpoint 1 interrupts.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE84 | 0000 0000 | R/W | EP1INT | AK1FG | AK1EN | NK1FG | NK1EN | ER1FG | ER1EN | ST1FG | ST1EN |

### AK1FG (bit 7): EP1 ACK end flag

<1> This bit is set to 1 when the endpoint 1 transaction terminates normally with an ACK. See the column entitled "End Flag" in Table 3.20.17.

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

### AK1EN (bit 6): EP1 ACK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, AK1FG, and ENPEN are all set to 1.

### NK1FG (bit 5): EP1 NAK end flag

<1> This bit is set to 1 if the endpoint 1 transaction terminates with a NAK. See the column entitled "End Flag" in Table 3.20.17.

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

### NK1EN (bit 4): EP1 NAK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, NK1FG, and ENPEN are all set to 1.

### ER1FG (bit 3): EP1 error end flag

<1> This bit is set to 1 if the endpoint 1 transaction terminates with an error. See the column entitled "End Flag" in Table 3.20.17.

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

### ER1EN (bit 2): EP1 error interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ER1FG, and ENPEN are all set to 1.

### ST1FG (bit 1): EP1 STALL end flag

<1> This bit is set to 1 if the endpoint 1 transaction terminates with a STALL. See the column entitled "End Flag" in Table 3.20.17.

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

### ST1EN (bit 0): EP1 STALL interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ST1FG, and ENPEN are all set to 1.

## 3.20.4.8 EP2 interrupt control register (EP2INT)

1) This register controls the endpoint 2 interrupts.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE85 | 0000 0000 | R/W | EP2INT | AK2FG | AK2EN | NK2FG | NK2EN | ER2FG | ER2EN | ST2FG | ST2EN |

### AK2FG (bit 7): EP2 ACK end flag

<1> This bit is set to 1 when the endpoint 2 transaction terminates normally with an ACK. See the column entitled "End Flag" in Table 3.20.17.

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

### AK2EN (bit 6): EP2 ACK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, AK2FG, and ENPEN are all set to 1.

### NK2FG (bit 5): EP2 NAK end flag

<1> This bit is set to 1 if the endpoint 2 transaction terminates with a NAK. See the column entitled "End Flag" in Table 3.20.17.

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

### NK2EN (bit 4): EP2 NAK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, NK2FG, and ENPEN are all set to 1.

### ER2FG (bit 3): EP2 error end flag

<1> This bit is set to 1 if the endpoint 2 transaction terminates with an error. See the column entitled "End Flag" in Table 3.20.17.

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

### ER2EN (bit 2): EP2 error interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ER2FG, and ENPEN are all set to 1.

### ST2FG (bit 1): EP2 STALL end flag

<1> This bit is set to 1 if the endpoint 2 transaction terminates with a STALL. See the column entitled "End Flag" in Table 3.20.17.

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

### ST2EN (bit 0): EP2 STALL interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ST2FG, and ENPEN are all set to 1.

### 3.20.4.9 EP3 interrupt control register (EP3INT)

1) This register controls the endpoint 3 interrupts.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE86 | 0000 0000 | R/W | EP3INT | AK3FG | AK3EN | NK3FG | NK3EN | ER3FG | ER3EN | ST3FG | ST3EN |

### AK3FG (bit 7): EP3 ACK end flag

<1> This bit is set to 1 when the endpoint 3 transaction terminates normally with an ACK. See the column entitled "End Flag" in Table 3.20.17.

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

### AK3EN (bit 6): EP3 ACK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, AK3FG, and ENPEN are all set to 1.

### NK3FG (bit 5): EP3 NAK end flag

<1> This bit is set to 1 if the endpoint 3 transaction terminates with a NAK. See the column entitled "End Flag" in Table 3.20.17.

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

### NK3EN (bit 4): EP3 NAK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, NK3FG, and ENPEN are all set to 1.

### ER3FG (bit 3): EP3 error end flag

<1> This bit is set to 1 if the endpoint 3 transaction terminates with an error. See the column entitled "End Flag" in Table 3.20.17.

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

### ER3EN (bit 2): EP3 error interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ER3FG, and ENPEN are all set to 1.

### ST3FG (bit 1): EP3 STALL end flag

<1> This bit is set to 1 if the endpoint 3 transaction terminates with a STALL. See the column entitled "End Flag" in Table 3.20.17.

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

### ST3EN (bit 0): EP3 STALL interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ST3FG, and ENPEN are all set to 1.

### 3.20.4.10 EP4 interrupt control register (EP4INT)

1) This register controls the endpoint 4 interrupts.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE87 | 0000 0000 | R/W | EP4INT | AK4FG | AK4EN | NK4FG | NK4EN | ER4FG | ER4EN | ST4FG | ST4EN |

#### AK4FG (bit 7): EP4 ACK end flag

<1> This bit is set to 1 when the endpoint 4 transaction terminates normally with an ACK. See the column entitled "End Flag" in Table 3.20.17.

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

#### AK4EN (bit 6): EP4 ACK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, AK4FG, and ENPEN are all set to 1.

#### NK4FG (bit 5): EP4 NAK end flag

<1> This bit is set to 1 if the endpoint 4 transaction terminates with a NAK. See the column entitled "End Flag" in Table 3.20.17.

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

#### NK4EN (bit 4): EP4 NAK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, NK4FG, and ENPEN are all set to 1.

#### ER4FG (bit 3): EP4 error end flag

<1> This bit is set to 1 if the endpoint 4 transaction terminates with an error. See the column entitled "End Flag" in Table 3.20.17.

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

#### ER4EN (bit 2): EP4 error interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ER4FG, and ENPEN are all set to 1.

#### ST4FG (bit 1): EP4 STALL end flag

<1> This bit is set to 1 if the endpoint 4 transaction terminates with a STALL. See the column entitled "End Flag" in Table 3.20.17.

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

#### ST4EN (bit 0): EP4 STALL interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ST4FG, and ENPEN are all set to 1.

### 3.20.4.11 EP5 interrupt control register (EP5INT)

1) This register controls the endpoint 5 interrupts.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE88 | 0000 0000 | R/W | EP5INT | AK5FG | AK5EN | NK5FG | NK5EN | ER5FG | ER5EN | ST5FG | ST5EN |

#### AK5FG (bit 7): EP5 ACK end flag

<1> This bit is set to 1 when the endpoint 5 transaction terminates normally with an ACK. See the column entitled "End Flag" in Table 3.20.17.

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

#### AK5EN (bit 6): EP5 ACK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, AK5FG, and ENPEN are all set to 1.

#### NK5FG (bit 5): EP5 NAK end flag

<1> This bit is set to 1 if the endpoint 5 transaction terminates with a NAK. See the column entitled "End Flag" in Table 3.20.17.

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

#### NK5EN (bit 4): EP5 NAK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, NK5FG, and ENPEN are all set to 1.

#### ER5FG (bit 3): EP5 error end flag

<1> This bit is set to 1 if the endpoint 5 transaction terminates with an error. See the column entitled "End Flag" in Table 3.20.17.

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

#### ER5EN (bit 2): EP5 error interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ER5FG, and ENPEN are all set to 1.

#### ST5FG (bit 1): EP5 STALL end flag

<1> This bit is set to 1 if the endpoint 5 transaction terminates with a STALL. See the column entitled "End Flag" in Table 3.20.17.

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

#### ST5EN (bit 0): EP5 STALL interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ST5FG, and ENPEN are all set to 1.

### 3.20.4.12 EP6 interrupt control register (EP6INT)

1) This register controls the endpoint 6 interrupts.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE89 | 0000 0000 | R/W | EP6INT | AK6FG | AK6EN | NK6FG | NK6EN | ER6FG | ER6EN | ST6FG | ST6EN |

#### AK6FG (bit 7): EP6 ACK end flag

<1> This bit is set to 1 when the endpoint 6 transaction terminates normally with an ACK. See the column entitled "End Flag" in Table 3.20.17.

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

### AK6EN (bit 6): EP6 ACK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, AK6FG, and ENPEN are all set to 1.

### NK6FG (bit 5): EP6 NAK end flag

<1> This bit is set to 1 if the endpoint 6 transaction terminates with a NAK. See the column entitled "End Flag" in Table 3.20.17.

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

### NK6EN (bit 4): EP6 NAK interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, NK6FG, and ENPEN are all set to 1.

### ER6FG (bit 3): EP6 error end flag

<1> This bit is set to 1 if the endpoint 6 transaction terminates with an error. See the column entitled "End Flag" in Table 3.20.17.

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

### ER6EN (bit 2): EP6 error interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ER6FG, and ENPEN are all set to 1.

### ST6FG (bit 1): EP6 STALL end flag

<1> This bit is set to 1 if the endpoint 6 transaction terminates with a STALL. See the column entitled "End Flag" in Table 3.20.17.

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

### ST6EN (bit 0): EP6 STALL interrupt request enable flag

<1> An interrupt request to vector address 003BH is generated when this bit, ST6FG, and ENPEN are all set to 1.

## 3.20.4.13 Frame number register (FRAMEL, FRAMEH)

1) This register is loaded with the frame number when SOF data is received.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE8A | 0000 0000 | R | FRAMEL | FRM07 | FRM06 | FRM05 | FRM04 | FRM03 | FRM02 | FRM01 | FRM00 |
| FE8B | HHHH H000 | R | FRAMEH | - | - | - | - | - | FRM10 | FRM09 | FRM08 |

## 3.20.4.14 USB address register (USBADR)

1) This register sets the USB device address.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE8C | 0000 0000 | R/W | USBADR | ADREN | ADDR6 | ADDR5 | ADDR4 | ADDR3 | ADDR2 | ADDR1 | ADDR0 |

**ADREN (bit 7): Device address enable flag**

<1> A 1 in this bit enables ADDR[6:0]. Of the tokens sent from the host, only the tokens whose device address matches the contents of ADDR[6: 0] are processed.

<2> A 0 in this bit disables ADDR[6:0]. Of the tokens sent from the host, only the tokens whose device address is 0 are processed.

**ADDR6 (bit 6): Device address**

**ADDR5 (bit 5): Device address**

**ADDR4 (bit 4): Device address**

**ADDR3 (bit 3): Device address**

**ADDR2 (bit 2): Device address**

**ADDR1 (bit 1): Device address**

**ADDR0 (bit 0): Device address**

<1> These bits designate the device address assigned by the host.

### 3.20.4.15 Endpoint information register (EPINFO)

1) Information about the endpoint number and token type can be read out through this register.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE8D | 0000 0000 | R | EPINFO | EPNO3 | EPNO2 | EPNO1 | EPNO0 | TKN1 | TKN0 | CTKN1 | CTKN0 |

**EPNO3 (bit 7): Endpoint number**

**EPNO2 (bit 6): Endpoint number**

**EPNO1 (bit 5): Endpoint number**

**EPNO0 (bit 4): Endpoint number**

<1> These register bit positions are loaded with the endpoint number from the token packet when one of the following conditions is established:

(1) The transaction terminates normally with an ACK.

(2) NKnEN (n = 0 to 6) is set to 1 and the transaction for endpoint n terminates with a NAK.

(3) ERnEN (n = 0 to 6) is set to 1 and the transaction for endpoint n terminates with an error.

(4) STnEN (n = 0 to 6) is set to 1 and the transaction for endpoint n terminates with a STALL.

**TKN1 (bit 3): Token ID**

**TKN0 (bit 2): Token ID**

<1> These register bit positions are loaded with the token ID when either of the above conditions (1) to (4) is established.

**CTKN1 (bit 1): EP0 token ID**

**CTKN0 (bit 0): EP0 token ID**

<1> These register bit positions are loaded with the token ID when either of the above conditions (1) to (4) is established for an endpoint 0 transaction.

**Table 3.20.15 Token ID**

| Token | Token ID |
|---|---|
| OUT | 00 |
| IN | 10 |
| SETUP | 11 |

### 3.20.4.16 EP0 status register (EP0STA)

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE8E | 0000 0000 | R/W | EP0STA | E0EN | E0TGL | E0OVR | E0STL | E0ACK | E0CSU | E0CST | E0CRW |

**E0EN (bit 7): EP0 enable flag**

<1> When this bit is set to 1, the token for endpoint 0 is processed.

<2> When this bit is set to 0, the token for endpoint 0 is not processed.

**E0TGL (bit 6): EP0 data toggle**

<1> This bit is automatically set to 1 when a SETUP transaction terminates normally with an ACK.

<2> In an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E0TGL. The state of E0TGL is automatically inverted when the transaction terminates normally with an ACK.

<3> In an IN transaction, the data packet whose packet ID matches E0TGL is transmitted. The state of E0TGL is automatically inverted when an ACK from the host is received.

<4> This bit is automatically cleared to 0 when an OUT transaction or IN transaction in the status stage terminates normally with an ACK.

**E0OVR (bit 5): EP0 payload over flag**

<1> This bit is automatically set to 1 when a volume of data exceeding the maximum payload size specified in EP0MP is received.

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

**E0STL (bit 4): EP0 STALL flag**

<1> When this bit is set to 1, a STALL handshake is returned for IN and OUT transactions.

<2> This bit is automatically set to 1 when a STALL handshake is returned due to a protocol violation.

<3> This bit is automatically cleared to 0 when a SETUP transaction terminates normally with an ACK.

**E0ACK (bit 3): EP0 ACK flag**

<1> When this bit is set to 1, a data packet is transmitted for an IN transaction and an ACK handshake is returned for an OUT transaction.

<2> When this bit is set to 0, a NAK handshake is returned for IN and OUT transactions.

<3> See Table 3.20.16.

**E0CSU (bit 2): EP0 setup stage complete flag**

<1> This bit is automatically set to 1 when a SETUP transaction terminates normally with an ACK.

<2> This bit is automatically cleared to 0 when an IN transaction in the status stage terminates normally with an ACK.

<3> See Table 3.20.16.

**E0CST (bit 1): EP0 status stage flag**

<1> A 1 in this bit indicates that the USB interface is in the control transfer status stage.

<2> This bit is automatically cleared to 0 when a SETUP transaction terminates normally with an ACK.

<3> This bit is automatically set to 1 when an OUT transaction in the status stage terminates normally with an ACK.

<4> See Table 3.20.16.

### E0CRW (bit 0): EP0 transfer direction flag

<1> A 1 in this bit identifies a control read transfer.

<2> A 0 in this bit identifies a control write transfer.

<3> This bit is automatically cleared to 0 when a SETUP transaction terminates normally with an ACK.

<4> See Table 3.20.16.

### 3.20.4.17 EP0 maximum payload register (EP0MP)

1) This register defines the maximum payload size of endpoint 0.

2) The setting value is either 08[H], 10[H], 20[H], or 40[H].

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE8F | H000 0000 | R/W | EP0MP | - | E0MP6 | E0MP5 | E0MP4 | E0MP3 | E0MP2 | E0MP1 | E0MP0 |

### 3.20.4.18 EP0 receive data count register (EP0RX)

1) The number of data bytes received at endpoint 0 can be read out through this register.

2) The contents of this register are updated when a SETUP or OUT transaction for endpoint 0 terminates normally with an ACK.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE90 | H000 0000 | R/W | EP0RX | - | E0RX6 | E0RX5 | E0RX4 | E0RX3 | E0RX2 | E0RX1 | E0RX0 |

### 3.20.4.19 EP0 transmit data count register (EP0TX)

1) This register is loaded with the number of data bytes to be transmitted through endpoint 0.

2) The legitimate value range is from 00[H] to 40[H].

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE91 | H000 0000 | R/W | EP0TX | - | E0TX6 | E0TX5 | E0TX4 | E0TX3 | E0TX2 | E0TX1 | E0TX0 |

### 3.20.4.20 EP1 status register (EP1STA)

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE92 | 0000 0000 | R/W | EP1STA | E1EN | E1TGL | E1OVR | E1STL | E1ACK | E1DIR | E1ISO | E1BNK |

### E1EN (bit 7): EP1 enable flag

<1> When this bit is set to 1, the token for endpoint 1 is processed.

<2> When this bit is set to 0, the token for endpoint 1 is not processed.

### E1TGL (bit 6): EP1 data toggle

<1> In an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E1TGL. The state of E1TGL is automatically inverted when the transaction terminates normally with an ACK. The E1TGL state is not inverted, however, for isochronous transfers.

<2> In an IN transaction, the data packet whose packet ID matches E1TGL is transmitted. The state of E1TGL is automatically inverted when an ACK from the host is received.

### E1OVR (bit 5): EP1 payload over flag

<1> This bit is automatically set to 1 when a volume of data exceeding the maximum payload size specified in EP1CNT is received.

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

### E1STL (bit 4): EP1 STALL flag

<1> A STALL handshake is returned for IN and OUT transactions.

### E1ACK (bit 3): EP1 ACK flag

<1> When this bit is set to 1, a data packet is transmitted for an IN transaction and an ACK handshake is returned for an OUT transaction.

<2> When this bit is set to 0, a NAK handshake is returned for IN and OUT transactions.

<3> See Table 3.20.17.

### E1DIR (bit 2): EP1 transfer direction flag

<1> When this bit is set to 1, only the IN token for endpoint 1 is processed.

<2> When this bit is set to 0, only the OUT token for endpoint 1 is processed.

<3> See Table 3.20.17.

### E1ISO (bit 1): EP1 isochronous transfer flag

<1> When this bit is set to 1, the transfer type of endpoint 1 is set to isochronous transfer.

<2> When this bit is set to 0, the transfer type of endpoint 1 is set to either bulk transfer or interrupt transfer.

<3> See Table 3.20.17.

### E1BNK (bit 0): EP1 transfer RAM address control flag

<1> The values of this bit and the EPBMOD register (FEABH) switch the location of the data transmit/receive buffer area for endpoint 1.

<2> See Subsection 3.20.3.6, “Endpoint configuration.”

## 3.20.4.21 EP2 status register (EP2STA)

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE93 | 0000 0000 | R/W | EP2STA | E2EN | E2TGL | E2OVR | E2STL | E2ACK | E2DIR | E2ISO | E2BNK |

### E2EN (bit 7): EP2 enable flag

<1> When this bit is set to 1, the token for endpoint 2 is processed.

<2> When this bit is set to 0, the token for endpoint 2 is not processed.

### E2TGL (bit 6): EP2 data toggle

<1> In an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E2TGL. The state of E2TGL is automatically inverted when the transaction terminates normally with an ACK. The E2TGL state is not inverted, however, for isochronous transfers.

<2> In an IN transaction, the data packet whose packet ID matches E2TGL is transmitted. The state of E2TGL is automatically inverted when an ACK from the host is received.

#### E2OVR (bit 5): EP2 payload over flag

<1> This bit is automatically set to 1 when a volume of data exceeding the maximum payload size specified in EP2CNT is received.

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

#### E2STL (bit 4): EP2 STALL flag

<1> A STALL handshake is returned for IN and OUT transactions.

#### E2ACK (bit 3): EP2 ACK flag

<1> When this bit is set to 1, a data packet is transmitted for an IN transaction and an ACK handshake is returned for an OUT transaction.

<2> When this bit is set to 0, a NAK handshake is returned for IN and OUT transactions.

<3> See Table 3.20.17.

#### E2DIR (bit 2): EP2 transfer direction flag

<1> When this bit is set to 1, only the IN token for endpoint 2 is processed.

<2> When this bit is set to 0, only the OUT token for endpoint 2 is processed.

<3> See Table 3.20.17.

#### E2ISO (bit 1): EP2 isochronous transfer flag

<1> When this bit is set to 1, the transfer type of endpoint 2 is set to isochronous transfer.

<2> When this bit is set to 0, the transfer type of endpoint 2 is set to either bulk transfer or interrupt transfer.

<3> See Table 3.20.17.

#### E2BNK (bit 0): EP2 transfer RAM address control flag

<1> The values of this bit and the EPBMOD register (FEABH) switch the location of the data transmit/receive buffer area for endpoint 2.

<2> See Subsection 3.20.3.6, “Endpoint configuration.”

### 3.20.4.22 EP3 status register (EP3STA)

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE94 | 0000 0000 | R/W | EP3STA | E3EN | E3TGL | E3OVR | E3STL | E3ACK | E3DIR | E3ISO | E3BNK |

#### E3EN (bit 7): EP3 enable flag

<1> When this bit is set to 1, the token for endpoint 3 is processed.

<2> When this bit is set to 0, the token for endpoint 3 is not processed.

#### E3TGL (bit 6): EP3 data toggle

<1> In an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E3TGL. The state of E3TGL is automatically inverted when the transaction terminates normally with an ACK. The E3TGL state is not inverted, however, for isochronous transfers.

<2> In an IN transaction, the data packet whose packet ID matches E3TGL is transmitted. The state of E3TGL is automatically inverted when an ACK from the host is received.

### E3OVR (bit 5): EP3 payload over flag

<1> This bit is automatically set to 1 when a volume of data exceeding the maximum payload size specified in EP3CNT is received.

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

### E3STL (bit 4): EP3 STALL flag

<1> A STALL handshake is returned for IN and OUT transactions.

### E3ACK (bit 3): EP3 ACK flag

<1> When this bit is set to 1, a data packet is transmitted for an IN transaction and an ACK handshake is returned for an OUT transaction.

<2> When this bit is set to 0, a NAK handshake is returned for IN and OUT transactions.

<3> See Table 3.20.17.

### E3DIR (bit 2): EP3 transfer direction flag

<1> When this bit is set to 1, only the IN token for endpoint 3 is processed.

<2> When this bit is set to 0, only the OUT token for endpoint 3 is processed.

<3> See Table 3.20.17.

### E3ISO (bit 1): EP3 isochronous transfer flag

<1> When this bit is set to 1, the transfer type of endpoint 3 is set to isochronous transfer.

<2> When this bit is set to 0, the transfer type of endpoint 3 is set to either bulk transfer or interrupt transfer.

<3> See Table 3.20.17.

### E3BNK (bit 0): EP3 transfer RAM address control flag

<1> The values of this bit and the EPBMOD register (FEABH) switch the location of the data transmit/receive buffer area for endpoint 3.

<2> See Subsection 3.20.3.6, “Endpoint configuration.”

### 3.20.4.23 EP4 status register (EP4STA)

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE95 | 0000 0000 | R/W | EP4STA | E4EN | E4TGL | E4OVR | E4STL | E4ACK | E4DIR | E4ISO | E4BNK |

#### E4EN (bit 7): EP4 enable flag

<1> When this bit is set to 1, the token for endpoint 4 is processed .

<2> When this bit is set to 0, the token for endpoint 4 is not processed.

#### E4TGL (bit 6): EP4 data toggle

<1> In an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E4TGL. The state of E4TGL is automatically inverted when the transaction terminates normally with an ACK. The E4TGL state is not inverted, however, for isochronous transfers.

<2> In an IN transaction, the data packet whose packet ID matches E4TGL is transmitted. The state of E4TGL is automatically inverted when an ACK from the host is received.

#### E4OVR (bit 5): EP4 payload over flag

<1> This bit is automatically set to 1 when a volume of data exceeding the maximum payload size specified in EP4CNT is received.

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

#### E4STL (bit 4): EP4 STALL flag

<1> A STALL handshake is returned for IN and OUT transactions.

#### E4ACK (bit 3): EP4 ACK flag

<1> When this bit is set to 1, a data packet is transmitted for an IN transaction and an ACK handshake is returned for an OUT transaction.

<2> When this bit is set to 0, a NAK handshake is returned for IN and OUT transactions.

<3> See Table 3.20.17.

#### E4DIR (bit 2): EP4 transfer direction flag

<1> When this bit is set to 1, only the IN token for endpoint 4 is processed.

<2> When this bit is set to 0, only the OUT token for endpoint 4 is processed.

<3> See Table 3.20.17.

#### E4ISO (bit 1): EP4 isochronous transfer flag

<1> When this bit is set to 1, the transfer type of endpoint 4 is set to isochronous transfer.

<2> When this bit is set to 0, the transfer type of endpoint 4 is set to either bulk transfer or interrupt transfer.

<3> See Table 3.20.17.

#### E4BNK (bit 0): EP4 transfer RAM address control flag

<1> The values of this bit and the EPBMOD register (FEABH) switch the location of the data transmit/receive buffer area for endpoint 4.

<2> See Subsection 3.20.3.6, “Endpoint configuration.”

### 3.20.4.24 EP5 status register (EP5STA)

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE96 | 0000 0000 | R/W | EP5STA | E5EN | E5TGL | E5OVR | E5STL | E5ACK | E5DIR | E5ISO | E5BNK |

**E5EN (bit 7): EP5 enable flag**

<1> When this bit is set to 1, the token for endpoint 5 is processed.

<2> When this bit is set to 0, the token for endpoint 5 is not processed.

**E5TGL (bit 6): EP5 data toggle**

<1> In an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E5TGL. The state of E5TGL is automatically inverted when the transaction terminates normally with an ACK. The E5TGL state is not inverted, however, for isochronous transfers.

<2> In an IN transaction, the data packet whose packet ID matches E5TGL is transmitted. The state of E5TGL is automatically inverted when an ACK from the host is received.

**E5OVR (bit 5): EP5 payload over flag**

<1> This bit is automatically set to 1 when a volume of data exceeding the maximum payload size specified in EP5CNT is received.

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

**E5STL (bit 4): EP5 STALL flag**

<1> A STALL handshake is returned for IN and OUT transactions.

**E5ACK (bit 3): EP5 ACK flag**

<1> When this bit is set to 1, a data packet is transmitted for an IN transaction and an ACK handshake is returned for an OUT transaction

<2> When this bit is set to 0, a NAK handshake is returned for IN and OUT transactions.

<3> See Table 3.20.17.

**E5DIR (bit 2): EP5 transfer direction flag**

<1> When this bit is set to 1, only the IN token for endpoint 5 is processed.

<2> When this bit is set to 0, only the OUT token for endpoint 5 is processed.

<3> See Table 3.20.17.

**E5ISO (bit 1): EP5 isochronous transfer flag**

<1> When this bit is set to 1, the transfer type of endpoint 5 is set to isochronous transfer.

<2> When this bit is set to 0, the transfer type of endpoint 5 is set to either bulk transfer or interrupt transfer.

<3> See Table 3.20.17.

**E5BNK (bit 0): EP5 transfer RAM address control flag**

<1> The values of this bit and the EPBMOD register (FEABH) switch the location of the data transmit/receive buffer area for endpoint 5.

<2> See Subsection 3.20.3.6, “Endpoint configuration.”

### 3.20.4.25 EP6 status register (EP6STR)

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE97 | 0000 0000 | R/W | EP6STA | E6EN | E6TGL | E6OVR | E6STL | E6ACK | E6DIR | E6ISO | E6BNK |

#### E6EN (bit 7): EP6 enable flag

<1> When this bit is set to 1, the token for endpoint 6 is processed.

<2> When this bit is set to 0, the token for endpoint 6 is not processed.

#### E6TGL (bit 6): EP6 data toggle

<1> In an OUT transaction, the receive data is transferred to RAM only when the packet ID in the data packet from the host matches E6TGL. The state of E6TGL is automatically inverted when the transaction terminates normally with an ACK. The E6TGL state is not inverted, however, for isochronous transfers.

<2> In an IN transaction, the data packet whose packet ID matches E6TGL is transmitted. The state of E6TGL is automatically inverted when an ACK from the host is received.

#### E6OVR (bit 5): EP6 payload over flag

<1> This bit is automatically set to 1 when a volume of data exceeding the maximum payload size specified in EP6CNT is received.

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

#### E6STL (bit 4): EP6 STALL flag

<1> A STALL handshake is returned for IN and OUT transactions.

#### E6ACK (bit 3): EP6 ACK flag

<1> When this bit is set to 1, a data packet is transmitted for an IN transaction and an ACK handshake is returned for an OUT transaction.

<2> When this bit is set to 0, a NAK handshake is returned for IN and OUT transactions.

<3> See Table 3.20.17.

#### E6DIR (bit 2): EP6 transfer direction flag

<1> When this bit is set to 1, only the IN token for endpoint 6 is processed.

<2> When this bit is set to 0, only the OUT token for endpoint 6 is processed.

<3> See Table 3.20.17.

#### E6ISO (bit 1): EP6 isochronous transfer flag

<1> When this bit is set to 1, the transfer type of endpoint 6 is set to isochronous transfer.

<2> When this bit is set to 0, the transfer type of endpoint 6 is set to either bulk transfer or interrupt transfer.

<3> See Table 3.20.17.

#### E6BNK (bit 0): EP6 transfer RAM address control flag

<1> The values of this bit and the EPBMOD register (FEABH) switch the location of the data transmit/receive buffer area for endpoint 6.

<2> See Subsection 3.20.3.6, “Endpoint configuration.”

### 3.20.4.26 EP1 count register (EP1CNT)

1) This register sets the number of transmit data bytes for endpoint 1 if the transfer direction of endpoint 1 is IN (E1DIR=1).
2) If the transfer direction of endpoint 1 is OUT (E1DIR=0), this register sets the maximum payload size of endpoint 1.
3) The legitimate value range is from 00[H] to 40[H].

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE98 | H000 0000 | R/W | EP1CNT | - | E1CN6 | E1CN5 | E1CN4 | E1CN3 | E1CN2 | E1CN1 | E1CN0 |

### 3.20.4.27 EP1 receive data count register (EP1RX)

1) The number of data bytes received at endpoint 1can be read out through this register.
2) The contents of this register are updated when an OUT transaction for endpoint 1 terminates normally with an ACK.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE99 | H000 0000 | R | EP1RX | - | E1RX6 | E1RX5 | E1RX4 | E1RX3 | E1RX2 | E1RX1 | E1RX0 |

### 3.20.4.28 EP2 count register (EP2CNT)

1) This register sets the number of transmit data bytes for endpoint 2 if the transfer direction of endpoint 2 is IN (E2DIR=1).
2) If the transfer direction of endpoint 2 is OUT (E2DIR=0), this register sets the maximum payload size of endpoint 2.
3) The legitimate value range is from 00[H] to 40[H].

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE9A | H000 0000 | R/W | EP2CNT | - | E2CN6 | E2CN5 | E2CN4 | E2CN3 | E2CN2 | E2CN1 | E2CN0 |

### 3.20.4.29 EP2 receive data count register (EP2RX)

1) The number of data bytes received at endpoint 2 can be read out through this register.
2) The contents of this register are updated when an OUT transaction for endpoint 2 terminates normally with an ACK.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE9B | H000 0000 | R/W | EP2RX | - | E2RX6 | E2RX5 | E2RX4 | E2RX3 | E2RX2 | E2RX1 | E2RX0 |

### 3.20.4.30 RP3 count register (EP3CNT)

1) This register sets the number of transmit data bytes for endpoint 3 if the transfer direction of endpoint 3 is IN (E3DIR=1).
2) If the transfer direction of endpoint 3 is OUT (E3DIR=0), this register sets the maximum payload size of endpoint 3.
3) The legitimate value range is from 00[H] to 40[H].

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE9C | H000 0000 | R/W | EP3CNT | - | E3CN6 | E3CN5 | E3CN4 | E3CN3 | E3CN2 | E3CN1 | E3CN0 |

### 3.20.4.31 EP3 receive data count register (EP3RX)

1) The number of data bytes received at endpoint 3 can be read out through this register.
2) The contents of this register are updated when an OUT transaction for endpoint 3 terminates normally with an ACK.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE9D | H000 0000 | R/W | EP3RX | - | E3RX6 | E3RX5 | E3RX4 | E3RX3 | E3RX2 | E3RX1 | E3RX0 |

### 3.20.4.32 EP4 count register (EP4CNT)

1) This register sets the number of transmit data bytes for endpoint 4 if the transfer direction of endpoint 4 is IN (E4DIR=1).
2) If the transfer direction of endpoint 4 is OUT (E4DIR=0), this register sets the maximum payload size of endpoint 4.
3) The legitimate value range is from 00[H] to 40[H].

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE9E | H000 0000 | R/W | EP4CNT | - | E4CN6 | E4CN5 | E4CN4 | E4CN3 | E4CN2 | E4CN1 | E4CN0 |

### 3.20.4.33 EP4 receive data count register (EP4RX)

1) The number of data bytes received at endpoint 4 can be read out through this register.
2) The contents of this register are updated when an OUT transaction for endpoint 4 terminates normally with an ACK.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE9F | H000 0000 | R/W | EP4RX | - | E4RX6 | E4RX5 | E4RX4 | E4RX3 | E4RX2 | E4RX1 | E4RX0 |

### 3.20.4.34 EP5 count register (EP5CNT)

1) This register sets the number of transmit data bytes for endpoint 5 if the transfer direction of endpoint 5 is IN (E5DIR=1).
2) If the transfer direction of endpoint 5 is OUT (E5DIR=0), this register sets the maximum payload size of endpoint 5.
3) The legitimate value range is from 00[H] to 40[H].

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEA0 | H000 0000 | R/W | EP5CNT | - | E5CN6 | E5CN5 | E5CN4 | E5CN3 | E5CN2 | E5CN1 | E5CN0 |

### 3.20.4.35 EP5 receive data count register (EP5RX)

1) The number of data bytes received at endpoint 5 can be read out through this register.
2) The contents of this register are updated when an OUT transaction for endpoint 5 terminates normally with an ACK.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEA2 | H000 0000 | R/W | EP5RX | - | E5RX6 | E5RX5 | E5RX4 | E5RX3 | E5RX2 | E5RX1 | E5RX0 |

### 3.20.4.36 EP6 count register (EP6CNT)

1) This register sets the number of transmit data bytes for endpoint 6 if the transfer direction of endpoint 6 is IN (E6DIR=1).
2) If the transfer direction of endpoint 6 is OUT (E6DIR=0), this register sets the maximum payload size of endpoint 6.
3) The legitimate value range is from 00[H] to 40[H].

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEA6 | H000 0000 | R/W | EP6CNT | - | E6CN6 | E6CN5 | E6CN4 | E6CN3 | E6CN2 | E6CN1 | E6CN0 |

### 3.20.4.37 EP6 receive data count register (EP6RX)

1) The number of data bytes received at endpoint 6 can be read out through this registert.

2) The contents of this register are updated when an OUT transaction for endpoint 6 terminates normally with an ACK.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEA8 | H000 0000 | R/W | EP6RX | - | E6RX6 | E6RX5 | E6RX4 | E6RX3 | E6RX2 | E6RX1 | E6RX0 |

### 3.20.4.38 Endpoint buffer mode register (EPBMOD)

1) This register sets the mapping address in RAM of the endpoint buffer.

2) See Subsection 3.20.3.6, “Endpoint configuration.”

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEAB | HHHH H000 | R/W | EPBMOD | - | - | - | - | - | EPBM2 | EPBM1 | EPBM0 |

### 3.20.4.39 Test register 0 (TESTR0)

1) This register is a test register.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEAC | 0000 0000 | R/W | TESTR0 | DPLTST | CMPTST | CMPKIL | TTXCLK | TTXREQ | TADR2 | TADR1 | P71NDL |

**DPLTST (bit 7): Test bit.**

This bit must always be set to 0.

**CMPTST (bit 6): Test bit.**

This bit must always be set to 0.

**CMPKIL (bit 5): Test bit.**

This bit must always be set to 0.

**TTXCLK (bit 4): Test bit.**

This bit must always be set to 0.

**TTXREQ (bit 3): Test bit.**

This bit must always be set to 0..

**TADR2 (bit 2): Test bit.**

This bit must always be set to 0.

**TADR1 (bit 1): Test bit.**

This bit must always be set to 0.

**P71NDL (bit 0): Test bit.**

This bit must always be set to 0.

### 3.20.4.40 Test register 1 (TESTR1)

1) This register is a test register.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEAD | 0000 0000 | R | TESTR1 | TDAT7 | TDAT6 | TDAT5 | TDAT4 | TDAT3 | TDAT2 | TDAT1 | TDAT0 |

## Table 3.20.16 Endpoint 0 Status Register Transition Chart

| EP0STA[3:0] | | | | Token | Receive Toggle | Receive Data | Receive Buffer | Response | EP0STA[3:0] | | | | End Flag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACK | CSU | CST | CRW | | | | | | ACK | CSU | CST | CRW | |
| 0 | 0 | 0 | 0 | SETUP | - | Invalid | Update | --- | 0 | 0 | 0 | 0 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | - | - | --- | 0 | 0 | 0 | 0 | - |
| | | | | IN | - | - | - | --- | 0 | 0 | 0 | 0 | - |
| 0 | 0 | 0 | 1 | SETUP | - | Invalid | Update | --- | 0 | 0 | 0 | 1 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | - | - | --- | 0 | 0 | 0 | 1 | - |
| | | | | IN | - | - | - | --- | 0 | 0 | 0 | 1 | - |
| 0 | 0 | 1 | 0 | SETUP | - | Invalid | Update | --- | 0 | 0 | 1 | 0 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | - | - | --- | 0 | 0 | 1 | 0 | - |
| | | | | IN | - | - | - | --- | 0 | 0 | 1 | 0 | - |
| 0 | 0 | 1 | 1 | SETUP | - | Invalid | Update | --- | 0 | 0 | 1 | 1 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | - | - | --- | 0 | 0 | 1 | 1 | - |
| | | | | IN | - | - | - | --- | 0 | 0 | 1 | 1 | - |
| 0 | 1 | 0 | 0 | SETUP | - | Invalid | Update | --- | 0 | 1 | 0 | 0 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | Invalid | - | --- | 0 | 1 | 0 | 0 | - |
| | | | | | - | Valid | - | NAK | 0 | 1 | 0 | 0 | NAK |
| | | | | IN | - | - | - | NAK | 0 | 1 | 0 | 0 | NAK |
| 0 | 1 | 0 | 1 | SETUP | - | Invalid | Update | --- | 0 | 1 | 0 | 1 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | Invalid | - | --- | 0 | 1 | 0 | 1 | - |
| | | | | | - | Valid | - | NAK | 0 | 1 | 0 | 1 | NAK |
| | | | | IN | - | - | - | NAK | 0 | 1 | 0 | 1 | NAK |
| 0 | 1 | 1 | 0 | SETUP | - | Invalid | Update | --- | 0 | 1 | 1 | 0 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | Invalid | - | --- | 0 | 1 | 1 | 0 | - |
| | | | | | - | Valid | - | NAK | 0 | 1 | 1 | 0 | NAK |
| | | | | IN | - | - | - | NAK | 0 | 1 | 1 | 0 | NAK |
| 0 | 1 | 1 | 1 | SETUP | - | Invalid | Update | --- | 0 | 1 | 1 | 1 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | Invalid | - | --- | 0 | 1 | 1 | 1 | - |
| | | | | | - | Valid | - | NAK | 0 | 1 | 1 | 1 | NAK |
| | | | | IN | - | - | - | NAK | 0 | 1 | 1 | 1 | NAK |
| 1 | 0 | 0 | 0 | SETUP | - | Invalid | Update | --- | 1 | 0 | 0 | 0 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | - | - | --- | 1 | 0 | 0 | 0 | - |
| | | | | IN | - | - | - | --- | 1 | 0 | 0 | 0 | - |
| 1 | 0 | 0 | 1 | SETUP | - | Invalid | Update | --- | 1 | 0 | 0 | 1 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | - | - | --- | 1 | 0 | 0 | 1 | - |
| | | | | IN | - | - | - | --- | 1 | 0 | 0 | 1 | - |
| 1 | 0 | 1 | 0 | SETUP | - | Invalid | Update | --- | 1 | 0 | 1 | 0 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | - | - | --- | 1 | 0 | 1 | 0 | - |
| | | | | IN | - | - | - | --- | 1 | 0 | 1 | 0 | - |
| 1 | 0 | 1 | 1 | SETUP | - | Invalid | Update | --- | 1 | 0 | 1 | 1 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | - | - | --- | 1 | 0 | 1 | 1 | - |
| | | | | IN | - | - | - | --- | 1 | 0 | 1 | 1 | - |

*(Continued on next page.)*

*(Continued from preceding page.)*

| EP0STA[3:0] | | | | Token | Receive | Receive | Receive | Response | EP0STA[3:0] | | | | End Flag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACK | CSU | CST | CRW | | Toggle | Data | Buffer | | ACK | CSU | CST | CRW | |
| 1 | 1 | 0 | 0 | SETUP | - | Invalid | Update | --- | 1 | 1 | 0 | 0 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | Mis-match | Invalid | - | --- | 1 | 1 | 0 | 0 | Error |
| | | | | | | Valid | - | ACK | 1 | 1 | 0 | 0 | - |
| | | | | | Match | Invalid | Update | --- | 1 | 1 | 0 | 0 | Error |
| | | | | | | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | IN | - | - | - | Status-IN (*1) | 0 | 0 | 0 | 0 | ACK (*3) |
| 1 | 1 | 0 | 1 | SETUP | - | Invalid | Update | --- | 1 | 1 | 0 | 1 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | Invalid | - | --- | 1 | 1 | 0 | 1 | Error |
| | | | | | - | Valid | - | ACK | 0 | 1 | 1 | 1 | ACK |
| | | | | IN | - | - | - | Tx Data (*2) | 0 | 1 | 0 | 1 | ACK (*3) |
| 1 | 1 | 1 | 0 | SETUP | - | Invalid | Update | --- | 1 | 1 | 1 | 0 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | Invalid | - | --- | 1 | 1 | 1 | 0 | Error |
| | | | | | - | Valid | - | STALL | 1 | 1 | 1 | 0 | STALL |
| | | | | IN | - | - | - | Status-IN (*1) | 0 | 0 | 0 | 0 | ACK (*3) |
| 1 | 1 | 1 | 1 | SETUP | - | Invalid | Update | --- | 1 | 1 | 1 | 1 | Error |
| | | | | | - | Valid | Update | ACK | 0 | 1 | 0 | 0 | ACK |
| | | | | OUT | - | Invalid | - | --- | 1 | 1 | 1 | 1 | Error |
| | | | | | - | Valid | - | ACK | 0 | 1 | 1 | 1 | ACK |
| | | | | IN | - | - | - | STALL | 1 | 1 | 1 | 1 | STALL |

Gray-shaded columns indicate register contents, receive buffer update, or responses to the token from the host.

*1: A data packet containing no data is transmitted.

*2: The number of bytes specified in EP0TX are transmitted.

*3: If the reception of an ACK packet from host fails, the error end flag (bit 3) in EP0INT is automatically set to 1 but the ACK flag (bit 3) in EP0STA is not cleared automatically.

### Table 3.20.17 Endpoint n (n＝1 to 6) Status Register Transition Chart

| EPnSTA[4:1] | | | | Token | Receive | Receive | Receive | Response | EPnSTA[4:1] | | | | End Flag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STL | ACK | DIR | ISO | | Toggle | Data | Buffer | | STL | ACK | DIR | ISO | |
| 0 | 0 | 0 | 0 | OUT | - | Invalid | - | --- | 0 | 0 | 0 | 0 | Error |
| | | | | | | Valid | - | NAK | 0 | 0 | 0 | 0 | NAK |
| | | | | IN | - | - | - | --- | 0 | 0 | 0 | 0 | - |
| 0 | 0 | 0 | 1 | OUT | - | Invalid | - | --- | 0 | 0 | 0 | 1 | Error |
| | | | | | | Valid | | --- | 0 | 0 | 0 | 1 | - |
| | | | | IN | - | - | - | --- | 0 | 0 | 0 | 1 | - |
| 0 | 0 | 1 | 0 | OUT | - | - | - | --- | 0 | 0 | 1 | 0 | - |
| | | | | IN | - | - | - | NAK | 0 | 0 | 1 | 0 | NAK |
| 0 | 0 | 1 | 1 | OUT | - | - | - | --- | 0 | 0 | 1 | 1 | - |
| | | | | IN | - | - | - | Tx 0 (*1) | 0 | 0 | 1 | 1 | - |
| 0 | 1 | 0 | 0 | OUT | Mis-match | Invalid | - | --- | 0 | 1 | 0 | 0 | Error |
| | | | | | | Valid | - | ACK | 0 | 1 | 0 | 0 | - |
| | | | | | Match | Invalid | Update | --- | 0 | 1 | 0 | 0 | Error |
| | | | | | | Valid | Update | ACK | 0 | 0 | 0 | 0 | ACK |
| | | | | IN | - | - | - | --- | 0 | 1 | 0 | 0 | - |
| 0 | 1 | 0 | 1 | OUT | - | Invalid | Update | --- | 0 | 1 | 0 | 1 | Error |
| | | | | | | Valid | Update | --- | 0 | 0 | 0 | 1 | ACK |
| | | | | IN | - | - | - | --- | 0 | 1 | 0 | 1 | - |
| 0 | 1 | 1 | 0 | OUT | - | - | - | --- | 0 | 1 | 1 | 0 | - |
| | | | | IN | - | - | - | Tx Data (*2) | 0 | 0 | 1 | 0 | ACK |
| 0 | 1 | 1 | 1 | OUT | - | - | - | --- | 0 | 1 | 1 | 1 | - |
| | | | | IN | - | - | - | Tx Data (*2) | 0 | 0 | 1 | 1 | ACK |
| 1 | 0 | 0 | 0 | OUT | - | Invalid | | --- | 1 | 0 | 0 | 0 | Error |
| | | | | | | Valid | | STALL | 1 | 0 | 0 | 0 | STALL |
| | | | | IN | - | - | - | --- | 1 | 0 | 0 | 0 | - |
| 1 | 0 | 0 | 1 | OUT | - | Invalid | - | --- | 1 | 0 | 0 | 1 | Error |
| | | | | | | Valid | - | --- | 1 | 0 | 0 | 1 | - |
| | | | | IN | - | - | - | --- | 1 | 0 | 0 | 1 | - |
| 1 | 0 | 1 | 0 | OUT | - | - | - | --- | 1 | 0 | 1 | 0 | - |
| | | | | IN | | - | - | STALL | 1 | 0 | 1 | 0 | STALL |
| 1 | 0 | 1 | 1 | OUT | - | - | - | --- | 1 | 0 | 1 | 1 | - |
| | | | | IN | - | - | - | Tx 0 (*1) | 1 | 0 | 1 | 1 | - |
| 1 | 1 | 0 | 0 | OUT | - | Invalid | - | --- | 1 | 1 | 0 | 0 | Error |
| | | | | | | Valid | - | STALL | 1 | 1 | 0 | 0 | STALL |
| | | | | IN | - | - | - | | 1 | 1 | 0 | 0 | - |
| 1 | 1 | 0 | 1 | OUT | - | Invalid | Update | --- | 1 | 1 | 0 | 1 | Error |
| | | | | | | Valid | Update | --- | 1 | 0 | 0 | 1 | ACK |
| | | | | IN | - | - | - | --- | 1 | 1 | 0 | 1 | - |
| 1 | 1 | 1 | 0 | OUT | - | - | - | --- | 1 | 1 | 1 | 0 | - |
| | | | | IN | - | - | - | STALL | 1 | 1 | 1 | 0 | STALL |
| 1 | 1 | 1 | 1 | OUT | - | - | - | --- | 1 | 1 | 1 | 1 | - |
| | | | | IN | - | - | - | Tx Data (*2) | 1 | 0 | 1 | 1 | ACK |

Gray-shaded columns indicate register contents, receive buffer update, or responses to the token from the host.

*1: A data packet containing no data is transmitted.

*2: The number of bytes specified in EPnTX are transmitted.

## 3.20.5 USB Communication Example

### 3.20.5.1 Setting up the clocks

1) Configure the user option, USBDIV (FE04H), and PLLCNT (FE0DH) according to the frequency (8 or 12 MHz) of the CF main clock oscillator. See Subsection 3.20.4.1, "USB frequency divided clock control register," and Subsection 3.20.4.2, "USB PLL control register."

2) Set OCR register, bit 7 (CLKSGL) and bit 4 (CLKCB4) to 1 to designate the main clock as the system clock.

* *Secure a stabilization time of 20 ms or longer after setting up the PLL circuit.*

* *To perform USB communication, it is necessary to set the system clock frequency to 8 to 12 MHz.*

**Table 3.20.18 Sample Clock Settings**

| Main Clock Oscillator Frequency | User Option | Register | | |
|---|---|---|---|---|
| | Main Clock 8 MHz Selection | PLLCNT | USBDIV | OCR |
| 8 MHz | Enable | 00[H] | 10[H] | 92[H] |
| 12 MHz | Disable | 00[H] or 80[H] | | |

### 3.20.5.2 Setting up the endpoint buffer

1) Select the most appropriate endpoint buffer configuration by setting the EPBMOD register (FEABH) according to the number of endpoints to be used and required data buffer size. See Subsection 3.20.3.6, “Endpoint configuration."

### 3.20.5.3 Configuration for USB communication

Refer to the separate document entitled "USB Application Note."

## 3.20.6 USB Interface HALT Mode Operation

1) When HALT mode is entered, USB communication involving data transmission and reception cannot proceed normally since the automatic data transfer between RAM (endpoint buffer) and the transmit/receive buffer is interrupted in that case.

2) HALT mode can be released by generating a USB interface interrupt that involves neither data transmission nor reception.

# 3.21.1 High-current Drivers

## 3.21.1 Function

This series of microcontrollers incorporates 2 channels of high-current P-channel drivers and 3 channels of high-current N-channel drivers.

| Port Name | Driver Type |
|---|---|
| P02 | N-channel driver (TDN2) |
| P01 | P-channel driver (TDP1) |
| P00 | N-channel driver (TDN1) |
| PWM0 | P-channel driver (TDP0) |
| PWM1 | N-channel driver (TDN0) |

## 3.21.2 Related Register

### 3.21.2.1 High-current driver control register (BUFCNT)

1) This register controls the high-current drivers for P02 to P00, PWM0, and PWM1.

2) The register is also used to select the AD conversion input signals.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEFF | 0000 0000 | R/W | BUFCNT | ADCHSEL4 | BUFCNT6 | BUFP1 | BUFP0 | BUFCNT3 | BUFN2 | BUFN1 | BUFN0 |

**ADCHSEL4 (bit 7): AD conversion input signal select**

This bit and bits 7 to 4 (ADCHSEL3 to ADCHSEL0) of the AD control register (ADCRC:FE58) are used to select the AD conversion input signals (AN0 to AN19).

**BUFCNT6 (bit 6): Fixed bit**

This bit must always be set to 0.

**BUFP1 (bit 5): P01 high-current P-channel driver control**

The high-current P-channel driver for P01 is turned on if this bit is set to 1 and P01DDR to 0 when the CMOS is selected as the output type of port P01 through user option configuration.

**BUFP0 (bit 4):PWM0 high-current P-channel driver control**

The high-current P-channel driver for PWM0 is turned on when this bit is set to 1, ENPWM0 ( PWM0C:FE24, bit 2) to 0, and PWM0L2 (PWM0L:FE20, bit 6) to 0.

**BUFCNT3 (bit 3): Fixed bit**

This bit must always be set to 0.

**BUFN2 (bit 2): P02 high-current N-channel driver control**

The high-current N-channel driver for P02 is turned on when this bit is set to 1 and P02DDR to 0.

**BUFN1 (bit 1):P00 high-current N-channel driver control**

The high-current N-channel driver for P00 is turned on when this bit is set to 1 and P00DDR to 0.

**BUFN0 (bit 0): PWM1 high-current N-channel driver control**

The high-current N-channel driver for PWM1 is turned on when this bit is set to 1, ENPWM1 (PWM0C:FE24, bit 3) to 0, and PWM1L2 (PWM1L:FE22, bit 6) to 0.

## High-current Drivers

| Register Setting | P-channel Driver State | | N-channel Driver State | | |
|---|---|---|---|---|---|
| BUFCNT | TDP1 | TDP0 | TDN2 | TDN1 | TDN0 |
| 0001__0001 | OFF | ON | OFF | OFF | ON |
| 0001__0010 | OFF | ON | OFF | ON | OFF |
| 0001__0100 | OFF | ON | ON | OFF | OFF |
| 0010__0001 | ON | OFF | OFF | OFF | ON |
| 0010__0010 | ON | OFF | OFF | ON | OFF |
| 0010__0100 | ON | OFF | ON | OFF | OFF |

# 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 CPU 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, it 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 an interrupt request of the same level or lower level than the level 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/USB bus active |
| 4 | 0001BH | H or L | INT3/INT5/base timer |
| 5 | 00023H | H or L | T0H/INT6 |
| 6 | 0002BH | H or L | TIL/TIH/INT7 |
| 7 | 00033H | H or L | SIO0/USB bus reset/USB suspend/UART1 receive end/ SCUART receive end |
| 8 | 0003BH | H or L | SIO1/USB endpoint/USB-SOF/SIO4/UART1 buffer empty/UART1 transmit end/SCUART buffer empty/ SCUART transmit end |
| 9 | 00043H | H or L | ADC/T6/T7 |
| 10 | 0004BH | H or L | Port 0/PWM0/PWM1/T4/T5 |

- 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 requests to be accepted.

A 0 in this bit disables H- and L-level interrupt requests 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 |  | Interrupt Level |
|---|---|---|---|---|
|  |  |  | Value |  |
| 7 | 0004BH | IP4B | 0 | L |
|  |  |  | 1 | H |
| 6 | 00043H | IP43 | 0 | L |
|  |  |  | 1 | H |
| 5 | 0003BH | IP3B | 0 | L |
|  |  |  | 1 | H |
| 4 | 00033H | IP33 | 0 | L |
|  |  |  | 1 | H |
| 3 | 0002BH | IP2B | 0 | L |
|  |  |  | 1 | H |
| 2 | 00023H | IP23 | 0 | L |
|  |  |  | 1 | H |
| 1 | 0001BH | IP1B | 0 | L |
|  |  |  | 1 | H |
| 0 | 00013H | IP13 | 0 | L |
|  |  |  | 1 | H |

# 4.2 System Clock Generator Function

## 4.2.1 Overview

This series of microcontrollers incorporates four systems of oscillator circuits, i.e., a main clock oscillator, a subclock oscillator, an RC oscillator, and a USB PLL oscillator as system clock generator circuits. The RC oscillator circuit has internal resistors and capacitors, so that no external circuit is required.

The system clock can be selected from these four types of clock sources under program control.

## 4.2.2 Functions

1) System clock select
   - Allows the system clock to be selected under program control from four types of clocks generated by the main clock oscillator, subclock oscillator, RC oscillator, and USB 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 $\frac{1}{128}$.

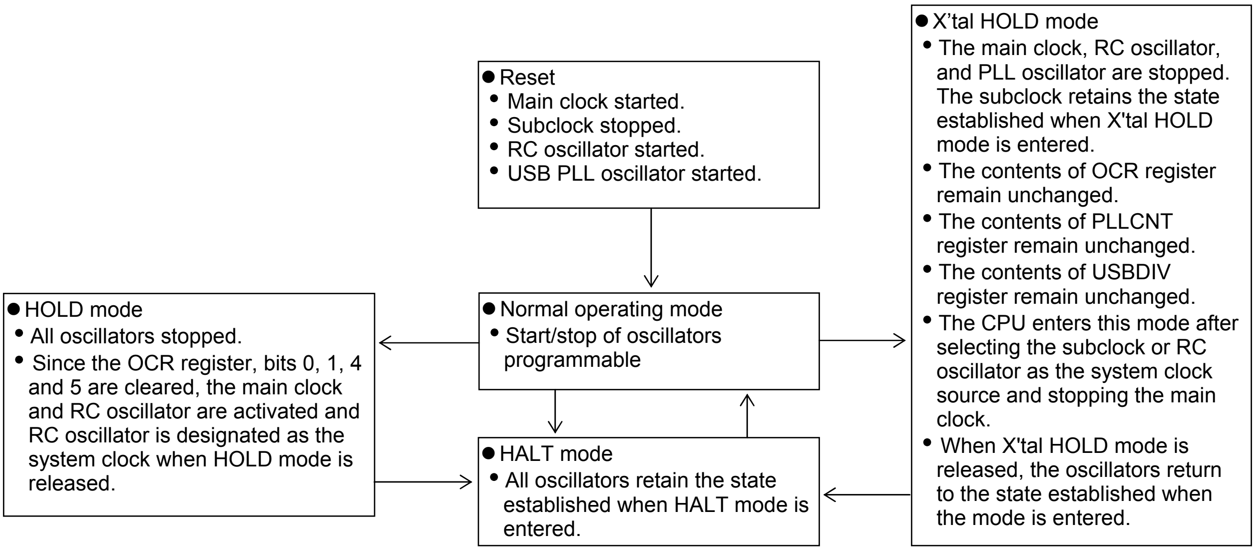
3) Oscillator circuit control
   - The four systems of oscillators are stopped/started by instructions.

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 | RC Oscillator | USB PLL Oscillator | System Clock |
|---|---|---|---|---|---|
| Reset | Running | Stopped | Running | Running | RC oscillator |
| Normal mode | Programmable | Programmable | Programmable | Programmable | Programmable |
| HALT | State established at entry time | State established at entry time | State established at entry time | State established at entry time | State established at entry time |
| HOLD | Stopped | Stopped | Stopped | Stopped | Stopped |
| Immediately after exit from HOLD mode | Running | State established at entry time | Running | State established at entry time | RC oscillator |
| X'tal HOLD | Stopped | State established at entry time | Stopped | Stopped | Stopped |
| Immediately after 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 |

See Section 4.3 “Standby Function” for the procedures to enter and exit microcontroller operating modes.



● Reset
- Main clock started.
- Subclock stopped.
- RC oscillator started.
- USB PLL oscillator started.

● Normal operating mode
- Start/stop of oscillators programmable

● HOLD mode
- All oscillators stopped.
- Since the OCR register, bits 0, 1, 4 and 5 are cleared, the main clock and RC oscillator are activated and RC oscillator is designated as the system clock when HOLD mode is released.

● HALT mode
- All oscillators retain the state established when HALT mode is entered.

● X'tal HOLD mode
- The main clock, RC oscillator, and PLL oscillator are stopped. The subclock retains the state established when X'tal HOLD mode is entered.
- The contents of OCR register remain unchanged.
- The contents of PLLCNT register remain unchanged.
- The contents of USBDIV register remain unchanged.
- The CPU enters this mode after selecting the subclock or RC oscillator as the system clock source and stopping the main clock.
- When X'tal HOLD mode is released, the oscillators return to the state established when the mode is entered.

6) It is necessary to manipulate the following special function registers to control the system clock.

- USBDIV, PCON, CLKDIV, PLLCNT, OCR

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE04 | 0000 0000 | R/W | USBDIV | CF48ON | DVCKON | DVCKDR | UREFSEL | CF12OFF | UDVSEL2 | UDVSEL1 | UDVSEL0 |
| FE07 | HHHH H000 | R/W | PCON | - | - | - | - | - | XTIDLE | PDN | IDLE |
| FE0C | HHHH H000 | R/W | CLKDIV | - | - | - | - | - | CLKDV2 | CLKDV1 | CLKDV0 |
| FE0D | 0000 0000 | R/W | PLLCNT | SELREF2 | SELREF1 | SELREF0 | PLLTEST | VCOSTP | CMPSTP | PLLCNT1 | PLLCNT0 |
| FE0E | 0000 XX00 | R/W | OCR | CLKSGL | EXTOSC | CLKCB5 | CLKCB4 | XT2IN | XT1IN | RCSTOP | CFSTOP |
| FE43 | 0000 0000 | R/W | XT2PC | XT2PCB7 | XT2PCB6 | XT2PCB5 | XT2PCB4 | XT2PCB3 | XT2PCB2 | XT2DR | XT2DT |

## 4.2.3 Circuit Configuration

### 4.2.3.1 Main clock oscillator circuit

1) The main clock oscillator circuit is prepared for oscillation by connecting a ceramic resonator and a capacitor 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, a feedback resistor, and a damping resistor to the XT1 and XT2 pins.
2) The data at the XT1 and XT2 pins can be read as bits 2 and 3 of the OCR register.
3) When the subclock is not to be used, connect XT1 to VDD, keep the XT2 pin open, and set bit 6 of the OCR register.

### 4.2.3.3 Internal RC oscillator circuit

1) This circuit oscillates according to the internal resistors and capacitors.
2) The clock from the RC oscillator is designated as the system clock after the reset or HOLD mode is released.
3) Unlike the main clock and subclock oscillator, the RC oscillator starts oscillation at a normal frequency from the beginning of oscillation.

#### 4.2.3.4 Internal PLL oscillator circuit for the USB

1) The main clock is oscillated by connecting a ceramic resonator and a capacitor to the CF1 and CF2 pins.
2) An external circuit is connected to the UFILT/P34 pin (see the Data Sheet).
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.

#### 4.2.3.5 Power control register (PCON) (3-bit register)

1) This register specifies the operating mode (Normal/HALT/HOLD/X'tal HOLD).

#### 4.2.3.6 Oscillation control register (OCR) (8-bit register)

1) This register controls the start/stop operation of the oscillator circuits.
2) This register selects the system clock.
3) The register sets the 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.

#### 4.2.3.7 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.

#### 4.2.3.8 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).

#### 4.2.3.9 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 selects either CF or USB frequency-divided clock as the main clock.

#### 4.2.3.10 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.

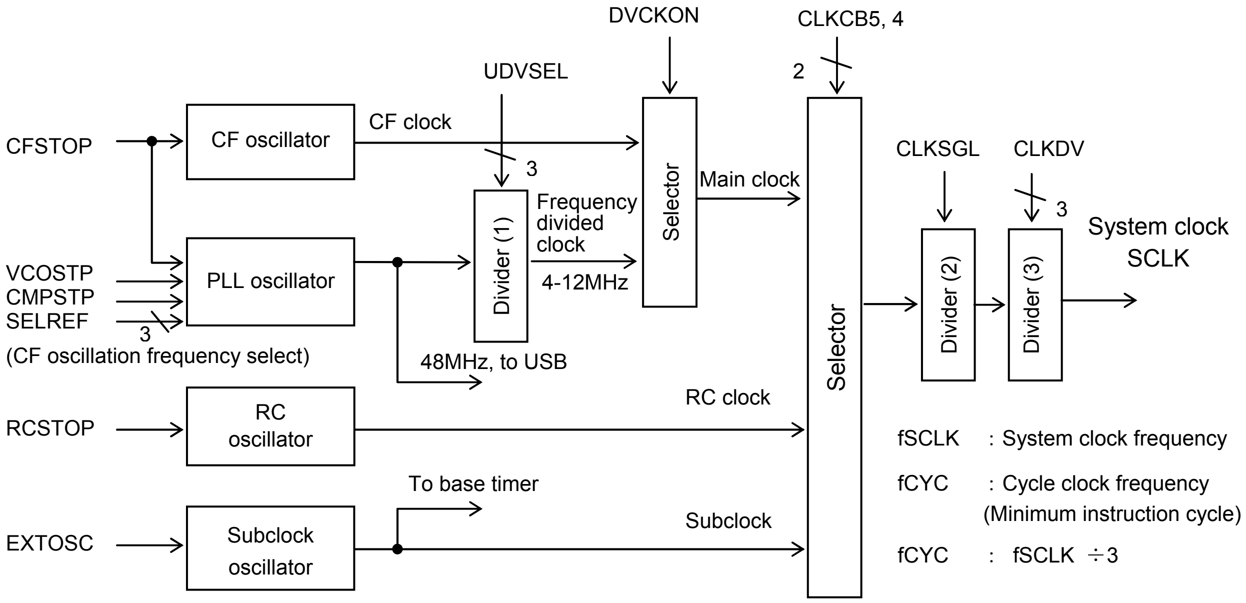


**Figure 4.2.1 System Clock Generator Block Diagram**

## 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 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 CPU enters HOLD mode, all oscillators (main clock, subclock, RC, and PLL) are suspended and bits 0, 1, 4, and 5 of the OCR are set to 0.
- When the CPU exits HOLD mode, the main clock and RC oscillator start oscillation. The subclock and PLL oscillator return to their state that was established before the CPU entered HOLD mode, and the system clock is set to RC.
- When the CPU enters X'tal HOLD mode, all oscillators except the subclock (main clock, RC, and PLL) are suspended but the state of the OCR register remains unchanged.
- 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 RC oscillator clock when X'tal HOLD mode is entered.

- Since X'tal HOLD mode is used usually for low-current clock counting, less current will be consumed if the system clock is switched to the subclock, and the main clock and RC oscillator 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, P0INT, or USB bus active) 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 CPU into HALT mode.

<2> This bit is automatically set when bit 1 is set.

<3> This bit is cleared on acceptance of an interrupt request or on receipt of a reset signal.

## 4.2.4.2 Oscillation control register (OCR) (8-bit register)

1) This register is an 8-bit register that is used 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 are prepared for oscillation when a crystal resonator (32.768 kHz typ), capacitors, feedback resistors, 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.

### CLKCB5 (bit 5): System clock select

### CLKCB4 (bit 4): System clock select

<1> CLKCB5 and CLKCB4 are used to select the system clock.

<2> CLKCB5 and CLKCB4 are cleared on reset or when HOLD mode is entered.

| CLKCB5 | CLKCB4 | System Clock |
|---|---|---|
| 0 | 0 | Internal RC oscillator |
| 0 | 1 | Main clock |
| 1 | 0 | Subclock |
| 1 | 1 | Main clock |

### XT2IN (bit 3): XT2 data (read only)

### XT1IN (bit 2): XT1 data (read only)

<1> Data that can be read via XT1IN varies as 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): Internal RC oscillator control

<1> Setting this bit to 1 stops the internal RC oscillator circuit.

<2> Setting this bit to 0 starts the internal RC oscillator circuit.

<3> This bit is cleared on reset or when HOLD mode is entered and oscillation becomes possible.

### CFSTOP (bit 0): Main clock oscillator control

<1> Setting this bit to 1 stops the main clock oscillator circuit.

<2> Setting this bit to 0 starts the main clock oscillator circuit.

<3> This bit is cleared on reset or when HOLD mode is entered 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 type) at the XT2 pin.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE43 | 0000 0000 | R/W | XT2PC | XT2PCB7 | XT2PCB6 | XT2PCB5 | XT2PCB4 | XT2PCB3 | XT2PCB2 | XT2DR | XT2DT |

### XT2PCB7 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 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 (bit 7): PLL reference clock frequency select

### SELREF1 (bit 6): PLL reference clock frequency select

### SELREF0 (bit 5): PLL reference clock frequency select

<1> Theses bits 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 can be 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 3.20.3 to the P34/UFILT pin.

<2> Set these bits to 1 when the PLL is not to be used.

### PLLCNT1 (bit 1): Reserved bit

### PLLCNT0 (bit 0): Reserved bit

## 4.2.4.5 USB frequency-divided clock control register (USBDIV) (8-bit register)

1) This register is used to select the frequency of the frequency-divided clock that is derived from the 48 MHz clock for the USB.

| Address | Initial Value | R/W | Name | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FE04 | 0000 0000 | R/W | USBDIV | CF48ON | DVCKON | DVCKDR | UREFSEL | CF12OFF | UDVSEL2 | UDVSEL1 | UDVSEL0 |

### CF48ON (bit 7): Reserved bit

This bit must always be set to 0.

### DVCKON (bit 6): Frequency-divided clock select flag

<1> This bit must be set to 1 to select the frequency-divided clock that is derived from the 48 MHz clock for the USB as the main clock. When the OCR register is configured to select the main clock (CLKCB4 set to 1) in this case, the frequency-divided clock is supplied as the system clock.

<2> Set this bit to 0 to select the CF clock as the main clock.

<3> When changing the setting of this bit, make sure that the system clock selection of the OCR register is set to a clock other than the main clock (CLKCB4 = 0).

### DVCKDR (bit 5): Frequency-divided clock external output control flag

<1> This bit must be set to 1 to transmit the frequency-divided clock that is derived from the 48 MHz clock for the 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 frequency-divided clock is transmitted from P73.

<2> Set this bit to 0 if the frequency-divided clock is not transmitted to any external device.

### UREFSEL (bit 4): PLL reference clock select

<1> When this bit is set to 0, the frequency-divided clock (8 or 12 division) of the CF main clock (8MHz or 12MHz) is multiplied by 48 and supplied as the 48 MHz clock for the USB.

<2> When this bit is set to 1, the CF main clock (8MHz or 12MHz) is multiplied by 6 or 4 and supplied as the 48 MHz clock for the USB.

### CF12OFF (bit 3): Reserved bit

This bit must always be set to 0

### UDVSEL2 (bit 2): Frequency-divided clock frequency select

### UDVSEL1 (bit 1): Frequency-divided clock frequency select

### UDVSEL0 (bit 0): Frequency-divided clock frequency select

<1> These bits select the frequency of the frequency-divided clock of the 48 MHz clock for the USB.

<2> Set the frequency-divided clock to 8 to 12 MHz to run the USB interface control circuit by supplying the frequency-divided clock as the system clock.

<3> To change the setting of the 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 suspended |
| 001 | 4.8 |
| 010 | 6 |
| 011 | 8 |
| 100 | 12 |
| 101 | 24 (inhibited) |
| 110 | 4 |
| 111 | 16 (inhibited) |

### 4.2.4.6 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 (bit 2):**
**CLKDV1 (bit 1):**
**CLKDV0 (bit 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 CPU stops execution of the instructions but the peripheral circuits continue processing.
- 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. Execution of the instructions is stopped and the peripheral circuits stop processing.
- HOLD mode is entered by setting bit 1 of the PCON register to 1 when bit 2 is set to 0. In this case, bit 0 of the PCON register (HALT mode setting flag) is automatically set.
- When a reset occurs or a HOLD mode release signal (INT0, INT1, INT2, INT4, INT5, P0INT, or USB bus active) occurs, bit 1 of the PCON register is cleared and the microcontroller switches into HALT mode.

3) X’tal HOLD mode

- All oscillators except the subclock are suspended. Execution of the instructions is stopped and all the peripheral circuits except the base timer are suspended.
- 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 or an X’tal HOLD mode release signal (base timer interrupt, INT0, INT1, INT2, INT4, INT5, P0INT, or USB bus active) occurs, bit 1 of the PCON register is cleared and the microcontroller switches into HALT mode.

## 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, RC, and PLL) are suspended and bits 0, 1, 4, and 5 of the OCR are set to 0.
- When the microcontroller exits HOLD mode, the main clock and RC oscillator start oscillation. The subclock and PLL oscillator return to their state that was established before HOLD mode is entered, and the system clock is set to RC.
- When the microcontroller enters X'tal HOLD mode, all oscillators except the subclock (main clock, RC, and PLL) are suspended but the state of the OCR register remains unchanged.
- 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 RC oscillator clock when X'tal HOLD mode is entered.
- Since X'tal HOLD mode is used usually for low-current clock counting, less current will be consumed if the system clock is switched to the subclock and the main clock and RC oscillator 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, P0INT, or USB bus active) or a reset signal occurs.

### IDLE (bit 0): HALT mode setting flag

<1> Setting this bit places the microcontroller into HALT mode.

<2> This bit is cleared on acceptance of an interrupt request or on receipt of a reset signal.

### Table 4.3.1 Standby Mode Operations

| Item/Mode | Reset State | HALT Mode | HOLD Mode | X'tal HOLD Mode |
|---|---|---|---|---|
| Entry conditions | • $\overline{\text{RES}}$ applied<br>• Reset from POR/ LVD<br>• Reset from watchdog timer | PCON register<br>Bit 1=0<br>Bit 0=1 | PCON register<br>Bit 2=0<br>Bit 1=1 | PCON register<br>Bit 2=1<br>Bit 1=1 |
| Data changed on entry | Initialized as shown in separate table. (WDTCNT register, bit 7 is set when reset is generated by watchdog timer.) | • WDTCNT, bit 5 is cleared if WDTCNT register, bits 4/3 are set to 0/1. | • WDTCNT, bit 5 is cleared if WDTCNT register, bit 4/3 are set to 0/1.<br>• PCON register, bit 0 is set.<br>• OCR register (FE0E), bits 5, 4, 1 and 0 are cleared. | • WDTCNT, bit 5 is cleared if WDTCNT register, bit 4/3 are set to 0/1.<br>• PCON register, bit 0 is set. |
| Main clock oscillation | Running | State established at entry time | Stopped | Stopped |
| Internal RC oscillation | Running | State established at entry time | Stopped | Stopped |
| Subclock oscillation | Stopped | State established at entry time | Stopped | State established at entry time |
| USB-dedicated PLL oscillation | Running | State established at entry time | Stopped | Stopped |
| CPU | Initialized | Stopped | Stopped | Stopped |
| I/O pin state | See Table 4.3.2. | ← | ← | ← |
| RAM | • $\overline{\text{RES}}$: Undefined<br>• LVD: Undefined or data retained (depends on supply voltage)<br>• Watchdog timer: Data retained | Data retained | Data retained | Data retained |
| Base timer | Stopped | State established at entry time | Stopped | State established at entry time |
| Peripheral modules except base timer | Stopped | State established at entry time (Note 2) | Stopped | Stopped |
| Exit conditions | Entry conditions cancelled | • Interrupt request accepted<br>• Reset/entry conditions established | • Interrupt request from INT0 to INT2, INT4, INT5, P0INT, or USB bus active generated<br>• Reset/entry conditions established | • Interrupt request from INT0 to INT2, INT4, INT5, P0INT, USB bus active, or base timer generated<br>• Reset/entry conditions established |
| Returned mode | Normal mode | Normal mode (Note 1) | HALT mode (Note 1) | HALT mode (Note 1) |
| Data changed on exit | None | PCON register, bit 0 is cleared | PCON register, bit 1 is cleared | PCON register, bit 1 is cleared |

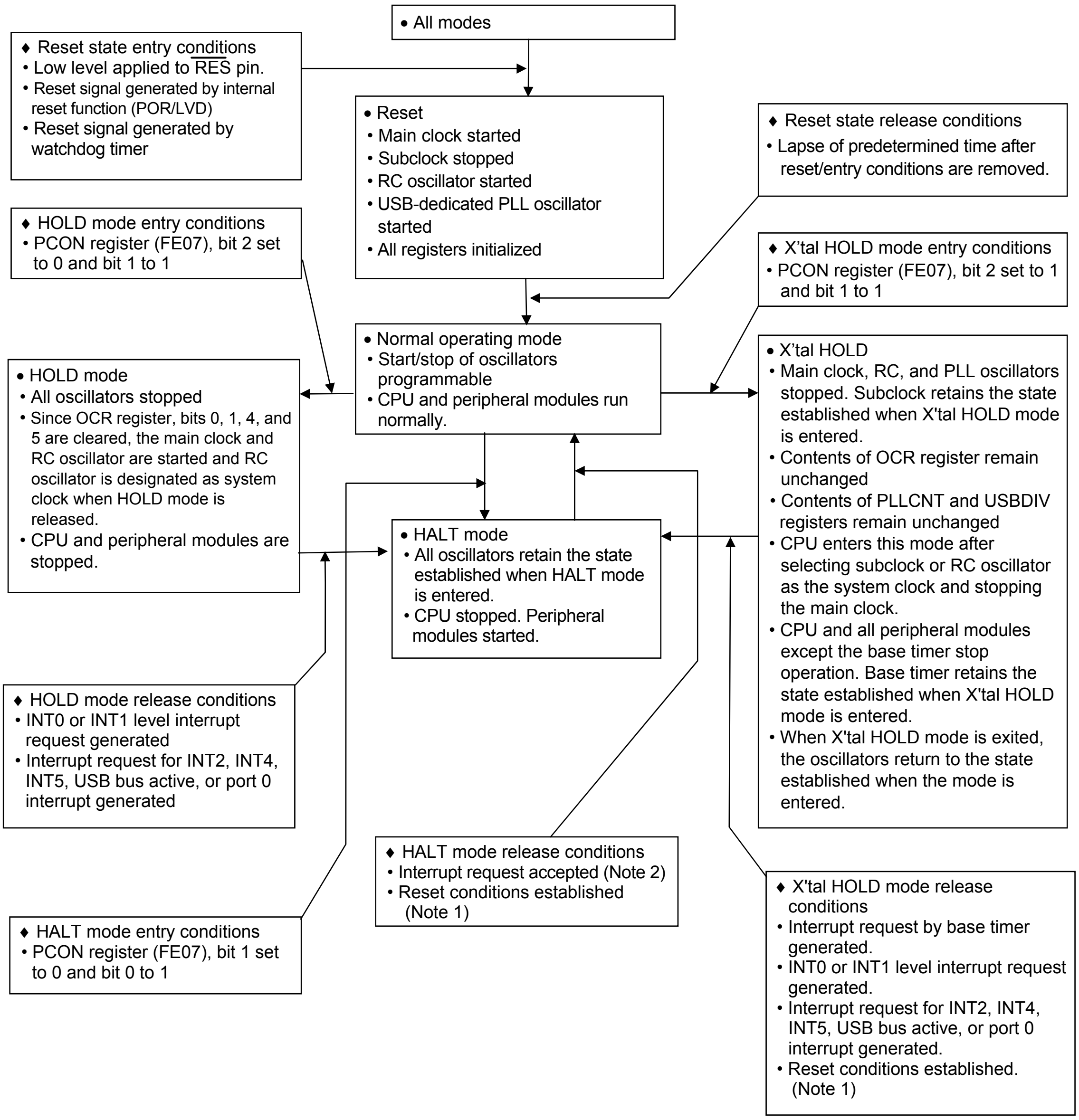
*Note 1: The microcontroller switches into the reset state if it exits the current mode on the establishment of reset/entry conditions.*

*Note 2: Some functions of serial transfer and USB interface control circuit are stopped.*

### 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 |
|---|---|---|---|---|---|
| $\overline{\text{RES}}$ | • Input<br>*Low level is generated when internal reset circuit is active. | Input | ← | ← | ← |
| XT1 | • Input<br>• X'tal oscillator will not start.<br>• Feedback resistor between XT1 and XT2 is turned off. | • Controlled by the register OCR (FE0EH) as X'tal oscillator input pin<br>• XT1 data can be read through the register OCR (FE0EH) (0 is always read in oscillation mode.)<br>• Feedback resistor between XT1 and XT2 is controlled by a program. | ← | • Oscillation suspended when used as X'tal oscillator input pin<br>* Oscillation state maintained in X'tal HOLD mode<br>• Feedback resistor between XT1 and XT2 is in the state established at entry time. | • State established at entry time |
| XT2 | • Input<br>• X'tal oscillator will not start.<br>• Feedback resistor between XT1 and XT2 is turned off. | • Controlled by the register OCR (FE0EH) as X'tal oscillator output pin<br>• XT2 data can be read through the register OCR (FE0EH).<br>• 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<br>* Oscillation state maintained in X'tal HOLD mode<br>• Feedback resistor between XT1 and XT2 is in the state established at entry time. | • State established at entry time |
| CF1 | • CF oscillator inverter input<br>• Feedback resistor present between CF1 and CF2. | • CF oscillator inverter input<br>• Oscillation enabled/ disabled by the register OCR (FE0EH)<br>• Feedback resistor present between CF1 and CF2. | ← | • CF oscillator inverter input<br>• Oscillation enabled<br>• Feedback resistor present between CF1 and CF2. | • Same as reset time |
| CF2 | • CF oscillator inverter output<br>• Oscillation enabled | • CF oscillator inverter output<br>• Oscillation enabled/ disabled by the register OCR (FE0EH)<br>• Always set to VDD level regardless of CF1 state when oscillation is suspended. | ← | • CF oscillator inverter output<br>• Oscillation suspended<br>• Always set to VDD level regardless of CF1 state | • Same as reset time |
| P00 to P07 | • Input mode<br>• Pull-up resistor off | • Input/output/pull-up resistor controlled by a program | ← | ← | ← |
| P10 to P17 | | | | | |
| P20 to P27 | | | | | |
| P31 to P33 | | | | | |
| P34 | • Input mode<br>• Pull-up resistor off (Note 1) | • Input/output/pull-up resistor controlled by a program (Note 1) | ← | • Input/output/pull-up resistor controlled by a program | • Same as in normal mode |
| P70 | • Input mode<br>• Pull-up resistor off | • Input/output/pull-up resistor controlled by a program | • Input mode<br>• Pull-up resistor off | ← | • Same as in normal mode |
| P71 to P73 | • Input mode<br>• Pull-up resistor off | • Input/output/pull-up resistor controlled by a program | ← | ← | ← |

*Note 1: PLL frequency adjustment signals are output in USB-dedicated PLL oscillation mode.*


• All modes
♦ Reset state entry conditions
• Low level applied to RES pin.
• Reset signal generated by internal reset function (POR/LVD)
• Reset signal generated by watchdog timer
• Reset
• Main clock started
• Subclock stopped
• RC oscillator started
• USB-dedicated PLL oscillator started
• All registers initialized
♦ Reset state release conditions
• Lapse of predetermined time after reset/entry conditions are removed.
♦ HOLD mode entry conditions
• PCON register (FE07), bit 2 set to 0 and bit 1 to 1
♦ X'tal HOLD mode entry conditions
• PCON register (FE07), bit 2 set to 1 and bit 1 to 1
• Normal operating mode
• Start/stop of oscillators programmable
• CPU and peripheral modules run normally.
• HOLD mode
• All oscillators stopped
• Since OCR register, bits 0, 1, 4, and 5 are cleared, the main clock and RC oscillator are started and RC oscillator is designated as system clock when HOLD mode is released.
• CPU and peripheral modules are stopped.
• X'tal HOLD
• Main clock, RC, and PLL oscillators stopped. Subclock retains the state established when X'tal HOLD mode is entered.
• Contents of OCR register remain unchanged
• Contents of PLLCNT and USBDIV registers remain unchanged
• CPU enters this mode after selecting subclock or RC oscillator as the system clock and stopping the main clock.
• CPU and all peripheral modules except the base timer stop operation. Base timer retains the state established when X'tal HOLD mode is entered.
• When X'tal HOLD mode is exited, the oscillators return to the state established when the mode is entered.
• HALT mode
• All oscillators retain the state established when HALT mode is entered.
• CPU stopped. Peripheral modules started.
♦ HOLD mode release conditions
• INT0 or INT1 level interrupt request generated
• Interrupt request for INT2, INT4, INT5, USB bus active, or port 0 interrupt generated
♦ HALT mode release conditions
• Interrupt request accepted (Note 2)
• Reset conditions established (Note 1)
♦ X'tal HOLD mode release conditions
• Interrupt request by base timer generated.
• INT0 or INT1 level interrupt request generated.
• Interrupt request for INT2, INT4, INT5, USB bus active, or port 0 interrupt generated.
• Reset conditions established. (Note 1)
♦ HALT mode entry conditions
• PCON register (FE07), bit 1 set to 0 and bit 0 to 1

*Note 1: The CPU enters the reset state when the reset/entry conditions are established.*

*Note 2: The CPU cannot return from HALT mode since no interrupt request can be accepted unless its interrupt level is higher than the interrupt level that placed the CPU into HALT, HOLD, or X'tal HOLD mode.*

| Interrupt level at which the CPU entered HALT, HOLD, or X'tal HOLD mode | Interrupt request level that can release HALT mode |
|---|---|
| No interrupt request present | X, H and L levels |
| L level | X and H levels |
| H level | X level |
| X level | None (unable to release with interrupt) |

**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 three types of reset functions:

1) External reset via the $\overline{\mathrm{RES}}$ pin

- The microcontroller is reset without fail by applying a low level to the $\overline{\mathrm{RES}}$ 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 $\overline{\mathrm{RES}}$ 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 select the power-on reset release level, to enable (use) and disable (non-use) the low-voltage detection reset function, and its threshold level.

3) Reset function by the watchdog timer

- Reset signals can be generated at regular 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.

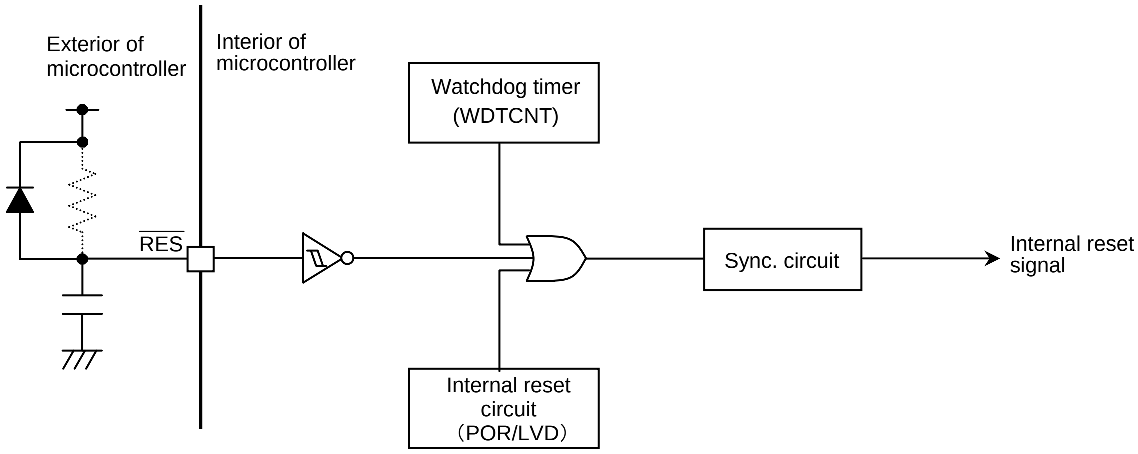

Exterior of microcontroller
Interior of microcontroller
Watchdog timer (WDTCNT)
RES
Sync. circuit
Internal reset signal
Internal reset circuit （POR/LVD）

**Figure 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 RC oscillator when a reset occurs, hardware initialization is also carried out immediately even at power-on time. Be sure to secure the oscillation stabilization time before switching the system clock source to the main clock.

On reset, the program counter is initialized to the program start address that is selected through a user option. The special function registers (SFRs) are also initialized to the values that are listed in 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 4.6, "Internal Reset Function."*

# 4.5 Watchdog Timer (WDT)

## 4.5.1 Overview

This series of microcontrollers incorporates a watchdog timer (WDT) that has the following functions:

1) Capable of generating an internal reset signal 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, stop operation, and stop count operation while retaining the count value).

* The primary function of the watchdog timer is to detect program runaway conditions. The use of the watchdog timer is highly recommended to enhance system reliability.

## 4.5.2 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 the program so that the WDTCT can be cleared at regular 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 stopped by a program runaway condition, the WDT continues operation, making it possible to detect the runaway condition.
- If the WDT operation is started when the subclock is selected as the WDT clock source, a WDT reset is generated on detection of a subclock oscillation stopped by the XT1 and XT2 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, stop operation, and stop 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 dozens of 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 (WDT).

- 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 |

## 4.5.3 Circuit Configuration

### 4.5.3.1 WDT control register (WDTCNT) (8-bit register)

1) This register is used to manipulate the WDT reset detection flag, to select operations in standby mode, to select the overflow time, and to control the operation of the WDT.

*Note: WDTCNT is initialized to 00H when a low level is applied to the external* $\overline{\text{RES}}$ *pin or when a reset is generated by the internal reset (POR/LVD) function. Bit 6 and bits 4 to 0 of WDTCNT are not initialized, however, when a reset is generated by the WDT.*

*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 tens of µA flows. (For details, refer to the latest "SANYO Semiconductors Data Sheet").*

### 4.5.3.2 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 to IDLOP0 (WDTCNT, bits 4, 3) are set to 2, the CPU 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, the WDTRUN clear signal, and the WDTRSTF (WDTCNT, bit 7) set signal.

4) Reset: Setting WDTRUN to 0 or WDTRUN to 1 and executing the **MOV #55H, WDTCNT** instruction.

* See Figure 4.5.2 for details on WDT operation.

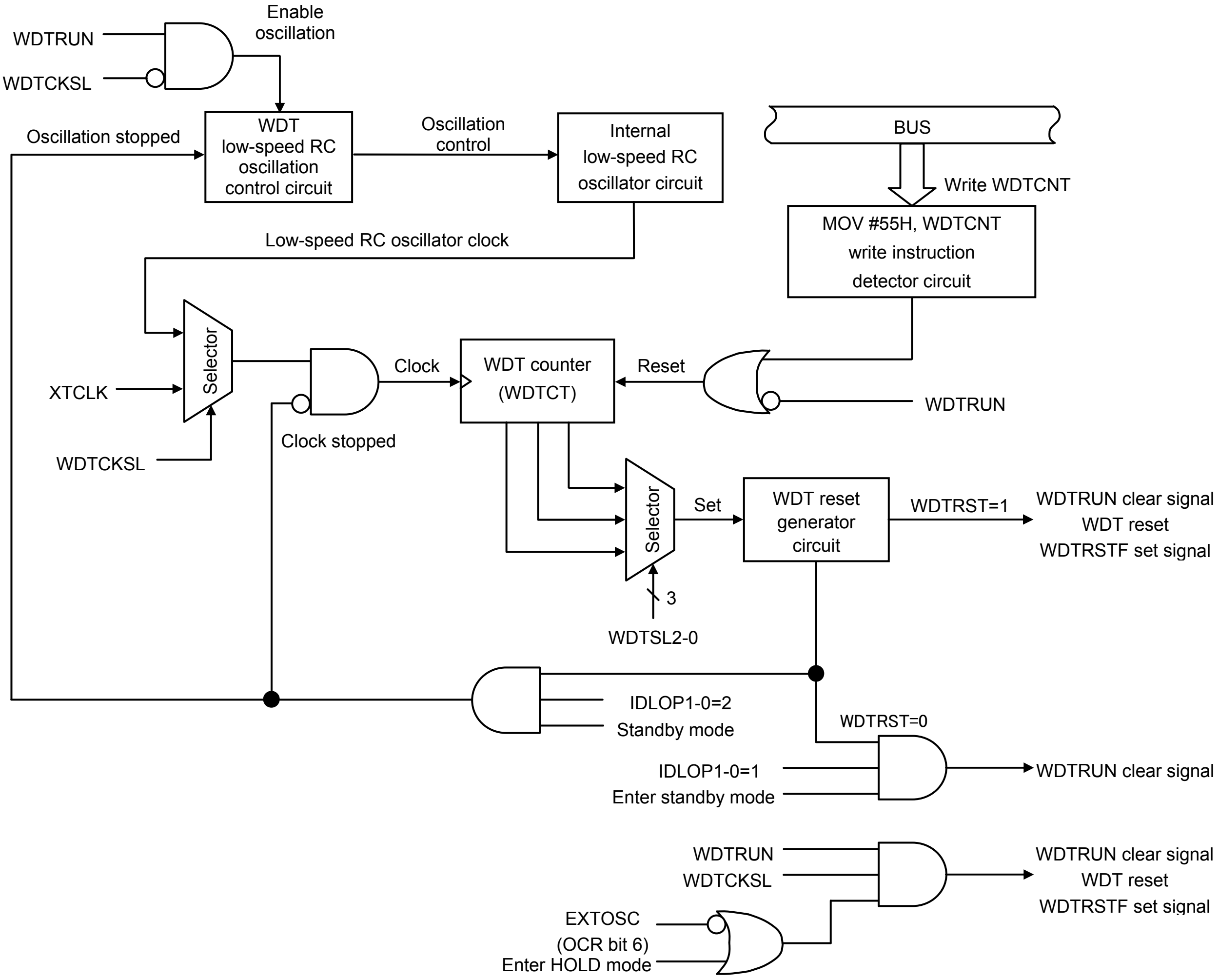


**Figure 4.5.1 Watchdog Timer Block Diagram**

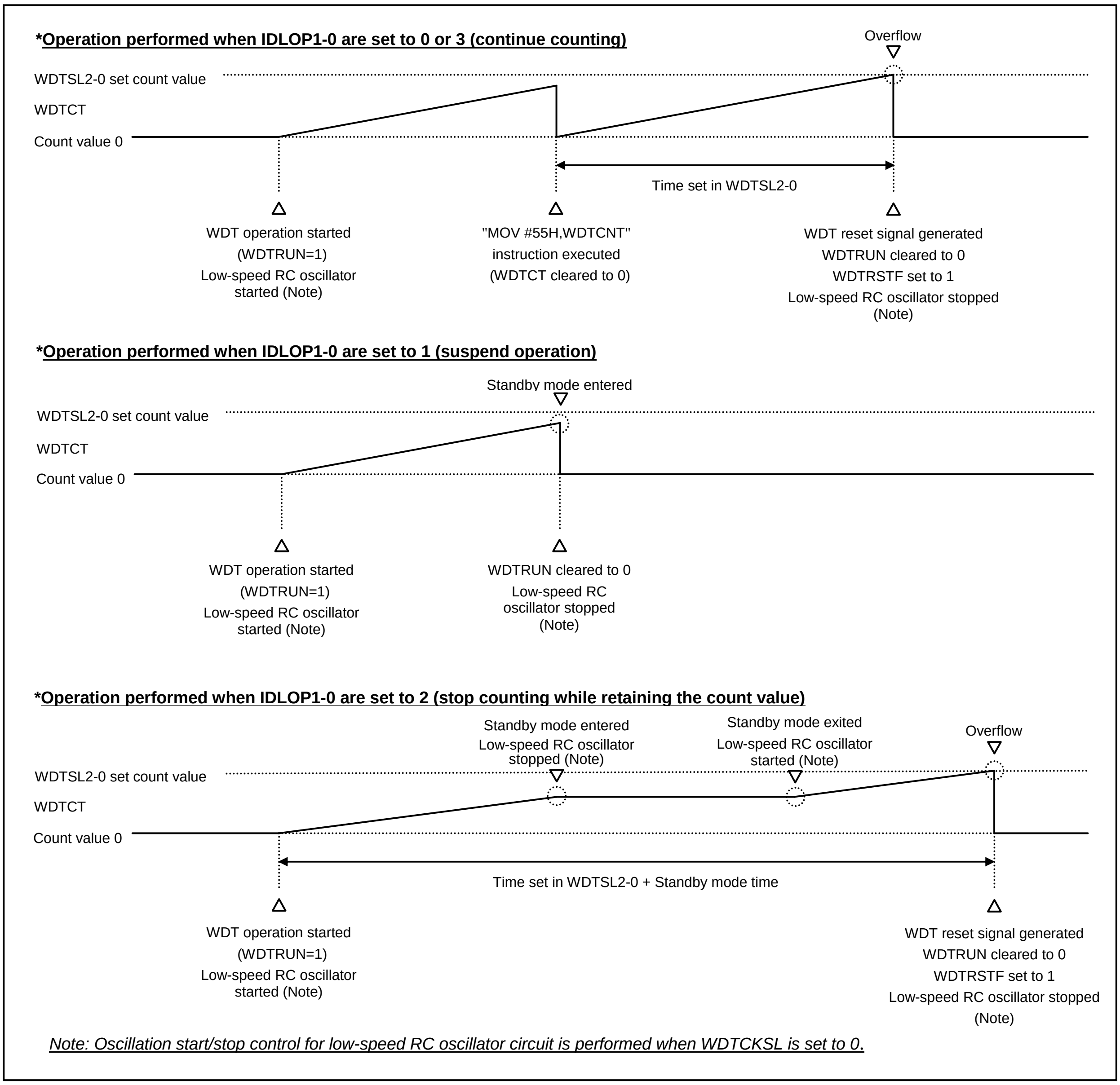

*Operation performed when IDLOP1-0 are set to 0 or 3 (continue counting)
Overflow
WDTSL2-0 set count value
WDTCT
Count value 0
Time set in WDTSL2-0
WDT operation started
(WDTRUN=1)
Low-speed RC oscillator started (Note)
"MOV #55H,WDTCNT" instruction executed
(WDTCT cleared to 0)
WDT reset signal generated
WDTRUN cleared to 0
WDTRSTF set to 1
Low-speed RC oscillator stopped (Note)
*Operation performed when IDLOP1-0 are set to 1 (suspend operation)
Standby mode entered
WDTSL2-0 set count value
WDTCT
Count value 0
WDT operation started
(WDTRUN=1)
Low-speed RC oscillator started (Note)
WDTRUN cleared to 0
Low-speed RC oscillator stopped (Note)
*Operation performed when IDLOP1-0 are set to 2 (stop counting while retaining the count value)
Standby mode entered
Low-speed RC oscillator stopped (Note)
Standby mode exited
Low-speed RC oscillator started (Note)
Overflow
WDTSL2-0 set count value
WDTCT
Count value 0
Time set in WDTSL2-0 + Standby mode time
WDT operation started
(WDTRUN=1)
Low-speed RC oscillator started (Note)
WDT reset signal generated
WDTRUN cleared to 0
WDTRSTF set to 1
Low-speed RC oscillator stopped (Note)
Note: Oscillation start/stop control for low-speed RC oscillator circuit is performed when WDTCKSL is set to 0.

**Figure 4.5.2 Sample Watchdog Timer Operation Waveforms**

## 4.5.4 Related Register

### 4.5.4.1 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 | Operation in Standby Mode |
|---|---|---|
| 0 | 0 | Continue count operation |
| 0 | 1 | Stop operation |
| 1 | 0 | Stop count operation while retaining the count value |
| 1 | 1 | Continue count operation |

* See Figure 4.5.2 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 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. Bit 6 and bits 4 to 0 of WDTCNT are not initialized, however, when a reset is triggered by the WDT.*

*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 tens of μA flows (For details, refer to the latest "SANYO Semiconductors Data Sheet").*

### 4.5.5 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 writes; 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 (POR/LVD) function 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 at regular intervals, 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 option.)

### 4.5.6 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)

- 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 IDLOP0 to 1 or 2. When setting IDLOP1 and IDLOP0 to 0 or 3, several tens of μA of operating current flows at all times because the internal low-speed RC oscillator circuit continues oscillating even in HOLD mode.
- If standby mode is entered when the watchdog timer is running with IDLOP1 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 this configuration (standby mode is on), several tens of μA of operating current flows because the low-speed RC oscillator circuit is active though the watchdog timer is inactive.

  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 detection of the stopping of the subclock oscillation when EXTOSC (OCR, bit 6) is set to 0 or when HOLD mode is entered while 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.

# 4.6 Internal Reset Function

## 4.6.1 Overview

This series of microcontroller incorporates internal reset functions, i.e., 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.).

## 4.6.2 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 is a possibility that 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.

## 4.6.3 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 4.6.1.

- 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 level. See Subsection 4.6.4.

- 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 4.6.1, 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 $C_{RES}$.

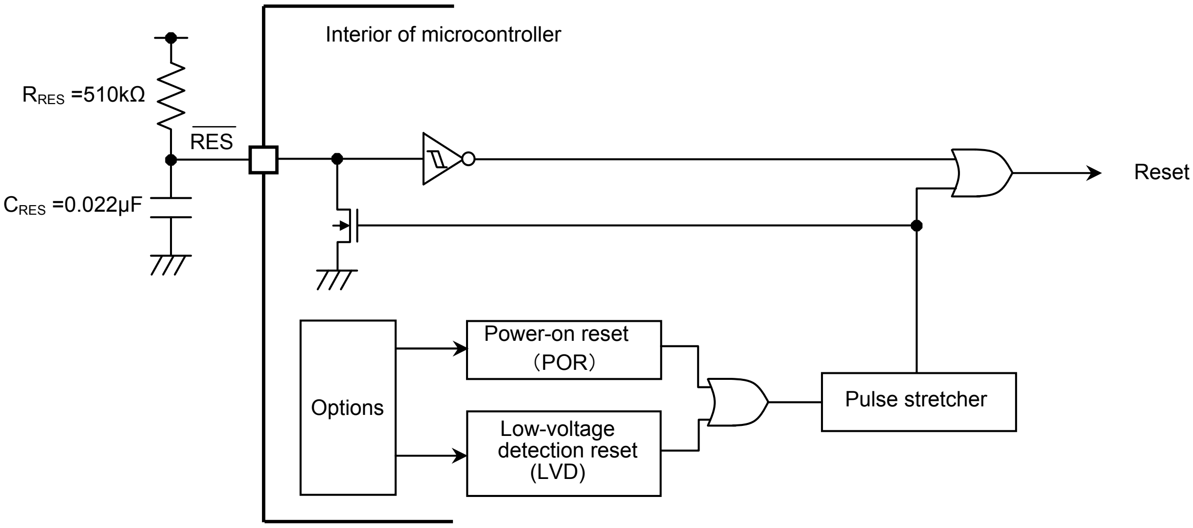


**Figure 4.6.1 Internal Reset Circuit Configuration**

## 4.6.4 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 (*) |
| - | - | "2.57V" | 2.7V to |
| "2.81V" | 3.0V to | "2.87V" | 3.0V to |
| "3.79V" | 4.0V to | "3.86V" | 4.0V to |
| "4.28V" | 4.5V to | "4.35V" | 4.5V to |

* The minimum operating VDD value specifies the approximate lower limit 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 4.6.5 for details.

<2> LVD reset level option

The LVD reset level can be selected from 3 levels only when the LVD level 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 4 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 (2.57V) that will not affect the minimum guaranteed operating voltage.

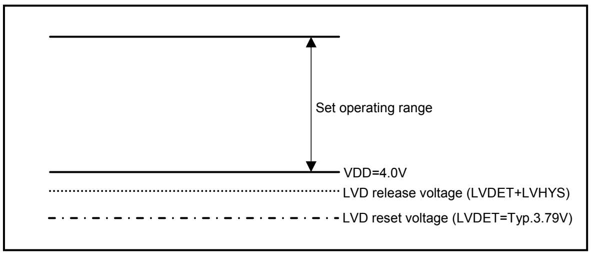
*Note 3: No operating current flows when the POR reset state is released.*

*Note 4: See the notes in paragraph 2) of Subsection 4.6.6 when selecting a POR release level that is lower than the minimum guaranteed operating voltage (2.57V).*

- **Selection example 1**

Selecting the optimum LVD reset level to keep the microcontroller running without resetting it until VDD falls below 4.0V according to the set's requirements

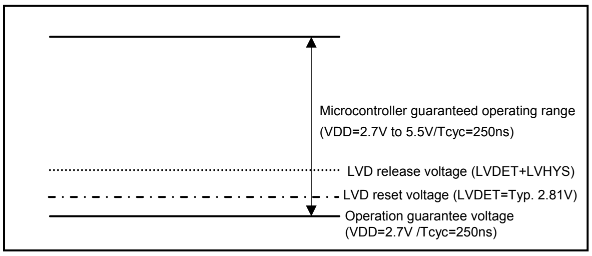
**Set the LVD reset function option to "Enable" and select "3.79V" as the LVD reset level.**


Set operating range
VDD=4.0V
LVD release voltage (LVDET+LVHYS)
LVD reset voltage (LVDET=Typ.3.79V)

- **Selection example 2**

Selecting the optimum LVD reset level that meets the guaranteed operating conditions of VDD = 2.7V/Tcyc = 250 ns

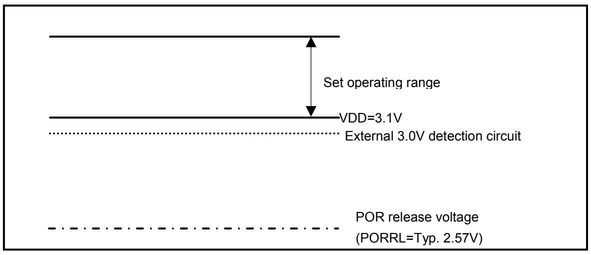
**Set the LVD reset function option to "Enable" and select "2.81V" as the LVD reset level option.**


Microcontroller guaranteed operating range
(VDD=2.7V to 5.5V/Tcyc=250ns)
LVD release voltage (LVDET+LVHYS)
LVD reset voltage (LVDET=Typ. 2.81V)
Operation guarantee voltage
(VDD=2.7V /Tcyc=250ns)

- **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 4.6.7)

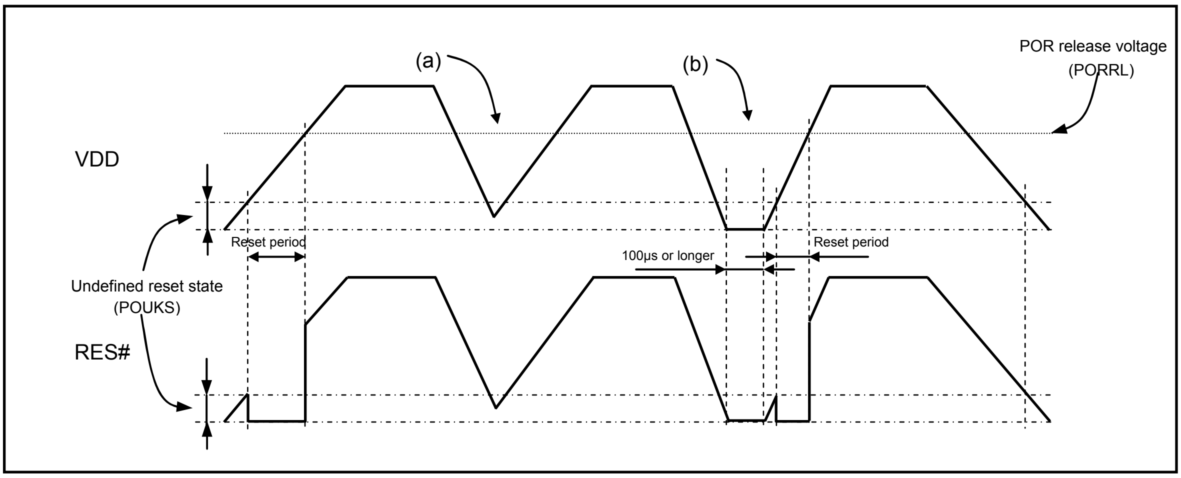
**Set the LVD reset function option to "Disable" and select "2.57V" as the POR release level option.**



*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.”*

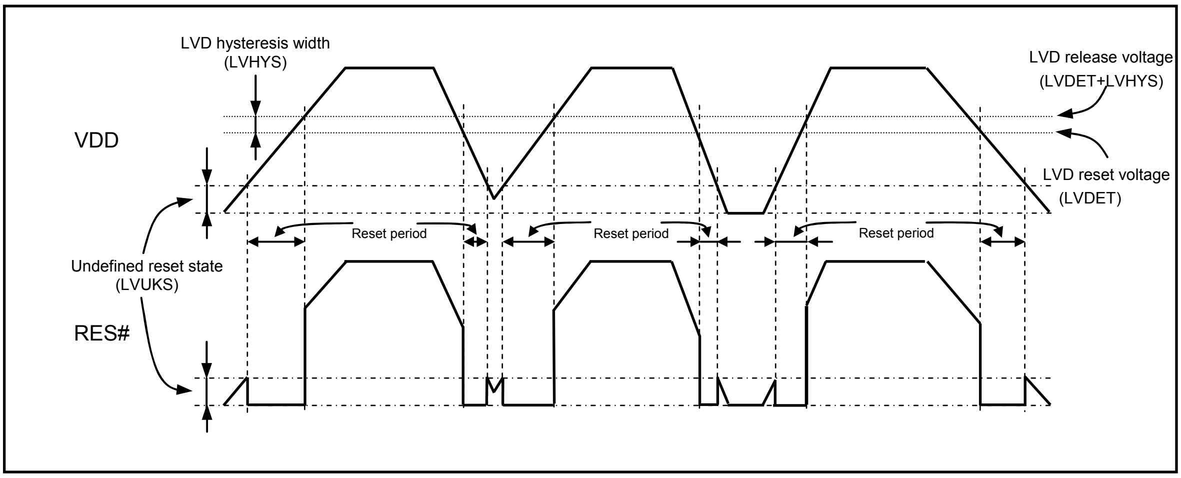
### 4.6.5 Sample Operating Waveforms of the Internal Reset Circuit

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 if 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 POR 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.

### 4.6.6 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 the pull-up resistor $R_{RES}$ alone. Test the circuit extensively under the anticipated power supply conditions to verify that resets are reliably generated.

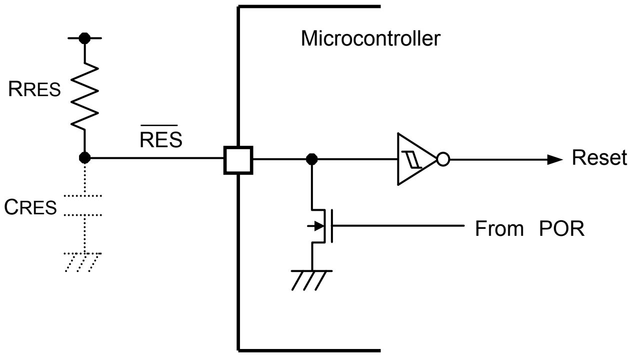


**Figure 4.6.2 Reset Circuit Configuration Using Only the Internal POR Function**

2) When selecting a release voltage level of 2.57V only with the internal POR function

When selecting an internal POR release level of 2.57V, 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.

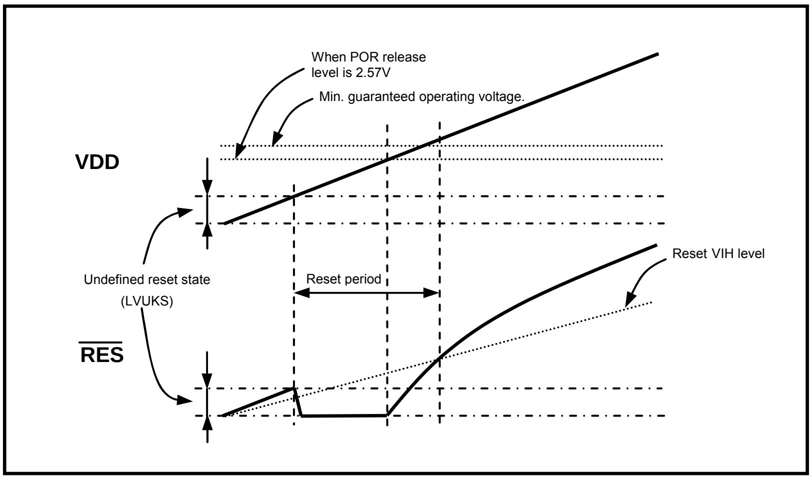


**Figure 4.6.3 Sample Release Level Waveform in Internal POR Only Configuration**

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 4.6.4 (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 4.6.5 or other necessary measures.

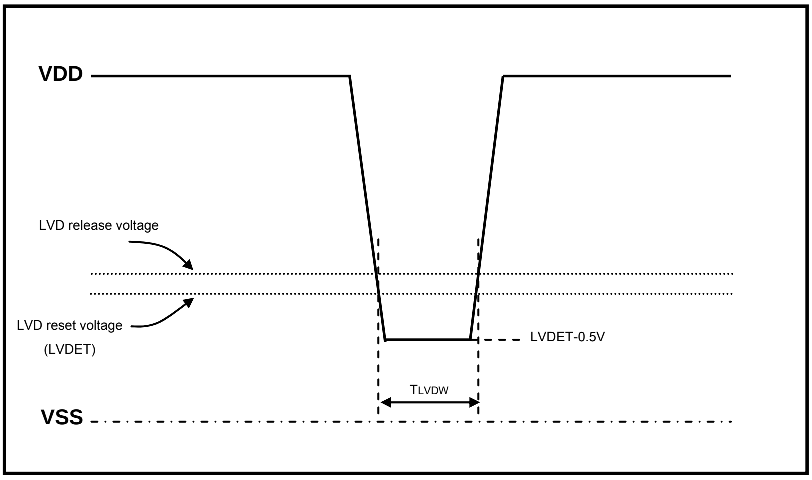


**Figure 4.6.4 Example of Momentary Power Loss or Voltage Fluctuation Waveform**

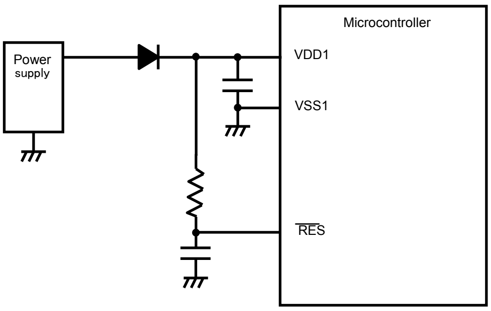


**Figure 4.6.5 Example of Momentary Power Loss/Voltage Fluctuation Countermeasures**

## 4.6.7 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 internal 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.

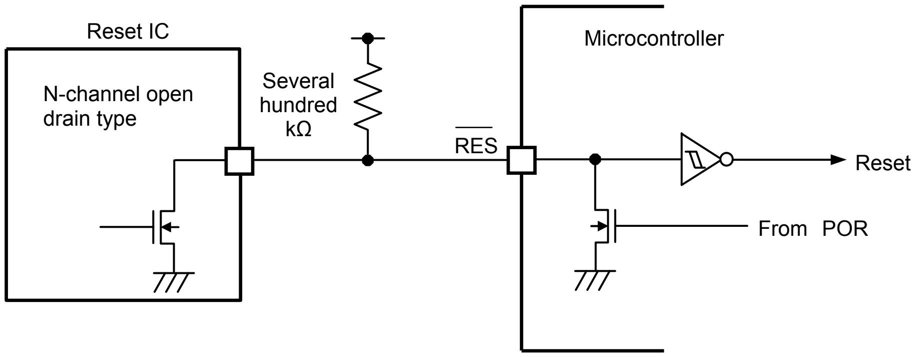


**Figure 4.6.6 Sample Reset Circuit Configuration Using an N-channel Open Drain Type Reset IC**

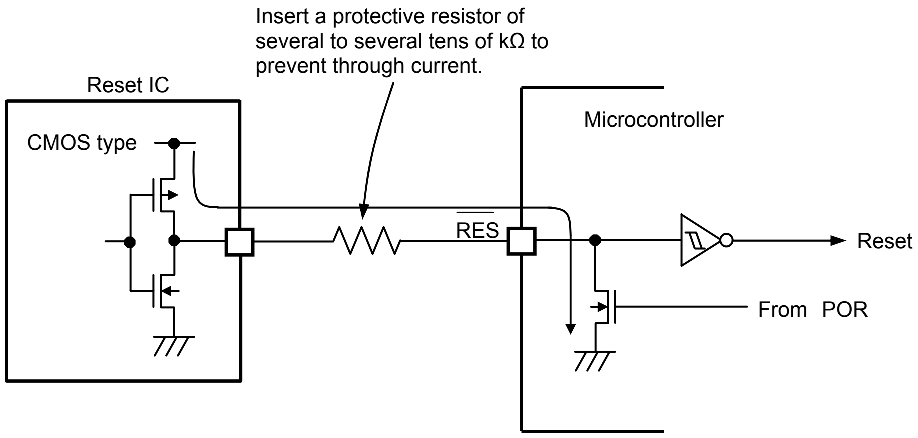

Insert a protective resistor of several to several tens of kΩ to prevent through current.
Reset IC
CMOS type
Microcontroller
$\overline{\text{RES}}$
Reset
From POR

**Figure 4.6.7 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 4.6.7. When configuring an external POR circuit with $C_{RES}$ 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 4.6.8.

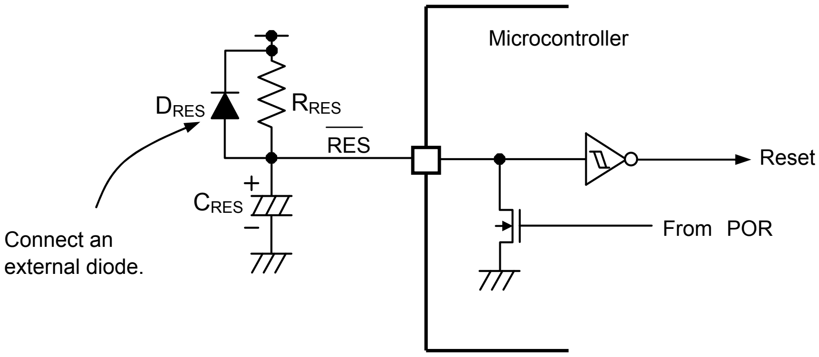


**Figure 4.6.8 Sample External POR Circuit Configuration**

## Internal Reset

# Appendixes

# Table of Contents

## Appendix-I

## Appendix-II

| Address | Initial value | R/W | LC871M00 | Remarks | BIT8 | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0-03FF | XXXX XXXX | R/W | RAM1KB | 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 | | – | CF48ON | DVCKON | DVCKDR | UREFSEL | CF12OFF | 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 | LOWVDEC | PONRES |
| FE0E | 0000 XX00 | R/W | OCR | XT1 and XT2 read at bits 2 and 3 | – | CLKSGL | EXTOSC | CLKCB5 | CLKCB4 | XT2IN | XT1IN | RCSTOP | CFSTOP |
| FE0F | HHHH HHHH | R/W | | | – | | | | | | | | |
| 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 |

| Address | Initial value | R/W | LC871M00 | 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 | PWM0OV | PWM0IE |
| FE25 | HHHH HHXX | R | PWM01P |  | - | - | - | - | - | - | - | PWM1IN | PWM0IN |
| FE26 | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE27 | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE28 | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE29 | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE2A | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE2B | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE2C | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE2D | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE2E | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE2F | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| 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 |
| FE34 | 0000 0000 | R/W | SCON1 |  | - | SI1M1 | SI1M0 | SI1RUN | SI1REC | SI1DIR | SI1OVR | SI1END | SI1IE |
| FE35 | 0000 0000 | R/W | SBUF1 | 9-bit REG | SBUF18 | SBUF17 | SBUF16 | SBUF15 | SBUF14 | SBUF13 | SBUF12 | SBUF11 | SBUF10 |
| FE36 | 0000 0000 | R/W | SBR1 |  | - | SBRG17 | SBRG16 | SBRG15 | SBRG14 | SBRG13 | SBRG12 | SBRG11 | SBRG10 |
| FE37 | 0000 0000 | R/W | SWCON0 | S0XBYT[4:0] is read-only | - | S0WSTP | SWCONB6 | SWCONB5 | S0XBYT4 | S0XBYT3 | S0XBYT2 | S0XBYT1 | S0XBYT0 |
| FE38 | HHHH HHHH |  | None |  | - |  |  |  |  |  |  |  |  |
| FE39 | HHHH HHHH |  | None |  | - |  |  |  |  |  |  |  |  |
| FE3A | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE3B | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |
| FE3C | 0000 0000 | R/W | T45CNT |  | - | T5C1 | T5C0 | T4C1 | T4C0 | T5OV | T5IE | T4OV | T4IE |
| FE3D | HHHH HHHH |  | None |  |  |  |  |  |  |  |  |  |  |

| Address | Initial value | R/W | LC871M00 | 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 | P0IE | 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 | - | - | - | - | DSNKOT | - | 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 000H | R/W | P3 | | - | - | - | - | P34 | P33 | P32 | P31 | - |
| FE4D | HHH0 000H | R/W | P3DDR | | - | - | - | - | P34DDR | P33DDR | P32DDR | P31DDR | - |
| FE4E | 0000 0000 | R/W | I67CR | | - | INT7HEG | INT7LEG | INT7IF | INT7IE | INT6HEG | INT6LEG | INT6IF | INT6IE |
| FE4F | 0000 0000 | R/W | P0FCRU | | - | T7OE | T6OE | SCKOSL5 | SCKOSL4 | CLKOEN | SCKDVC2 | SCKDVC1 | SCKDVC0 |
| 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 |

| Address | Initial value | R/W | LC871M00 | 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 | | – | ST0HCP | ST0LCP | BTIMC1 | BTIMC0 | BUZON | NFSEL | NFON | ST0IN |
| 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 | P0INTE | Port 0 interrupt enable control | | P07INTE | P06INTE | P05INTE | P04INTE | P03INTE | P02INTE | P01INTE | P00INTE |
| FE67 | 0000 0000 | R/W | P2INH | Port 2 input disable control | | P27INH | P26INH | P25INH | P24INH | P23INH | P22INH | P21INH | P20INH |
| 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 | | | RSTFLG | FIX0 | WDTRUN | IDLOP1 | IDLOP0 | WDTSL2 | WDTSL1 | WDTSL0 |
| FE7A | 0000 0000 | R/W | T6R | | – | T6R7 | T6R6 | T6R5 | T6R4 | T6R3 | T6R2 | T6R1 | T6R0 |
| FE7B | 0000 0000 | R/W | T7R | | – | T7R7 | T7R6 | T7R5 | T7R4 | T7R3 | T7R2 | T7R1 | T7R0 |
| FE7C | HHHH HHHH | | None | | | | | | | | | | |

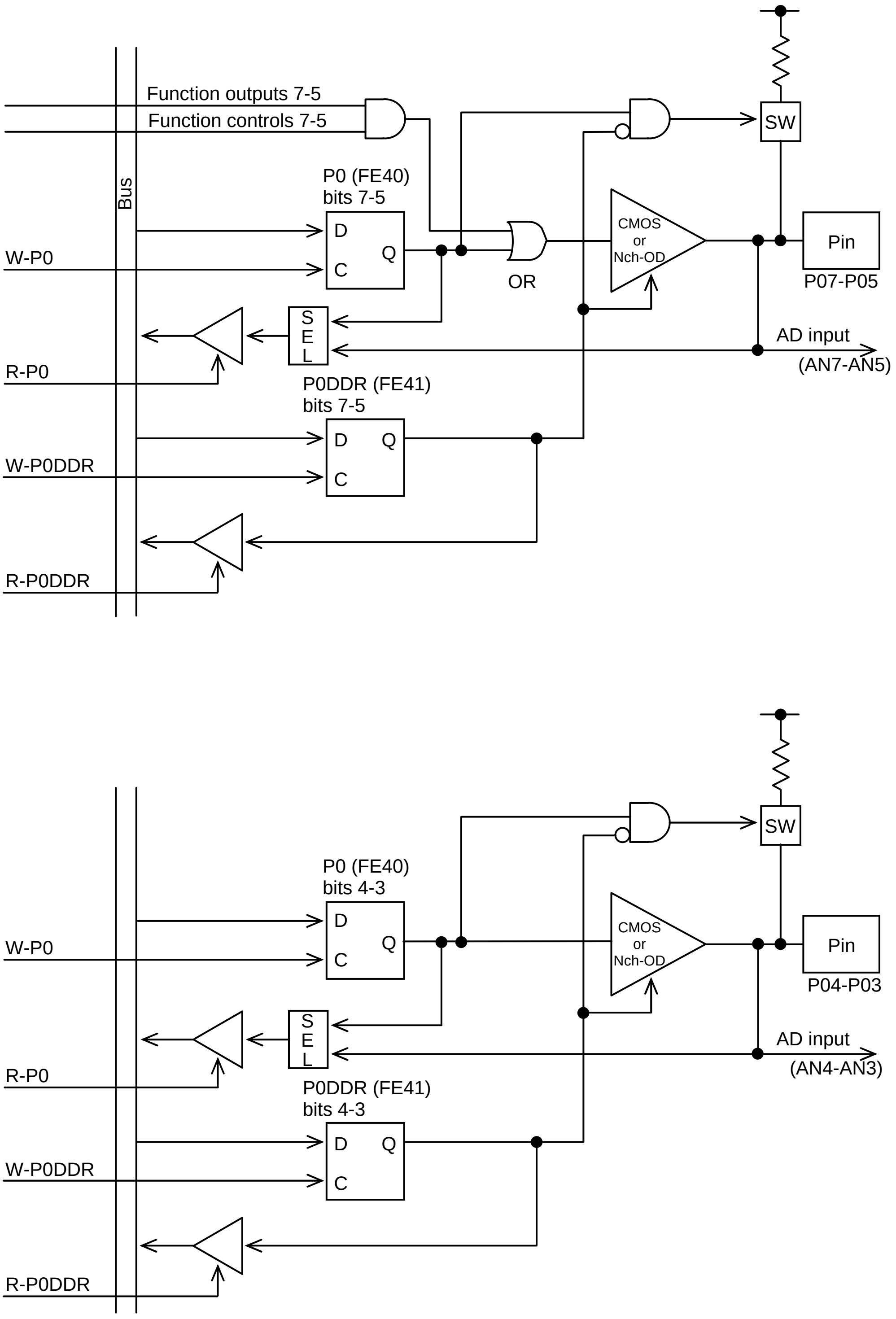
| Address | Initial value | R/W | LC871M00 | Remarks | BIT8 | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FE7D | 0000 0000 | R/W | NKREG | | - | NKEN | NKCMP2 | NKCMP1 | NKCMP0 | NKCOV | NKCAP2 | NKCAP1 | NKCAP0 |
| FE7E | 0000 0000 | R/W | FSR0 | FLASH control (bit 4 is R/O) | - | FSR0B7<br>Fix to 0 | FSR0B6<br>Fix to 0 | FSAERR | FSWOK | INTHIGH | FSLDAT | FSPGL | FSWREQ |
| FE7F | 0000 0000 | R/W | BTCR | Base timer control | - | BTFST | BTON | BTC11 | BTC10 | BTIF1 | BTIE1 | BTIF0 | BTIE0 |
| FE80 | 0000 0000 | R/W | USCTRL | | - | USBON | USBRUN | VD30EN | VD3KIL | IDLFG | IDLEN | DPIEZ | DMIEZ |
| FE81 | 0000 0000 | R/W | USPORT | | - | DDRSOF | P72NDL | USBSIO | SUSPND | DDRDP | DDRDM | PORTDP | PORTDM |
| FE82 | 0000 0000 | R/W | USBINT | | - | BRSFG | BRSEN | BACFG | BACEN | SOFFG | SOFEN | USBINT1 | ENPEN |
| FE83 | 0000 0000 | R/W | EP0INT | | - | AK0FG | AK0EN | NK0FG | NK0EN | ER0FG | ER0EN | ST0FG | ST0EN |
| FE84 | 0000 0000 | R/W | EP1INT | | - | AK1FG | AK1EN | NK1FG | NK1EN | ER1FG | ER1EN | ST1FG | ST1EN |
| FE85 | 0000 0000 | R/W | EP2INT | | - | AK2FG | AK2EN | NK2FG | NK2EN | ER2FG | ER2EN | ST2FG | ST2EN |
| FE86 | 0000 0000 | R/W | EP3INT | | - | AK3FG | AK3EN | NK3FG | NK3EN | ER3FG | ER3EN | ST3FG | ST3EN |
| FE87 | 0000 0000 | R/W | EP4INT | | - | AK4FG | AK4EN | NK4FG | NK4EN | ER4FG | ER4EN | ST4FG | ST4EN |
| FE88 | 0000 0000 | R/W | EP5INT | | | AK5FG | AK5EN | NK5FG | NK5EN | ER5FG | ER5EN | ST5FG | ST5EN |
| FE89 | 0000 0000 | R/W | EP6INT | | | AK6FG | AK6EN | NK6FG | NK6EN | ER6FG | ER6EN | ST6FG | ST6EN |
| FE8A | 0000 0000 | R | FRAMEL | | - | FRAME7 | FRAME6 | FRAME5 | FRAME4 | FRAME3 | FRAME2 | FRAME1 | FRAME0 |
| FE8B | HHHH H000 | R | FRAMEH | | - | - | - | - | - | - | FRAME10 | FRAME9 | FRAME8 |
| FE8C | 0000 0000 | R/W | USBADR | | - | ADREN | ADDR6 | ADDR5 | ADDR4 | ADDR3 | ADDR2 | ADDR1 | ADDR0 |
| FE8D | 0000 0000 | R/W | EPINFO | | - | EPN03 | EPN02 | EPN01 | EPN00 | TKN1 | TKN0 | CTKN1 | CTKN0 |
| FE8E | 0000 0000 | R/W | EP0STA | | - | E0EN | E0TGL | E0OVR | E0STL | E0ACK | E0CSU | E0CST | E0CRW |
| FE8F | H000 0000 | R/W | EP0MP | | - | - | E0MP6 | E0MP5 | E0MP4 | E0MP3 | E0MP2 | E0MP1 | E0MP0 |
| FE90 | H000 0000 | R/W | EP0RX | | - | - | E0RX6 | E0RX5 | E0RX4 | E0RX3 | E0RX2 | E0RX1 | E0RX0 |
| FE91 | H000 0000 | R/W | EP0TX | | - | - | E0TX6 | E0TX5 | E0TX4 | E0TX3 | E0TX2 | E0TX1 | E0TX0 |
| FE92 | 0000 0000 | R/W | EP1STA | | - | E1EN | E1TGL | E1OVR | E1STL | E1ACK | E1DIR | E1ISO | E1BNK |
| FE93 | 0000 0000 | R/W | EP2STA | | - | E2EN | E2TGL | E2OVR | E2STL | E2ACK | E2DIR | E2ISO | E2BNK |
| FE94 | 0000 0000 | R/W | EP3STA | | - | E3EN | E3TGL | E3OVR | E3STL | E3ACK | E3DIR | E3ISO | E3BNK |
| FE95 | 0000 0000 | R/W | EP4STA | | - | E4EN | E4TGL | E4OVR | E4STL | E4ACK | E4DIR | E4ISO | E4BNK |
| FE96 | 0000 0000 | R/W | EP5STA | | | E5EN | E5TGL | E5OVR | E5STL | E5ACK | E5DIR | E5ISO | E5BNK |
| FE97 | 0000 0000 | R/W | EP6STA | | | E6EN | E6TGL | E6OVR | E6STL | E6ACK | E6DIR | E6ISO | E6BNK |
| FE98 | H000 0000 | R/W | EP1CNT | | - | - | E1CN6 | E1CN5 | E1CN4 | E1CN3 | E1CN2 | E1CN1 | E1CN0 |
| FE99 | H000 0000 | R/W | EP1RX | | - | - | E1RX6 | E1RX5 | E1RX4 | E1RX3 | E1RX2 | E1RX1 | E1RX0 |
| FE9A | H000 0000 | R/W | EP2CNT | | - | - | E2CN6 | E2CN5 | E2CN4 | E2CN3 | E2CN2 | E2CN1 | E2CN0 |
| FE9B | H000 0000 | R/W | EP2RX | | - | - | E2RX6 | E2RX5 | E2RX4 | E2RX3 | E2RX2 | E2RX1 | E2RX0 |

| Address | Initial value | R/W | LC871M00 | Remarks | BIT8 | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FE9C | H000 0000 | R/W | EP3CNT | | － | － | E3CN6 | E3CN5 | E3CN4 | E3CN3 | E3CN2 | E3CN1 | E3CN0 |
| FE9D | H000 0000 | R/W | EP3RX | | － | － | E3RX6 | E3RX5 | E3RX4 | E3RX3 | E3RX2 | E3RX1 | E3RX0 |
| FE9E | H000 0000 | R/W | EP4CNT | | － | － | E4CN6 | E4CN5 | E4CN4 | E4CN3 | E4CN2 | E4CN1 | E4CN0 |
| FE9F | H000 0000 | R/W | EP4RX | | － | － | E4RX6 | E4RX5 | E4RX4 | E4RX3 | E4RX2 | E4RX1 | E4RX0 |
| FEA0 | H000 0000 | R/W | EP5CNT | | － | － | E5CN6 | E5CN5 | E5CN4 | E5CN3 | E5CN2 | E5CN1 | E5CN0 |
| FEA1 | HHHH HHHH | | None | | | | | | | | | | |
| FEA2 | H000 0000 | R/W | EP5RX | | － | － | E5RX6 | E5RX5 | E5RX4 | E5RX3 | E5RX2 | E5RX1 | E5RX0 |
| FEA3 | HHHH HHHH | | None | | | | | | | | | | |
| FEA4 | HHHH HHHH | | None | | | | | | | | | | |
| FEA5 | HHHH HHHH | | None | | | | | | | | | | |
| FEA6 | H000 0000 | R/W | EP6CNT | | － | － | E6CN6 | E6CN5 | E6CN4 | E6CN3 | E6CN2 | E6CN1 | E6CN0 |
| FEA7 | HHHH HHHH | | None | | | | | | | | | | |
| FEA8 | H000 0000 | R/W | EP6RX | | － | － | E6RX6 | E6RX5 | E6RX4 | E6RX3 | E6RX2 | E6RX1 | E6RX0 |
| FEA9 | HHHH HHHH | | None | | | | | | | | | | |
| FEAA | HHHH HHHH | | None | | | | | | | | | | |
| FEAB | HHHH H000 | | EPBMOD | | － | － | － | － | － | － | EPBM2 | EPBM1 | EPBM0 |
| FEAC | 0000 0000 | R/W | TESTR0 | | － | DPLLTEST | CMPTST | CMPKIL | TTXCLK | TTXREQ | TADR2 | TADR1 | TADR0 |
| FEAD | 0000 0000 | R | TESTR1 | | － | TDAT7 | TDAT6 | TDAT5 | TDAT4 | TDAT3 | TDAT2 | TDAT1 | TDAT0 |
| FEAE | 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 | HHHH HHHH | | None | | | | | | | | | | |
| FEB7 | HHHH HHHH | | None | | | | | | | | | | |
| FEB8 | HHHH HHHH | | None | | | | | | | | | | |
| FEB9 | HHHH HHHH | | None | | | | | | | | | | |
| FEBA | HHHH HHHH | | None | | | | | | | | | | |

| Address | Initial value | R/W | LC871M00 | Remarks | BIT8 | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FEBB | HHHH HHHH | | None | | | | | | | | | | |
| FEBC | HHHH HHHH | | None | | | | | | | | | | |
| FEBD | HHHH HHHH | | None | | | | | | | | | | |
| FEBE | HHHH HHHH | | None | | | | | | | | | | |
| FEBF | HHHH HHHH | | None | | | | | | | | | | |
| FEC0 | HHHH HHHH | | None | | | | | | | | | | |
| FEC1 | HHHH HHHH | | None | | | | | | | | | | |
| FEC2 | HHHH HHHH | | None | | | | | | | | | | |
| FEC3 | HHHH HHHH | | None | | | | | | | | | | |
| FEC4 | HHHH HHHH | | None | | | | | | | | | | |
| FEC5 | HHHH HHHH | | None | | | | | | | | | | |
| FEC6 | HHHH HHHH | | None | | | | | | | | | | |
| FEC7 | HHHH HHHH | | None | | | | | | | | | | |
| FEC8 | HHHH HHHH | | None | | | | | | | | | | |
| FEC9 | HHHH HHHH | | None | | | | | | | | | | |
| FECA | 0000 00H0 | R/W | SCCNT0 | Smart card UART | - | SCTDDR | SCTSTP | SCTRUN | SCTERR | SCTEND | SCTENDIE | SCTEMPTY | SCTEMPTYIE |
| FECB | 0000 0000 | R/W | SCCNT1 | Smart card UART | - | SCERCE | SCCHRL | SCRRUN | SCROVER | SCRFERR | SCRPERR | SCREND | SCRENDIE |
| FECC | H000 0000 | R/W | SCMOD | Smart card UART | - | - | SCSCK | SCDIV1 | SCDIV0 | SCPODD | SCPEN | SCDIR | SCINV |
| FECD | 0000 0000 | R/W | SCTBUF | Smart card UART | - | SCTBUF7 | SCTBUF6 | SCTBUF5 | SCTBUF4 | SCTBUF3 | SCTBUF2 | SCTBUF1 | SCTBUF0 |
| FECE | 0000 0000 | R | SCRBUF | Smart card UART | - | SCRBUF7 | SCRBUF6 | SCRBUF5 | SCRBUF4 | SCRBUF3 | SCRBUF2 | SCRBUF1 | SCRBUF0 |
| FECF | 0000 0000 | R/W | SCBRG | Smart card UART | - | SCBRG7 | SCBRG6 | SCBRG5 | SCBRG4 | SCBRG3 | SCBRG2 | SCBRG1 | SCBRG0 |
| FED0 | 0000 0000 | R/W | UCON0 | | - | UBRSEL | STRDET | RECRUN | STPERR | U0B3 | RBIT8 | RECEND | RECIE |
| FED1 | 0000 0000 | R/W | UCON1 | | - | TRUN | 8/9BIT | TDDR | TCMOS | 8/7BIT | TBIT8 | TEPTY | TRNSIE |
| FED2 | 0000 0000 | R/W | UBR | | - | UBRG7 | UBRG6 | UBRG5 | UBRG4 | UBRG3 | UBRG2 | UBRG1 | UBRG0 |
| FED3 | 0000 0000 | R/W | TBUF | | - | 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 | S4BYTH5 | S4BYTH4 | S4BYTH3 | S4BYTH2 | S4BYTH1 | S4BYTH0 |
| FED8 | 0000 0000 | R/W | CRCL | | - | CRC7 | CRC6 | CRC5 | CRC4 | CRC3 | CRC2 | CRC1 | CRC0 |
| FED9 | 0000 0000 | R/W | CRCH | | - | CRC15 | CRC14 | CRC13 | CRC12 | CRC11 | CRC10 | CRC9 | CRC8 |

| Address | Initial value | R/W | LC871M00 | Remarks | BIT8 | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FEDA | 0000 0000 | R/W | CRCCNT | | - | CRCON | CRCLRZ | CRCRD | 1/OSEL | 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 | | | | | | | | | | |

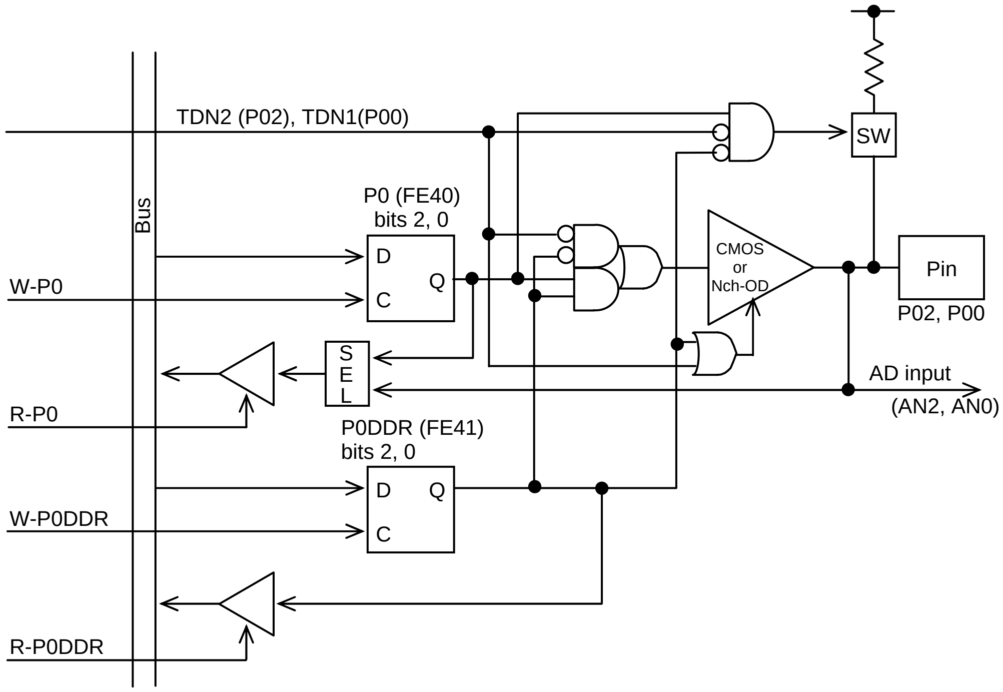
| Address | Initial value | R/W | LC871M00 | Remarks | BIT8 | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FEFA | HHHH HHHH | | None | | | | | | | | | | |
| FEFB | HHHH HHHH | | None | | | | | | | | | | |
| FEFC | HHHH HHHH | | None | | | | | | | | | | |
| FEFD | HHHH HHHH | | None | | | | | | | | | | |
| FEFE | HHHH HHHH | | None | | | | | | | | | | |
| FEFF | 0000 0000 | R/W | BUFCNT | B7:AD channel select,<br>B0-2,4-5:buffer output control | - | ADCHSEL4 | BUFCNT6 | BUFP1 | BUFP0 | BUFCNT3 | BUFN2 | BUFN1 | BUFN0 |

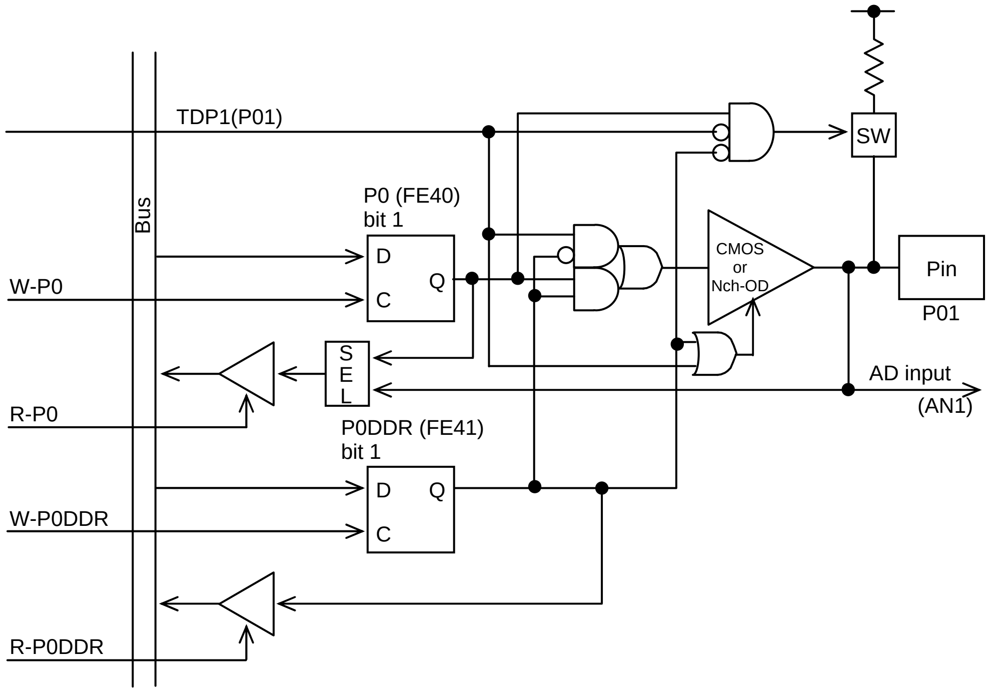


**Port 0 Block Diagram**

**Option: Output type (CMOS or N-channel OD) selectable in 1-bit units**

## Port block diagram



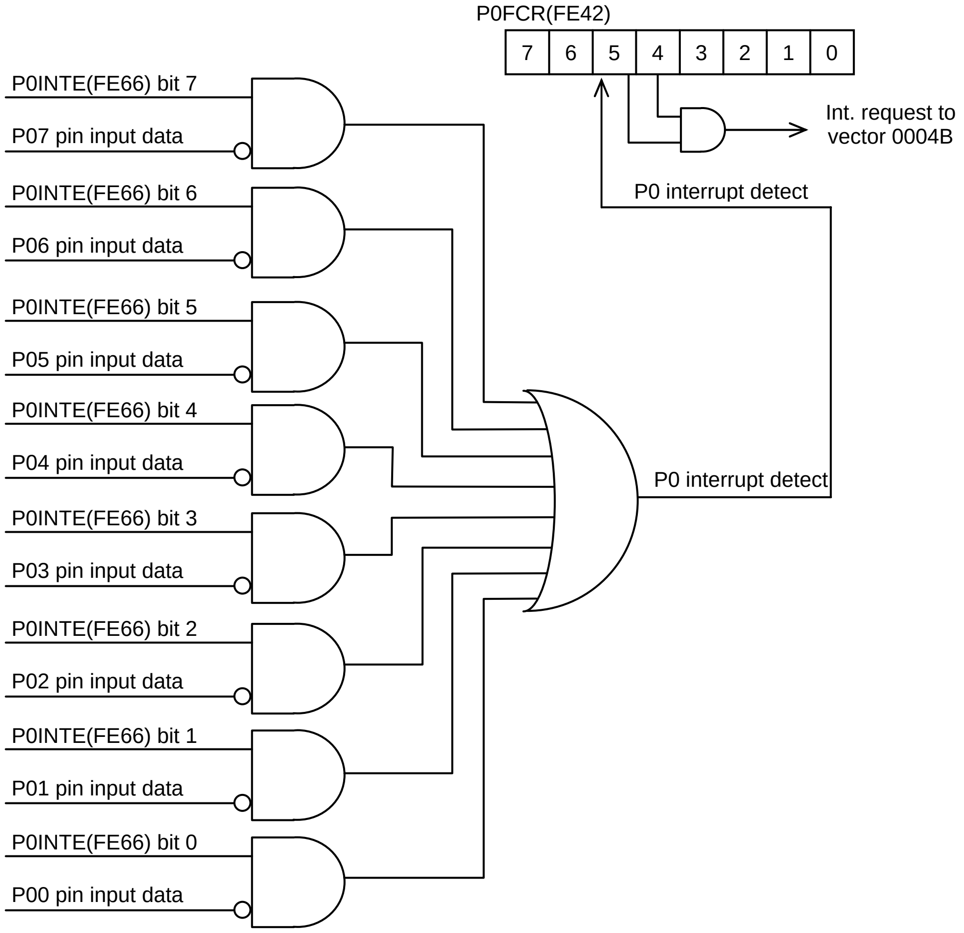


**Port 0 Block Diagram**

**Option: Output type (CMOS or N-channel OD) selectable in 1-bit units**

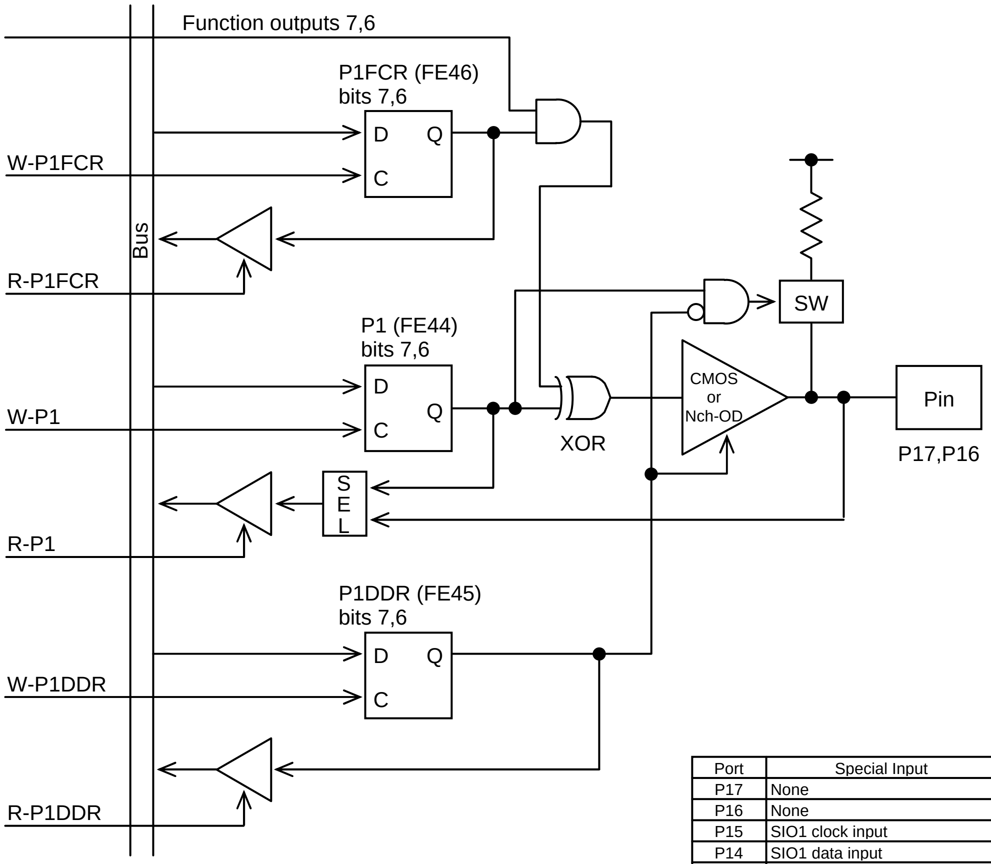
| Port | Multiplexed port pin function |
|---|---|
| P07 | Timer 7 toggle output |
| P06 | Timer 6 toggle output |
| P05 | Clock output (system clock/subclock selectable) |
| P02 | High-current N-channel driver |
| P01 | High-current P-channel driver |
| P00 | High-current N-channel driver |

**Table of Port 0 Multiplexed Functions**



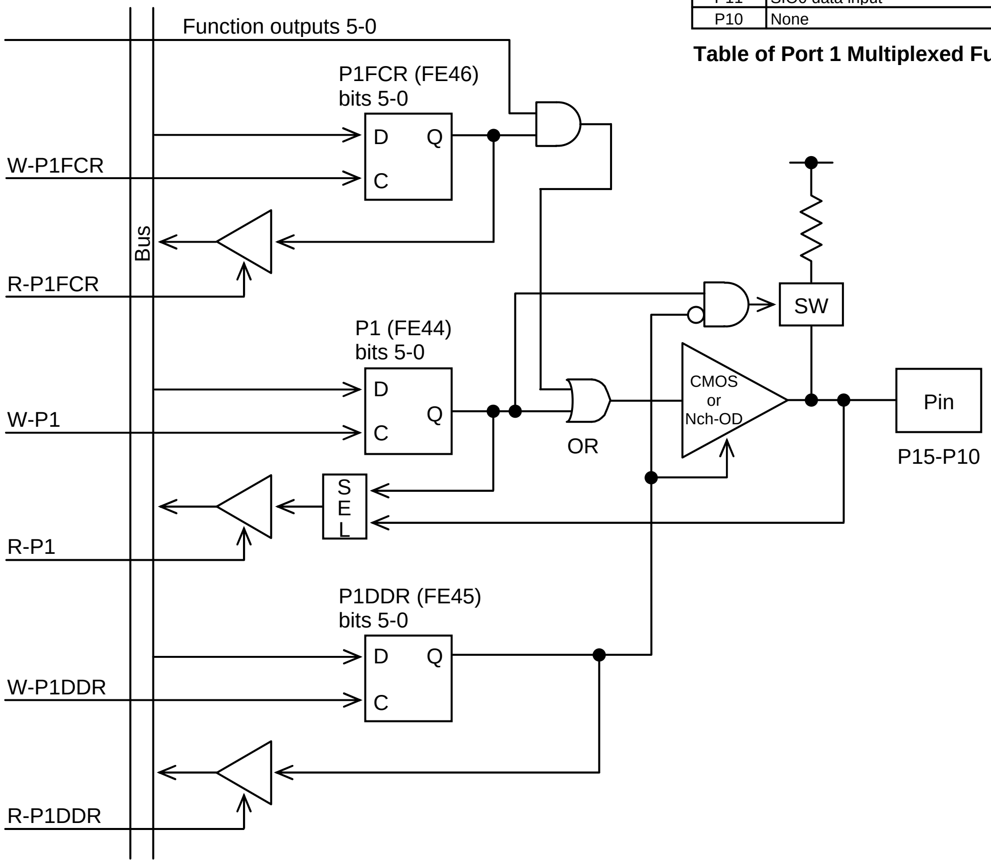
**Port 0 (Interrupt) Block Diagram**

## Port block diagram



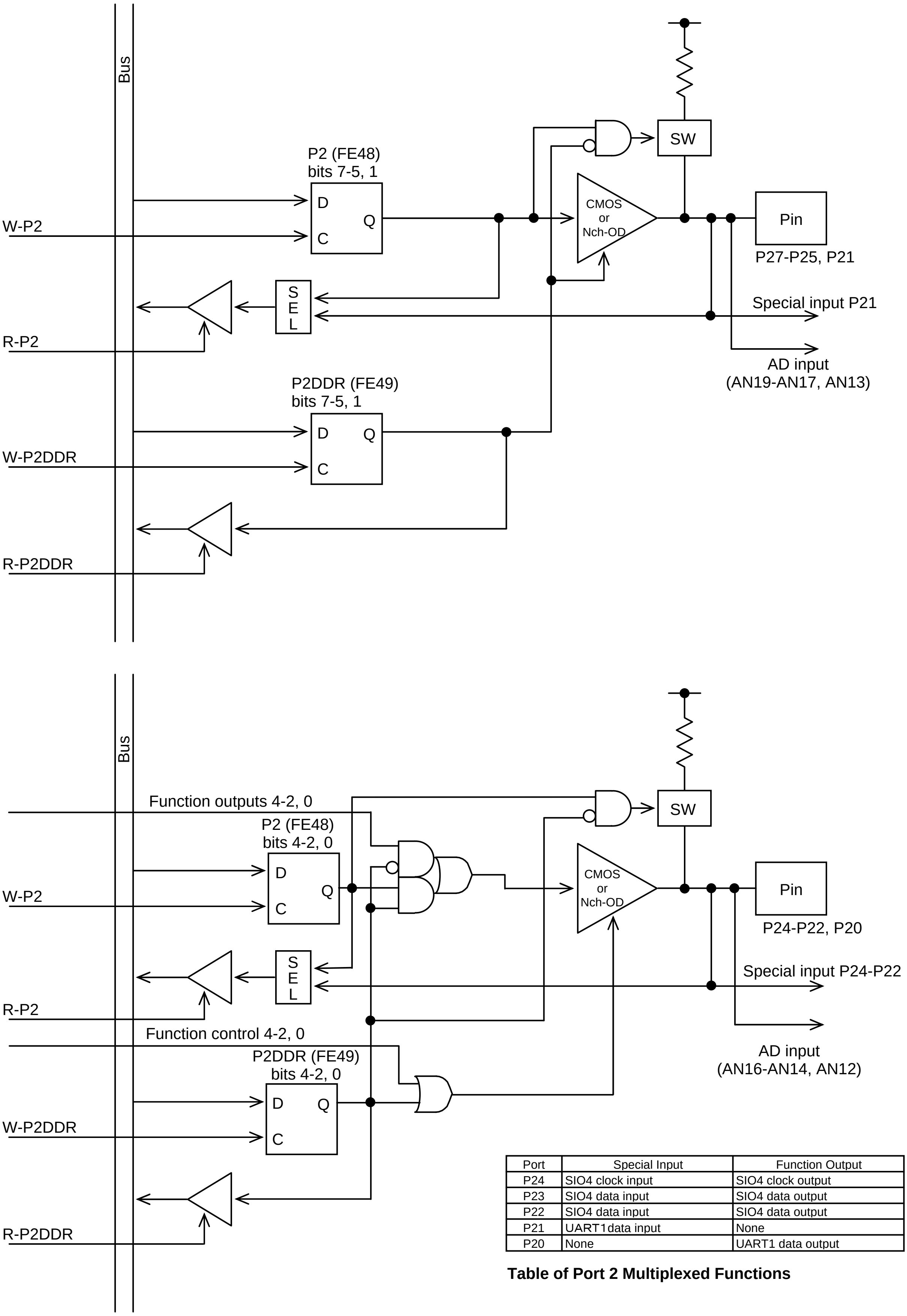
| Port | Special Input | Function Output |
|---|---|---|
| P17 | None | Timer 1HPWM output |
| P16 | None | Timer 1LPWM output |
| P15 | SIO1 clock input | SIO1 clock output |
| P14 | SIO1 data input | SIO1 data output |
| P13 | None | SIO1 data output |
| P12 | SIO0 clock input | SIO0 clock output |
| P11 | SIO0 data input | SIO0 data output |
| P10 | None | SIO0 data output |

**Table of Port 1 Multiplexed Functions**



**Port 1 Block Diagram**

**Option: Output type (CMOS or N-channel OD) selectable in 1-bit units**



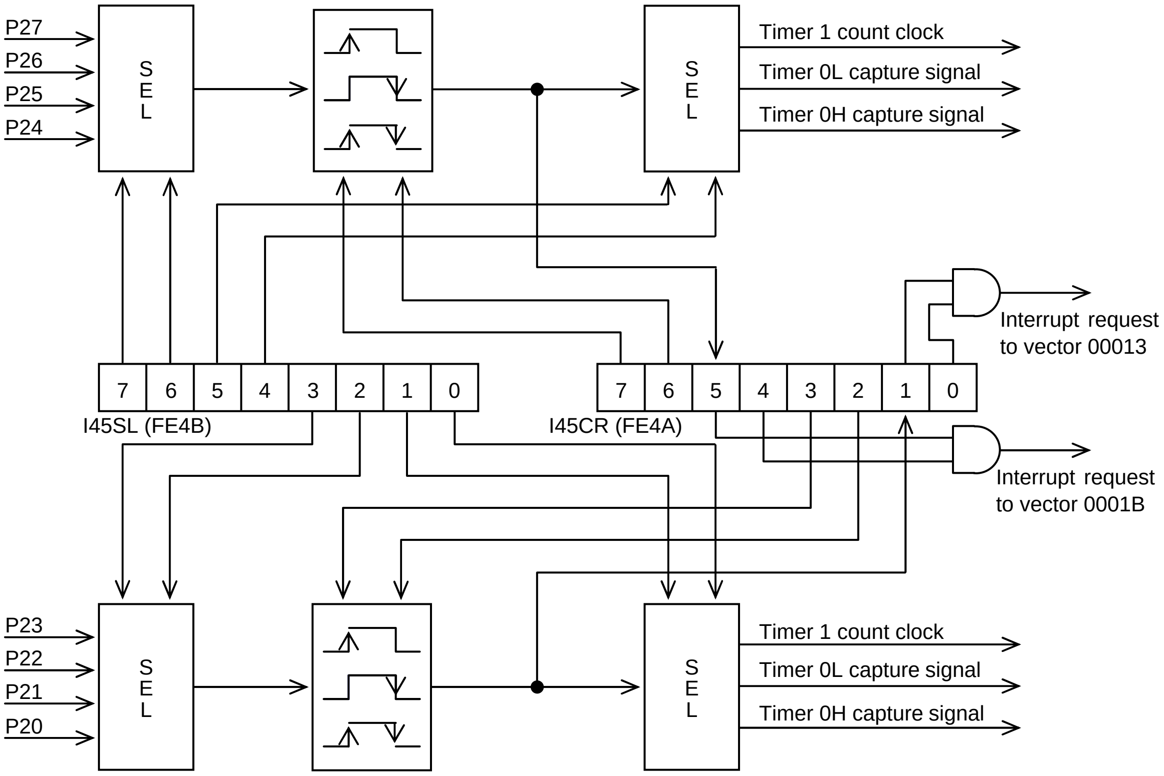
| Port | Special Input | Function Output |
|---|---|---|
| P24 | SIO4 clock input | SIO4 clock output |
| P23 | SIO4 data input | SIO4 data output |
| P22 | SIO4 data input | SIO4 data output |
| P21 | UART1data input | None |
| P20 | None | UART1 data output |

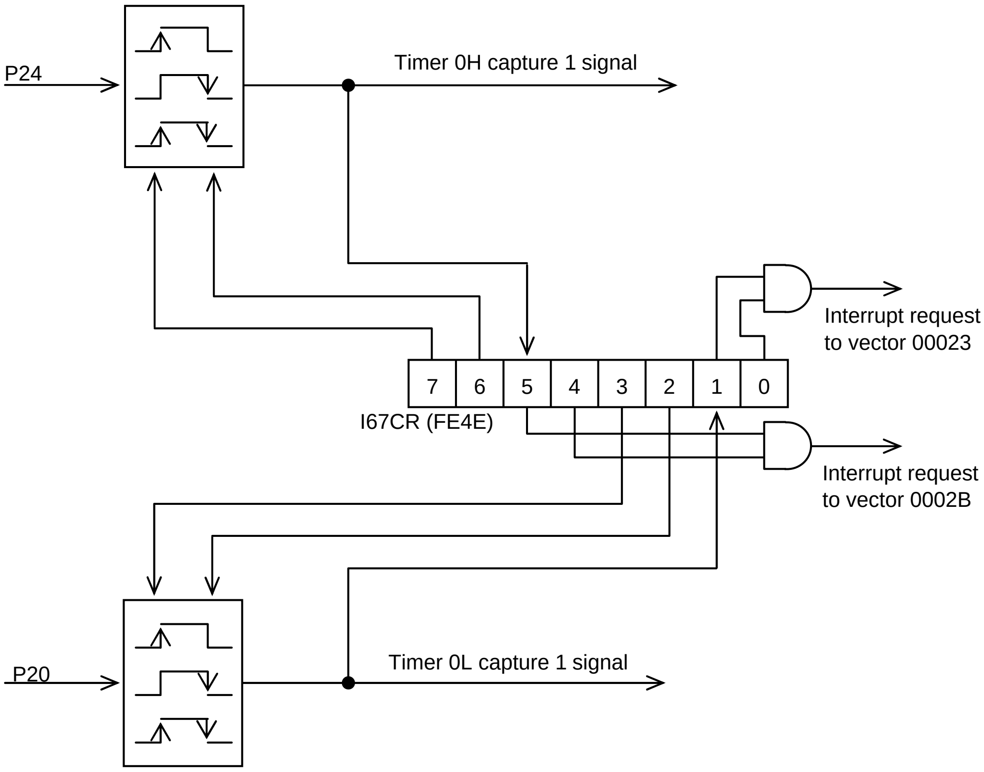
**Table of Port 2 Multiplexed Functions**

**Port 2 (Pin) Block Diagram**

**Option: Output type (CMOS or N-channel OD) selectable in 1-bit units**

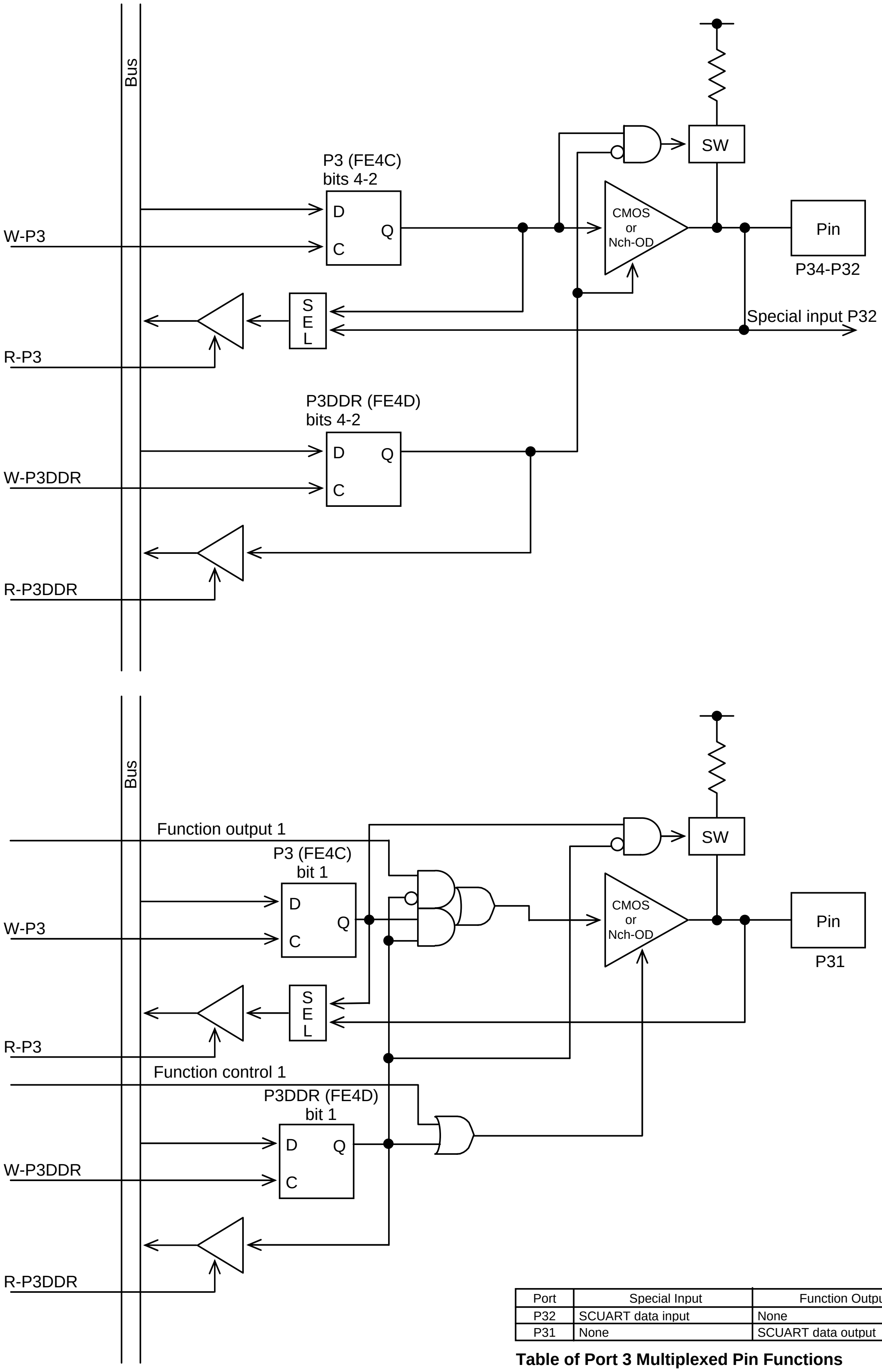
**Port block diagram**





P24
Timer 0H capture 1 signal
7 6 5 4 3 2 1 0
I67CR (FE4E)
Interrupt request to vector 00023
Interrupt request to vector 0002B
P20
Timer 0L capture 1 signal

**Port 2 (Interrupt) Block Diagram**



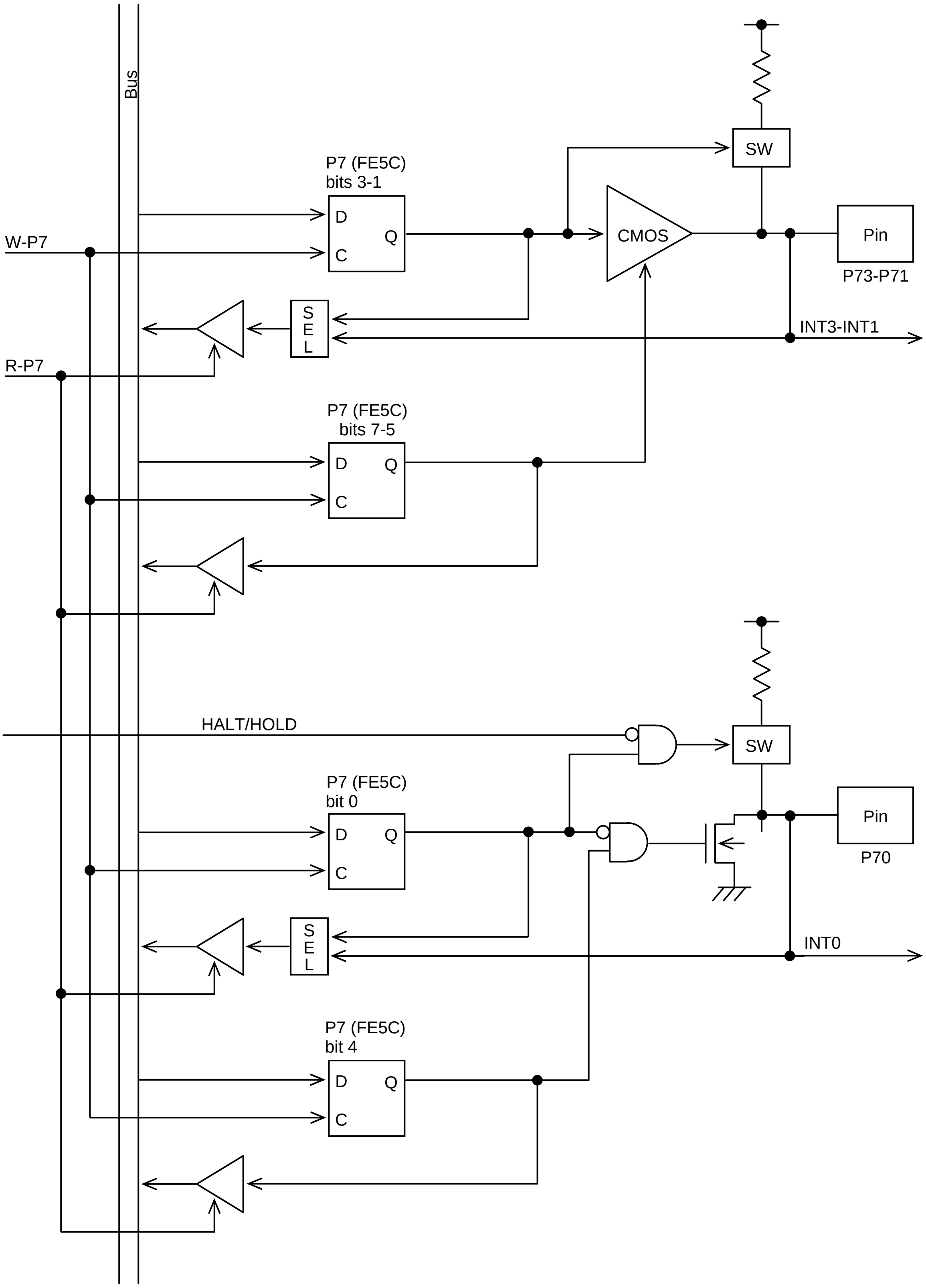
| Port | Special Input | Function Output |
|---|---|---|
| P32 | SCUART data input | None |
| P31 | None | SCUART data output |

**Table of Port 3 Multiplexed Pin Functions**

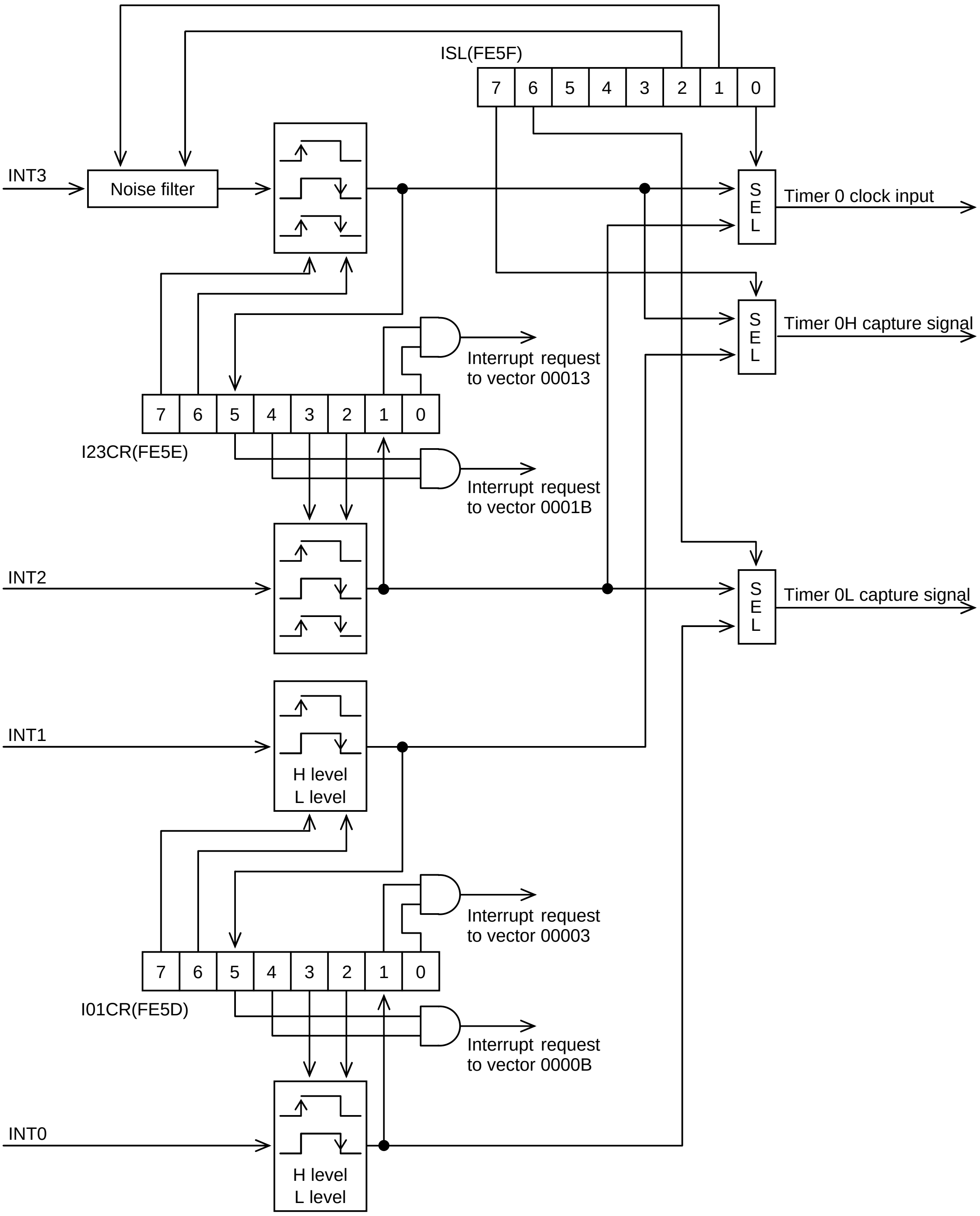
**Port 3 Block Diagram**

**Option: Output type (CMOS or N-channel OD) selectable in 1-bit units**

## Port block diagram

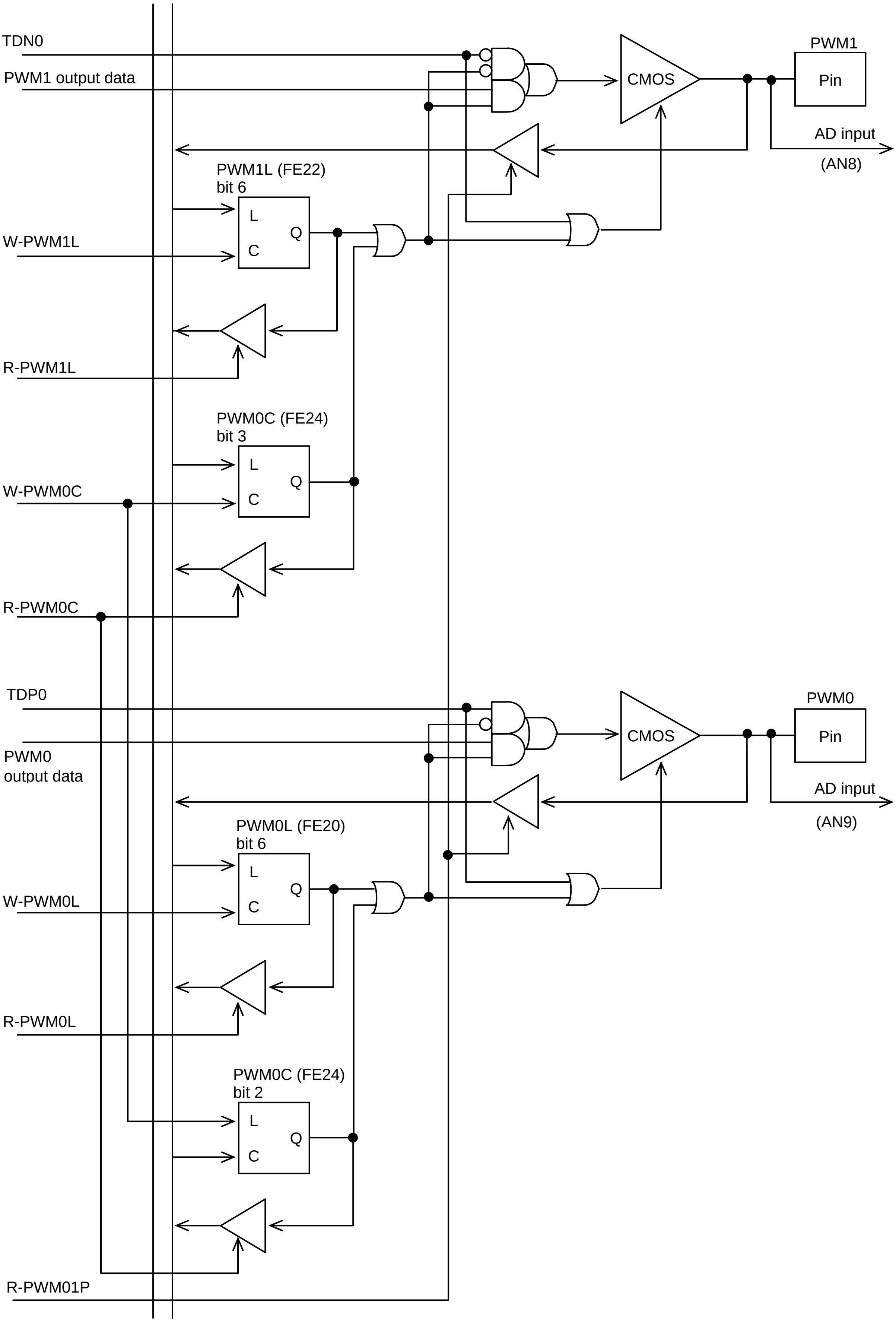


**Port 7 (Pin) Block Diagram**
**Option: None**



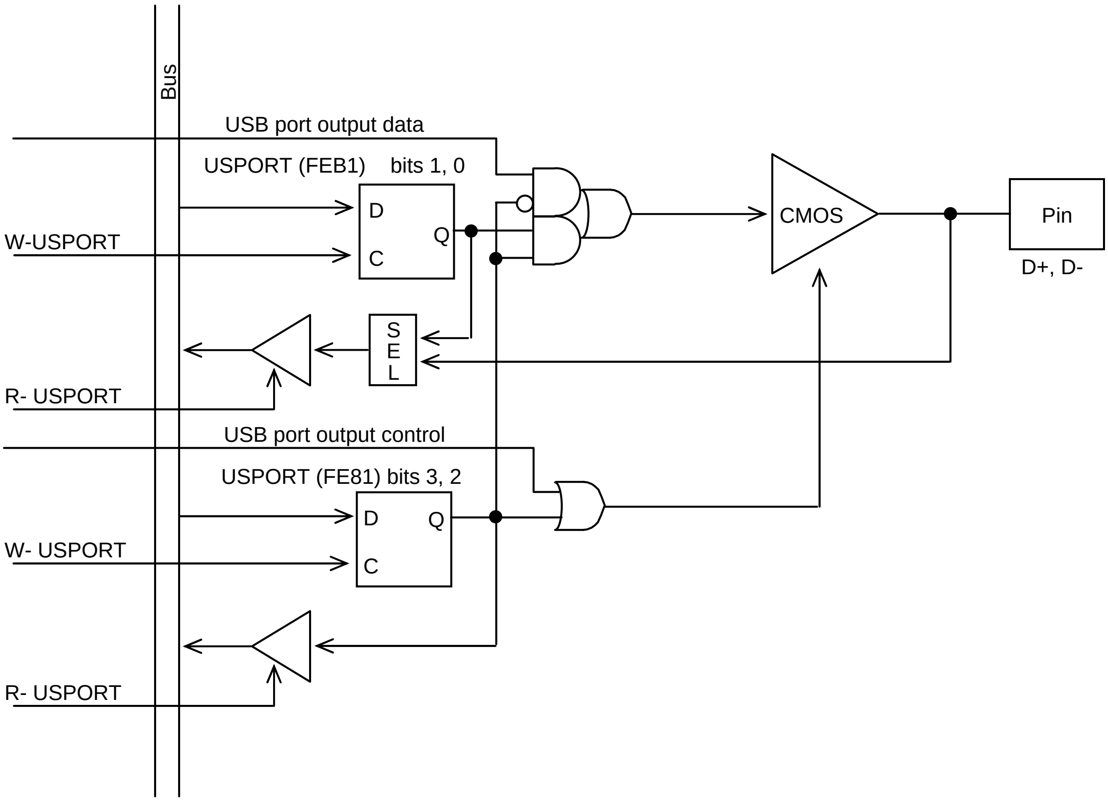
**Port 7 (Interrupt) Block Diagram**

## Port block diagram



**PWM0, PWM1 Block Diagram**

**Option: None**



**USB Port (Pin) Block Diagram**

**Port 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.*

**LC871M00 SERIES USER'S MANUAL**

**Rev : 1.00 April 26, 2012**

**ON Semiconductor**
**Digital Solution Division**
**Microcontroller & Flash Business Unit**