**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | H8SX |
| Core Size | 32-Bit Single-Core |
| Speed | 50MHz |
| Connectivity | SCI, SmartCard |
| Peripherals | DMA, PWM, WDT |
| Number of I/O | 82 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| EEPROM Size | - |
| RAM Size | 24K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -20°C ~ 75°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 120-LQFP |
| Supplier Device Package | 120-LQFP (14x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/ds61650cw50fpv |

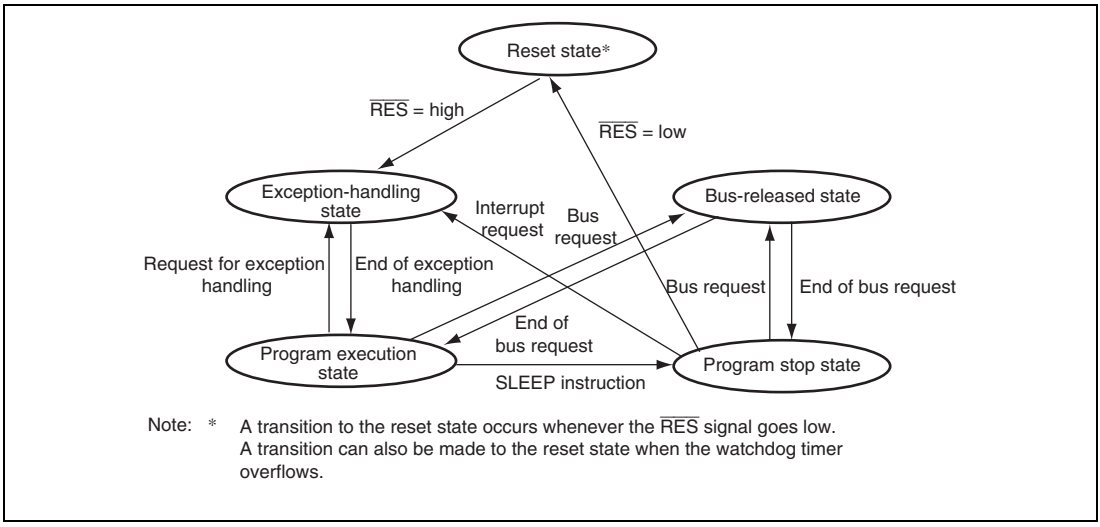


**Figure 2.16 State Transitions**

## 4.7 Instruction Exception Handling

There are three instructions that cause exception handling: trap instruction, sleep instruction, and illegal instruction.

### 4.7.1 Trap Instruction Exception Handling

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state. The trap instruction exception handling is as follows:

1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the vector number specified in the TRAPA instruction is generated, the start address of the exception service routine is loaded from the vector table to PC, and program execution starts from that address.

A start address is read from the vector table corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.8 shows the state of CCR and EXR after execution of trap instruction exception handling.

**Table 4.8 Status of CCR and EXR after Trap Instruction Exception Handling**

| | CCR | | EXR | |
|---|---|---|---|---|
| **Interrupt Control Mode** | **I** | **UI** | **T** | **I2 to I0** |
| 0 | 1 | — | — | — |
| 2 | 1 | — | 0 | — |

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains the previous value.

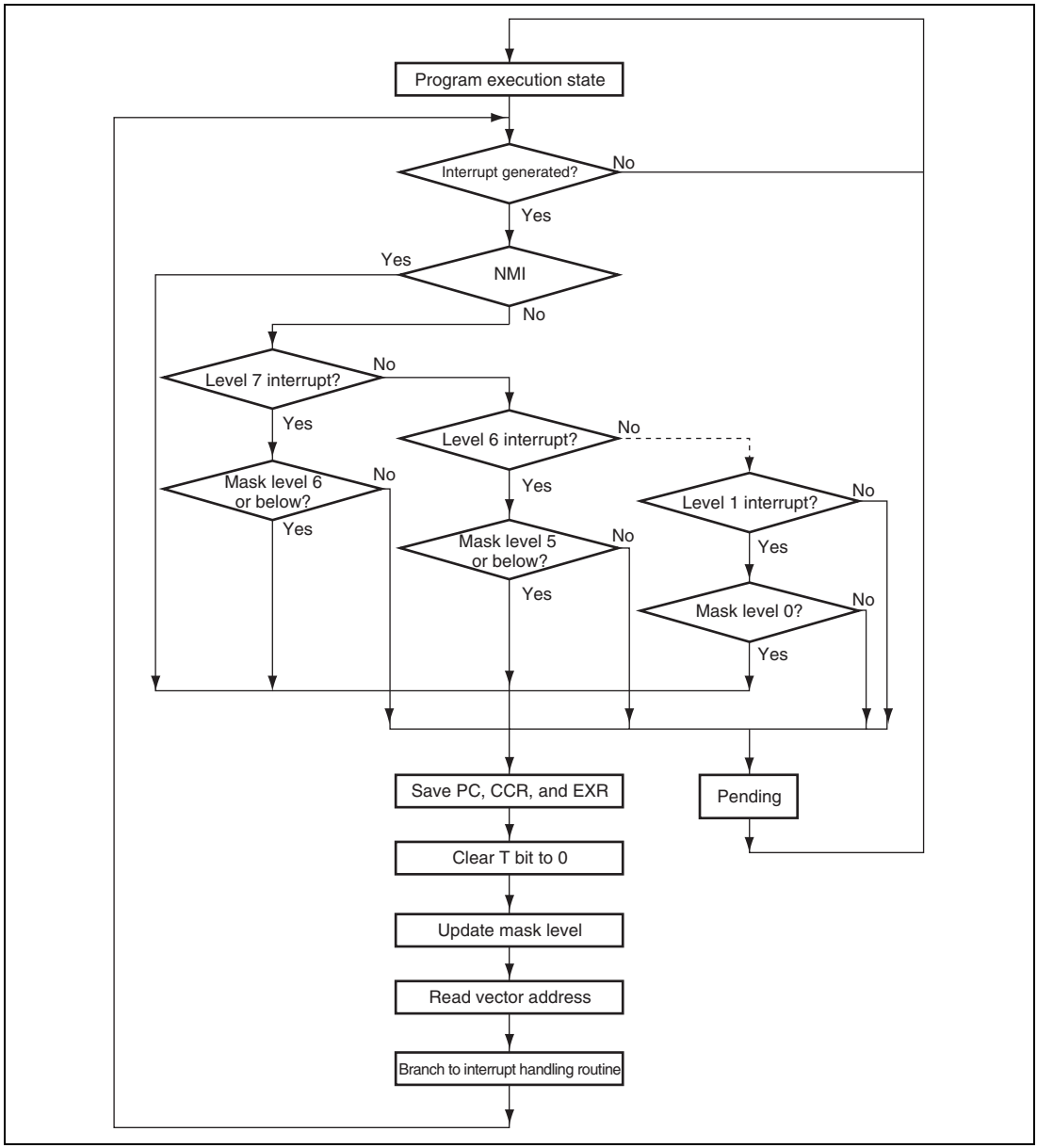


**Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2**

### 8.3.2 Port Function Control Register 1 (PFCR1)

PFCR1 selects the $\overline{CS}$ output pins.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Name | CS7SA | CS7SB | CS6SA | CS6SB | CS5SA | CS5SB | CS4SA | CS4SB |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CS7SA* | 0 | R/W | $\overline{CS7}$ Output Pin Select |
| 6 | CS7SB* | 0 | R/W | Selects the output pin for $\overline{CS7}$ when $\overline{CS7}$ output is enabled (CS7E = 1)<br>00: Specifies pin PB3 as $\overline{CS7}$-A output<br>01: Specifies pin PB1 as $\overline{CS7}$-B output<br>10: Specifies pin PF7 as $\overline{CS7}$-C output<br>11: Setting prohibited |
| 5 | CS6SA* | 0 | R/W | $\overline{CS6}$ Output Pin Select |
| 4 | CS6SB* | 0 | R/W | Selects the output pin for $\overline{CS6}$ when $\overline{CS6}$ output is enabled (CS6E = 1)<br>00: Specifies pin PB2 as $\overline{CS6}$-A output<br>01: Specifies pin PB1 as $\overline{CS6}$-B output<br>10: Specifies pin PF7 as $\overline{CS6}$-C output<br>11: Specifies pin PF6 as $\overline{CS6}$-D output |
| 3 | CS5SA* | 0 | R/W | $\overline{CS5}$ Output Pin Select |
| 2 | CS5SB* | 0 | R/W | Selects the output pin for $\overline{CS5}$ when $\overline{CS5}$ output is enabled (CS5E = 1)<br>00: Specifies pin PB1 as $\overline{CS5}$-A output<br>01: Specifies pin PB0 as $\overline{CS5}$-B output<br>10: Specifies pin PF7 as $\overline{CS5}$-C output<br>11: Specifies pin PF5 as $\overline{CS5}$-D output |

**Table 9.19 TIOR_4**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOB3** | **Bit 6 IOB2** | **Bit 5 IOB1** | **Bit 4 IOB0** | **TGRB_4 Function** | **TIOCB4 Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output disabled |
| 0 | 0 | 0 | 1 | | Initial output is 0 output<br>0 output at compare match |
| 0 | 0 | 1 | 0 | | Initial output is 0 output<br>1 output at compare match |
| 0 | 0 | 1 | 1 | | Initial output is 0 output<br>Toggle output at compare match |
| 0 | 1 | 0 | 0 | | Output disabled |
| 0 | 1 | 0 | 1 | | Initial output is 1 output<br>0 output at compare match |
| 0 | 1 | 1 | 0 | | Initial output is 1 output<br>1 output at compare match |
| 0 | 1 | 1 | 1 | | Initial output is 1 output<br>Toggle output at compare match |
| 1 | 0 | 0 | 0 | Input capture register | Capture input source is TIOCB4 pin<br>Input capture at rising edge |
| 1 | 0 | 0 | 1 | | Capture input source is TIOCB4 pin<br>Input capture at falling edge |
| 1 | 0 | 1 | x | | Capture input source is TIOCB4 pin<br>Input capture at both edges |
| 1 | 1 | x | x | | Capture input source is TGRC_3 compare match/input capture<br>Input capture at generation of TGRC_3 compare match/input capture |

[Legend]

X: Don't care

**Table 9.20 TIOR_5**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOB3** | **Bit 6 IOB2** | **Bit 5 IOB1** | **Bit 4 IOB0** | **TGRB_5 Function** | **TIOCB5 Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output disabled |
| 0 | 0 | 0 | 1 | | Initial output is 0 output<br>0 output at compare match |
| 0 | 0 | 1 | 0 | | Initial output is 0 output<br>1 output at compare match |
| 0 | 0 | 1 | 1 | | Initial output is 0 output<br>Toggle output at compare match |
| 0 | 1 | 0 | 0 | | Output disabled |
| 0 | 1 | 0 | 1 | | Initial output is 1 output<br>0 output at compare match |
| 0 | 1 | 1 | 0 | | Initial output is 1 output<br>1 output at compare match |
| 0 | 1 | 1 | 1 | | Initial output is 1 output<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register | Capture input source is TIOCB5 pin<br>Input capture at rising edge |
| 1 | x | 0 | 1 | | Capture input source is TIOCB5 pin<br>Input capture at falling edge |
| 1 | x | 1 | x | | Capture input source is TIOCB5 pin<br>Input capture at both edges |

[Legend]

X: Don't care

**Table 9.23 TIOR_1**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOA3** | **Bit 2 IOA2** | **Bit 1 IOA1** | **Bit 0 IOA0** | **TGRA_1 Function** | **TIOCA1 Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output disabled |
| 0 | 0 | 0 | 1 | | Initial output is 0 output<br>0 output at compare match |
| 0 | 0 | 1 | 0 | | Initial output is 0 output<br>1 output at compare match |
| 0 | 0 | 1 | 1 | | Initial output is 0 output<br>Toggle output at compare match |
| 0 | 1 | 0 | 0 | | Output disabled |
| 0 | 1 | 0 | 1 | | Initial output is 1 output<br>0 output at compare match |
| 0 | 1 | 1 | 0 | | Initial output is 1 output<br>1 output at compare match |
| 0 | 1 | 1 | 1 | | Initial output is 1 output<br>Toggle output at compare match |
| 1 | 0 | 0 | 0 | Input capture register | Capture input source is TIOCA1 pin<br>Input capture at rising edge |
| 1 | 0 | 0 | 1 | | Capture input source is TIOCA1 pin<br>Input capture at falling edge |
| 1 | 0 | 1 | X | | Capture input source is TIOCA1 pin<br>Input capture at both edges |
| 1 | 1 | X | X | | Capture input source is TGRA_0 compare match/input capture<br>Input capture at generation of channel 0/TGRA_0 compare match/input capture |

[Legend]

X: Don't care

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 0 | TGFA | 0 | R/(W)* | Input Capture/Output Compare Flag A<br>Status flag that indicates the occurrence of TGRA input capture or compare match.<br>[Setting conditions]<br>• When TCNT = TGRA while TGRA is functioning as output compare register<br>• When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register<br>[Clearing conditions]<br>• When DTC is activated by a TGIA interrupt while the DISEL bit in MRB of DTC is 0<br>• When 0 is written to TGFA after reading TGFA = 1 (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.) |

Note: * Only 0 can be written to clear the flag.

### (b) Example of input capture operation

Figure 9.9 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

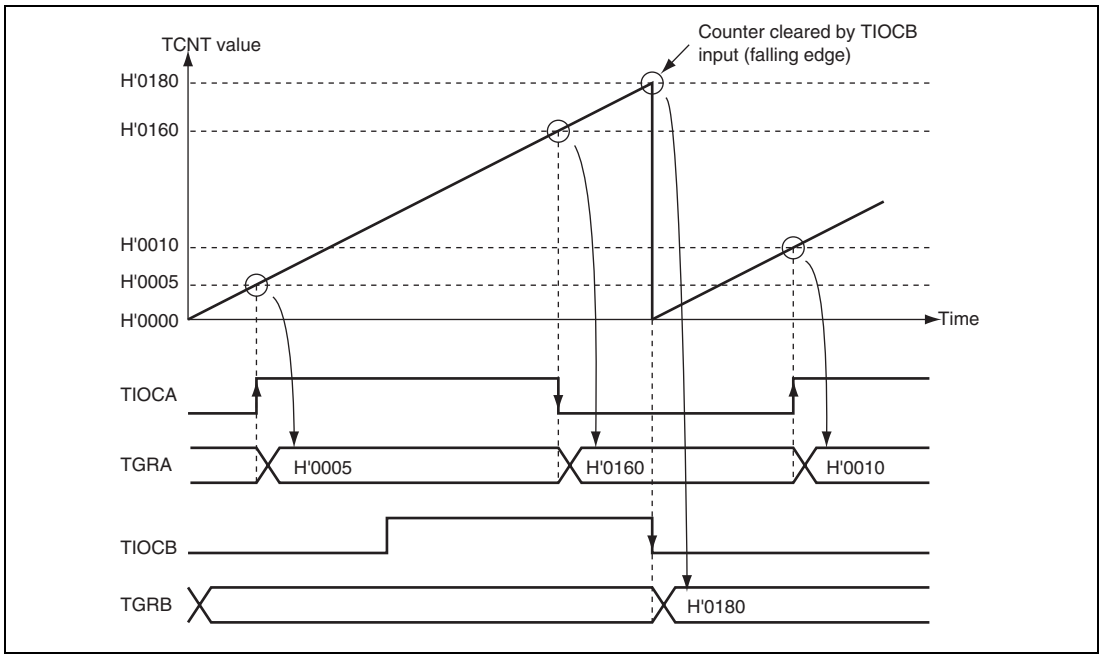


**Figure 9.9 Example of Input Capture Operation**

### 9.4.3 Buffer Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or a compare match register.

Table 9.29 shows the register combinations used in buffer operation.

**Table 9.29 Register Combinations in Buffer Operation**

| Channel | Timer General Register | Buffer Register |
|---|---|---|
| 0 | TGRA_0 | TGRC_0 |
| | TGRB_0 | TGRD_0 |
| 3 | TGRA_3 | TGRC_3 |
| | TGRB_3 | TGRD_3 |

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 9.12.

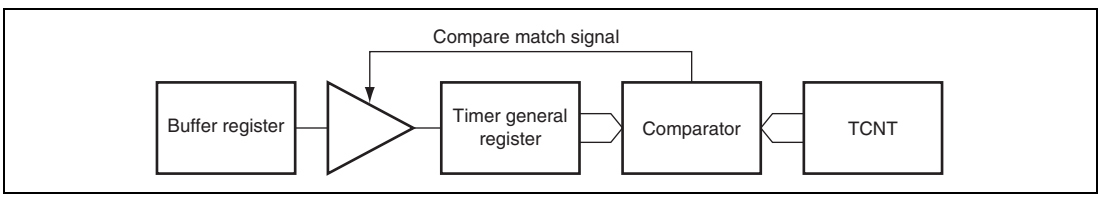


**Figure 9.12 Compare Match Buffer Operation**

### 9.9.11 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 9.53 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

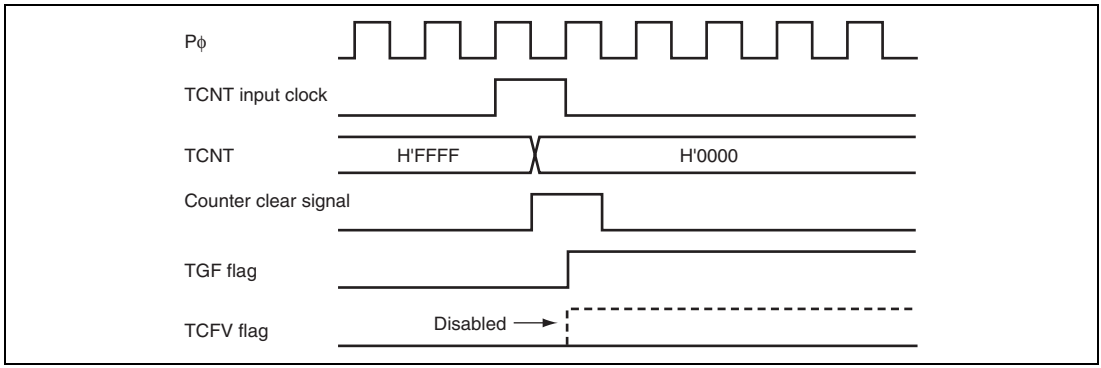


**Figure 9.53 Conflict between Overflow and Counter Clearing**

### 9.9.12 Conflict between TCNT Write and Overflow/Underflow

If an overflow/underflow occurs due to increment/decrement in the T2 state of a TCNT write cycle, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 9.54 shows the operation timing when there is conflict between TCNT write and overflow.

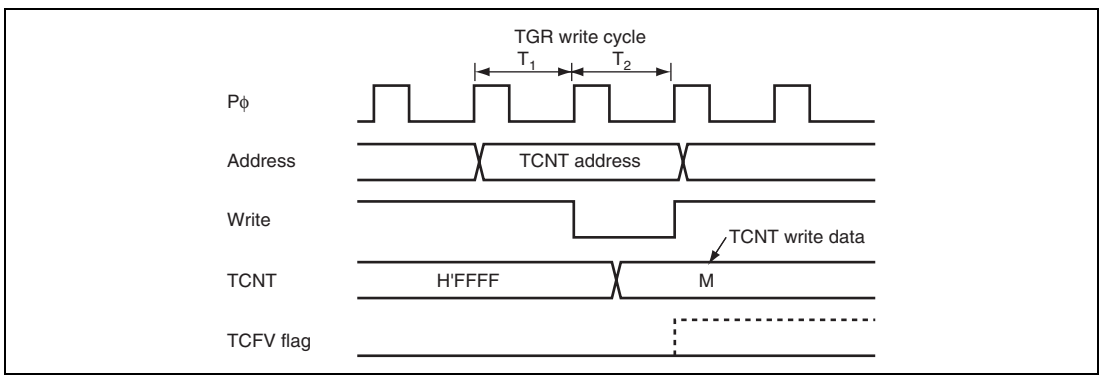


**Figure 9.54 Conflict between TCNT Write and Overflow**

## 13.3 Register Descriptions

The SCI has the following registers. Some bits in the serial mode register (SMR), serial status register (SSR), and serial control register (SCR) have different functions in different modes—normal serial communication interface mode and smart card interface mode; therefore, the bits are described separately for each mode in the corresponding register sections.

**Channel 0:**

- Receive shift register_0 (RSR_0)
- Transmit shift register_0 (TSR_0)
- Receive data register_0 (RDR_0)
- Transmit data register_0 (TDR_0)
- Serial mode register_0 (SMR_0)
- Serial control register_0 (SCR_0)
- Serial status register_0 (SSR_0)
- Smart card mode register_0 (SCMR_0)
- Bit rate register_0 (BRR_0)

**Channel 1:**

- Receive shift register_1 (RSR_1)
- Transmit shift register_1 (TSR_1)
- Receive data register_1 (RDR_1)
- Transmit data register_1 (TDR_1)
- Serial mode register_1 (SMR_1)
- Serial control register_1 (SCR_1)
- Serial status register_1 (SSR_1)
- Smart card mode register_1 (SCMR_1)
- Bit rate register_1 (BRR_1)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | MPIE | 0 | R/W | Multiprocessor Interrupt Enable (valid only when the MP bit in SMR is 1 in asynchronous mode)<br><br>When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is disabled. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, see section 13.5, Multiprocessor Communication Function.<br><br>When receive data including MPB = 0 in SSR is being received, transfer of the received data from RSR to RDR, detection of reception errors, and the settings of RDRF, FER, and ORER flags in SSR are not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is automatically cleared to 0, and RXI and ERI interrupt requests (in the case where the TIE and RIE bits in SCR are set to 1) and setting of the FER and ORER flags are enabled. |
| 2 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br><br>When this bit is set to 1, a TEI interrupt request is enabled. A TEI interrupt request can be cancelled by reading 1 from the TDRE flag and then clearing the flag to 0 in order to clear the TEND flag to 0, or by clearing the TEIE bit to 0. |

**Table 13.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)**

| Bit Rate (bit/s) | Operating Frequency Pϕ (MHz) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 10 | | 16 | | 20 | |
| | n | N | n | N | n | N | n | N |
| 110 | | | | | | | | |
| 250 | 3 | 124 | — | — | 3 | 249 | | |
| 500 | 2 | 249 | — | — | 3 | 124 | — | — |
| 1k | 2 | 124 | — | — | 2 | 249 | — | — |
| 2.5k | 1 | 199 | 1 | 249 | 2 | 99 | 2 | 124 |
| 5k | 1 | 99 | 1 | 124 | 1 | 199 | 1 | 249 |
| 10k | 0 | 199 | 0 | 249 | 1 | 99 | 1 | 124 |
| 25k | 0 | 79 | 0 | 99 | 0 | 159 | 0 | 199 |
| 50k | 0 | 39 | 0 | 49 | 0 | 79 | 0 | 99 |
| 100k | 0 | 19 | 0 | 24 | 0 | 39 | 0 | 49 |
| 250k | 0 | 7 | 0 | 9 | 0 | 15 | 0 | 19 |
| 500k | 0 | 3 | 0 | 4 | 0 | 7 | 0 | 9 |
| 1M | 0 | 1 | | | 0 | 3 | 0 | 4 |
| 2.5M | | | 0 | 0* | | | 0 | 1 |
| 5M | | | | | | | 0 | 0* |

### 13.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 13.5. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the RDRF, PER, FER, and ORER flags, or RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.

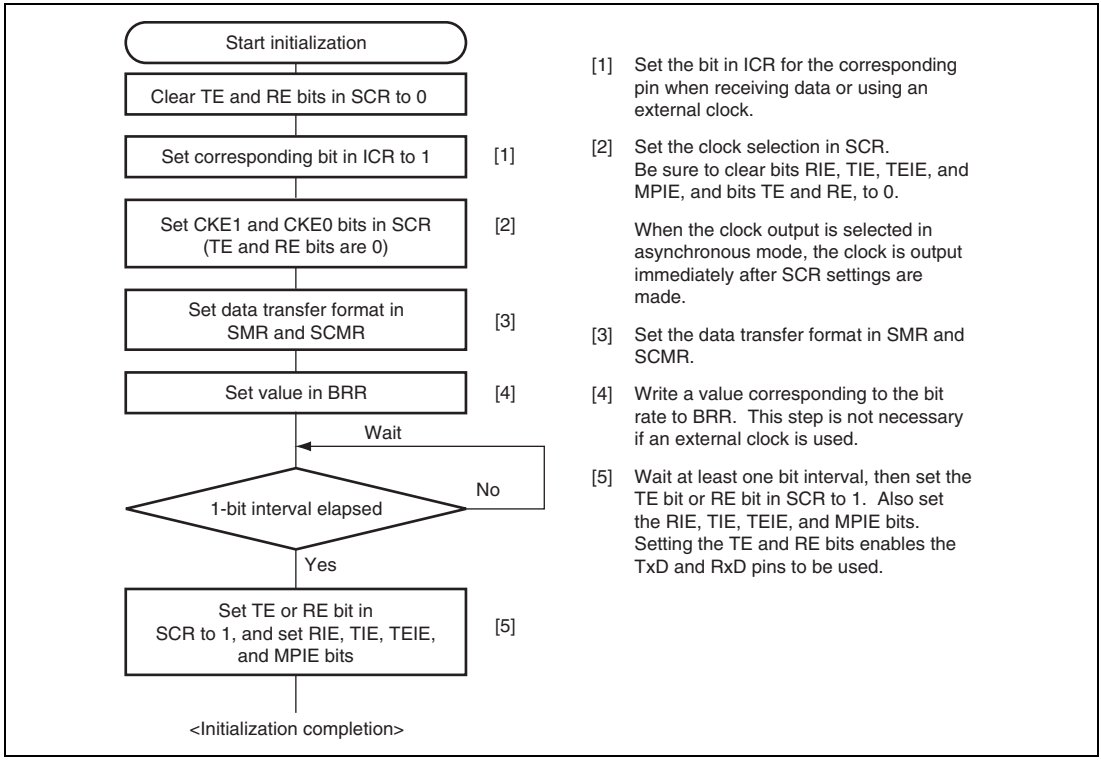


**Figure 13.5 Sample SCI Initialization Flowchart**

**Table 17.2 Crystal Resonator Characteristics**

| Frequency (MHz) | 8 | 12 | 18 |
|---|---|---|---|
| $R_S$ Max. (Ω) | 80 | 60 | 40 |
| $C_0$ Max. (pF) | | 7 | |

### 17.2.2 External Clock Input

An external clock signal can be input as shown in the examples in figure 17.4. If the XTAL pin is left open, make sure that parasitic capacitance is no more than 10 pF. When the counter clock is input to the XTAL pin, make sure that the external clock is held high in standby mode.

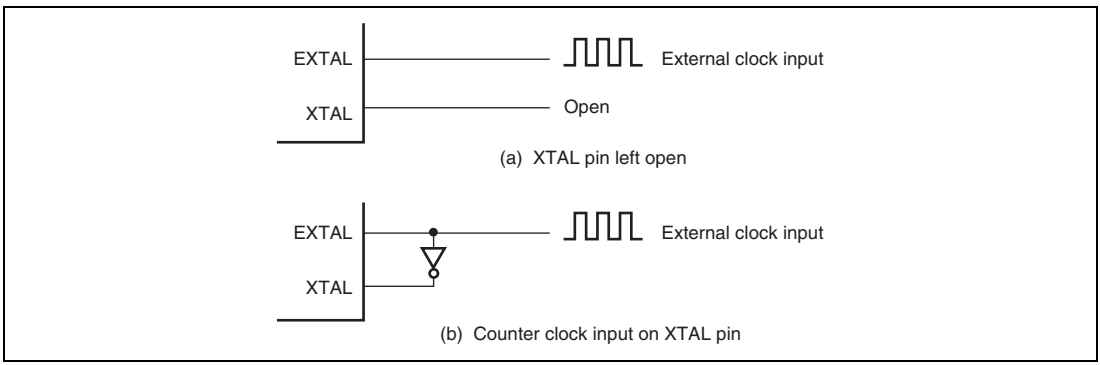


**Figure 17.4 External Clock Input (Examples)**

For the input conditions of the external clock, refer to table 20.4, Clock Timing, in section 20.3.1, Clock Timing. The input external clock should be from 8 to 18 MHz.

## 17.3 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscillator by a factor of 4. The frequency multiplication factor is fixed. The phase difference is controlled so that the timing of the rising edge of the internal clock is the same as that of the EXTAL pin signal.

## 17.4 Frequency Divider

The frequency divider divides the PLL clock to generate a 1/2, 1/4, or 1/8 clock. After bits ICK2 to ICK0, PCK 2 to PCK0, and BCK2 to BCK0 are modified, this LSI operates at the modified frequency.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 13 | — | 0 | R/W | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 12 | STS4 | 0 | R/W | Standby Timer Select 4 to 0<br>These bits select the time the MCU waits for the clock to settle when software standby mode is cleared by an external interrupt. With a crystal resonator, refer to table 18.2 and make a selection according to the operating frequency so that the standby time is at least equal to the oscillation settling time. With an external clock, a PLL circuit settling time is necessary. Refer to table 18.2 to set the standby time.<br>While oscillation is being settled, the timer is counted on the Pϕ clock frequency. Careful consideration is required in multi-clock mode.<br>00000: Reserved<br>00001: Reserved<br>00010: Reserved<br>00011: Reserved<br>00100: Reserved<br>00101: Standby time = 64 states<br>00110: Standby time = 512 states<br>00111: Standby time = 1024 states<br>01000: Standby time = 2048 states<br>01001: Standby time = 4096 states<br>01010: Standby time = 16384 states<br>01011: Standby time = 32768 states<br>01100: Standby time = 65536 states<br>01101: Standby time = 131072 states<br>01110: Standby time = 262144 states<br>01111: Standby time = 524288 states<br>1XXXX: Reserved |
| 11 | STS3 | 1 | R/W | |
| 10 | STS2 | 1 | R/W | |
| 9 | STS1 | 1 | R/W | |
| 8 | STS0 | 1 | R/W | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| PDPCR | PD7PCR | PD6PCR | PD5PCR | PD4PCR | PD3PCR | PD2PCR | PD1PCR | PD0PCR | I/O port |
| PEPCR | PE7PCR | PE6PCR | PE6PCR | PE4PCR | PE3PCR | PE2PCR | PE1PCR | PE0PCR | |
| PFPCR | PF7PCR | PF6PCR | PF5PCR | PF4PCR | PF3PCR | PF2PCR | PF1PCR | PF0PCR | |
| PHPCR | PH7PCR | PH6PCR | PH5PCR | PH4PCR | PH3PCR | PH2PCR | PH1PCR | PH0PCR | |
| PIPCR | PI7PCR | PI6PCR | PI5PCR | PI4PCR | PI3PCR | PI2PCR | PI1PCR | PI0PCR | |
| P2ODR | P27ODR | P26ODR | P25ODR | P24ODR | P23ODR | P22ODR | P21ODR | P20ODR | |
| PFODR | PF7ODR | PF6ODR | PF5ODR | PF4ODR | PF3ODR | PF2ODR | PF1ODR | PF0ODR | |
| PFCR0 | CS7E | CS6E | CS5E | CS4E | CS3E | CS2E | CS1E | CS0E | |
| PFCR1 | CS7SA | CS7SB | CS6SA | CS6SB | CS5SA | CS5SB | CS4SA | CS4SB | |
| PFCR2 | — | CS2S | BSS | BSE | — | RDWRE | ASOE | — | |
| PFCR4 | A23E | A22E | A21E | — | — | — | — | — | |
| PFCR6 | — | LHWROE | — | — | TCLKS | — | — | — | |
| PFCR9 | TPUMS5 | TPUMS4 | TPUMS3A | TPUMS3B | TPUMS2 | TPUMS1 | TPUMS0A | TPUMS0B | |
| PFCRB | — | — | — | — | ITS11 | ITS10 | ITS9 | ITS8 | |
| PFCRC | ITS7 | ITS6 | ITS5 | ITS4 | ITS3 | ITS2 | ITS1 | ITS0 | |
| SSIER | — | — | — | — | SSI11 | SSI10 | SSI9 | SSI8 | INTC |
| | SSI7 | SSI6 | SSI5 | SSI4 | SSI3 | SSI2 | SSI1 | SSI0 | |
| IPRA | — | IPRA14 | IPRA13 | IPRA12 | — | IPRA10 | IPRA9 | IPRA8 | |
| | — | IPRA6 | IPRA5 | IPRA4 | — | IPRA2 | IPRA1 | IPRA0 | |
| IPRB | — | IPRB14 | IPRB13 | IPRB12 | — | IPRB10 | IPRB9 | IPRB8 | |
| | — | IPRB6 | IPRB5 | IPRB4 | — | IPRB2 | IPRB1 | IPRB0 | |
| IPRC | — | IPRC14 | IPRC13 | IPRC12 | — | IPRC10 | IPRC9 | IPRC8 | |
| | — | IPRC6 | IPRC5 | IPRC4 | — | IPRC2 | IPRC1 | IPRC0 | |
| IPRE | — | — | — | — | — | IPRE10 | IPRE9 | IPRE8 | |
| | — | — | — | — | — | — | — | — | |
| IPRF | — | — | — | — | — | IPRF10 | IPRF9 | IPRF8 | |
| | — | IPRF6 | IPRF5 | IPRF4 | — | IPRF2 | IPRF1 | IPRF0 | |
| IPRG | — | IPRG14 | IPRG13 | IPRG12 | — | IPRG10 | IPRG9 | IPRG8 | |
| | — | IPRG6 | IPRG5 | IPRG4 | — | IPRG2 | IPRG1 | IPRG0 | |
| IPRH | — | IPRH14 | IPRH13 | IPRH12 | — | IPRH10 | IPRH9 | IPRH8 | |
| | — | IPRH6 | IPRH5 | IPRH4 | — | IPRH2 | IPRH1 | IPRH0 | |
| IPRK | — | — | — | — | — | — | — | — | |
| | — | IPRK6 | IPRK5 | IPRK4 | — | IPRK2 | IPRK1 | IPRK0 | |
| IPRL | — | IPRL14 | IPRL13 | IPRL12 | — | — | — | — | |
| | — | IPRL6 | IPRL5 | IPRL4 | — | — | — | — | |

| Register Abbreviation | Reset | Module Stop State | Sleep Mode | All-Module-Clock-Stop Mode | Software Standby Mode | Hardware Standby Mode | Module |
|---|---|---|---|---|---|---|---|
| TCNT_0 | Initialized | — | — | — | — | Initialized | TMR_0 |
| TCNT_1 | Initialized | — | — | — | — | Initialized | TMR_1 |
| TCCR_0 | Initialized | — | — | — | — | Initialized | TMR_0 |
| TCCR_1 | Initialized | — | — | — | — | Initialized | TMR_1 |
| TSTR | Initialized | — | — | — | — | Initialized | TPU |
| TSYR | Initialized | — | — | — | — | Initialized | |
| TCR_0 | Initialized | — | — | — | — | Initialized | TPU_0 |
| TMDR_0 | Initialized | — | — | — | — | Initialized | |
| TIORH_0 | Initialized | — | — | — | — | Initialized | |
| TIORL_0 | Initialized | — | — | — | — | Initialized | |
| TIER_0 | Initialized | — | — | — | — | Initialized | |
| TSR_0 | Initialized | — | — | — | — | Initialized | |
| TCNT_0 | Initialized | — | — | — | — | Initialized | |
| TGRA_0 | Initialized | — | — | — | — | Initialized | |
| TGRB_0 | Initialized | — | — | — | — | Initialized | |
| TGRC_0 | Initialized | — | — | — | — | Initialized | |
| TGRD_0 | Initialized | — | — | — | — | Initialized | |
| TCR_1 | Initialized | — | — | — | — | Initialized | TPU_1 |
| TMDR_1 | Initialized | — | — | — | — | Initialized | |
| TIOR_1 | Initialized | — | — | — | — | Initialized | |
| TIER_1 | Initialized | — | — | — | — | Initialized | |
| TSR_1 | Initialized | — | — | — | — | Initialized | |
| TCNT_1 | Initialized | — | — | — | — | Initialized | |
| TGRA_1 | Initialized | — | — | — | — | Initialized | |
| TGRB_1 | Initialized | — | — | — | — | Initialized | |
| TCR_2 | Initialized | — | — | — | — | Initialized | TPU_2 |
| TMDR_2 | Initialized | — | — | — | — | Initialized | |
| TIOR_2 | Initialized | — | — | — | — | Initialized | |
| TIER_2 | Initialized | — | — | — | — | Initialized | |