**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 | Not For New Designs |
| Core Processor | H8/300H |
| Core Size | 16-Bit |
| Speed | 20MHz |
| Connectivity | SCI |
| Peripherals | PWM, WDT |
| Number of I/O | 30 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 5.5V |
| Data Converters | A/D 4x10b |
| Oscillator Type | Internal |
| Operating Temperature | -20°C ~ 75°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP |
| Supplier Device Package | 64-LFQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/df36014fpv |

**Appendix**

## 2.1 Address Space and Memory Map

The address space of this LSI is 64 kbytes, which includes the program area and the data area. Figures 2.1 show the memory maps.

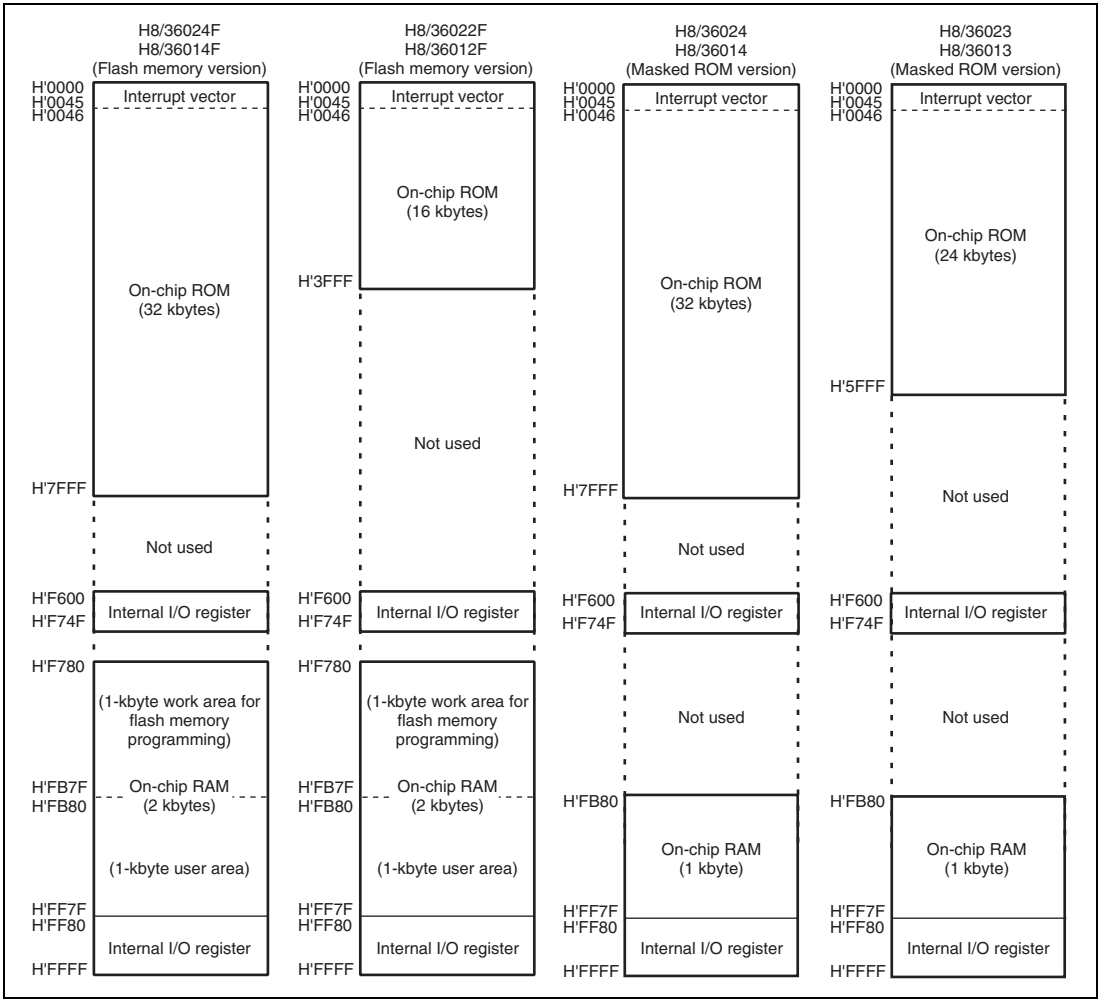



**Figure 2.1 Memory Map (1)**

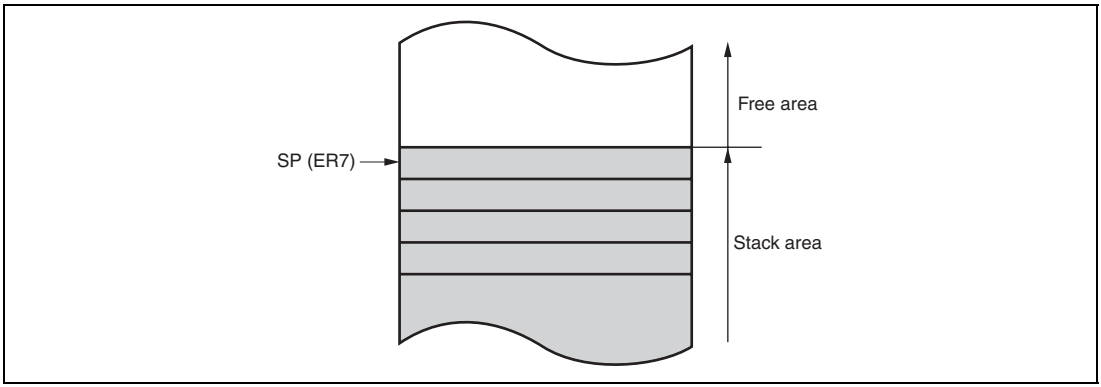



**Figure 2.4 Relationship between Stack Pointer and Stack Area**

### 2.2.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched, the least significant PC bit is regarded as 0). The PC is initialized when the start address is loaded by the vector address generated during reset exception-handling sequence.

### 2.2.3 Condition-Code Register (CCR)

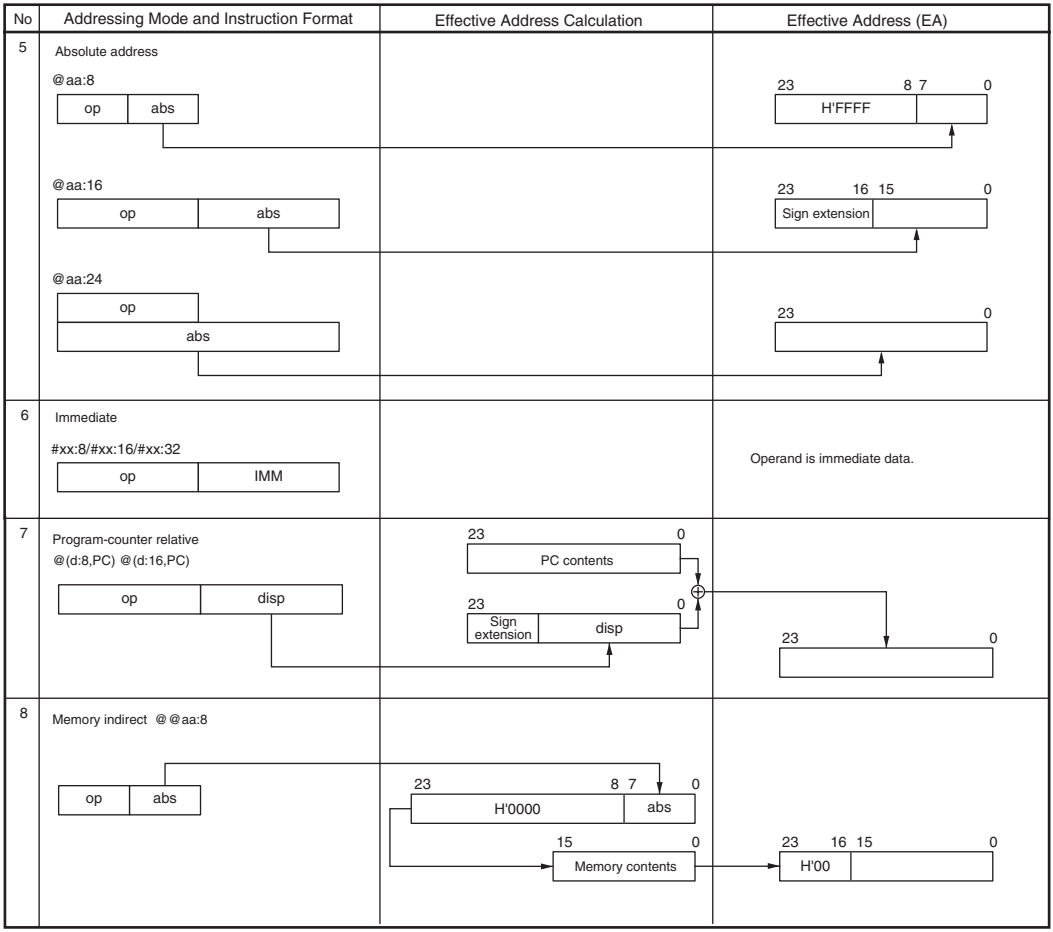
This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags. The I bit is initialized to 1 by reset exception-handling sequence, but other bits are not initialized.

Some instructions leave flag bits unchanged. Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

For the action of each instruction on the flag bits, see Appendix A.1, Instruction List.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | I | 1 | R/W | Interrupt Mask Bit<br>Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence. |
| 6 | UI | Undefined | R/W | User Bit<br>Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. |
| 5 | H | Undefined | R/W | Half-Carry Flag<br>When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise. |
| 4 | U | Undefined | R/W | User Bit<br>Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. |
| 3 | N | Undefined | R/W | Negative Flag<br>Stores the value of the most significant bit of data as a sign bit. |
| 2 | Z | Undefined | R/W | Zero Flag<br>Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data. |
| 1 | V | Undefined | R/W | Overflow Flag<br>Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times. |
| 0 | C | Undefined | R/W | Carry Flag<br>Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:<br>• Add instructions, to indicate a carry<br>• Subtract instructions, to indicate a borrow<br>• Shift and rotate instructions, to indicate a carry<br>The carry flag is also used as a bit accumulator by bit manipulation instructions. |

## Table 2.12 Effective Address Calculation (2)




**Legend**

r, rm,rn : Register field
op : Operation field
disp : Displacement
IMM : Immediate data
abs : Absolute address

### 5.3.2 Notes on Board Design

When using a crystal resonator (ceramic resonator), place the resonator and its load capacitors as close as possible to the $OSC_1$ and $OSC_2$ pins. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation (see figure 5.7).

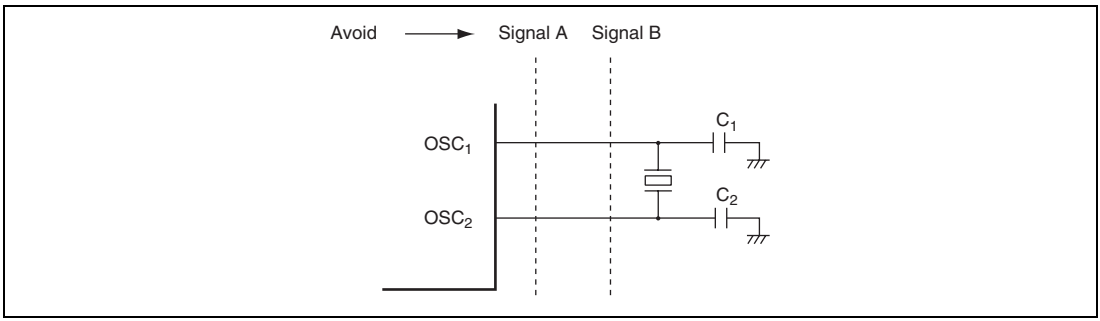



**Figure 5.7 Example of Incorrect Board Design**

## 7.3 On-Board Programming Modes

There is a mode for programming/erasing of the flash memory; boot mode, which enables on-board programming/erasing. On-board programming/erasing can also be performed in user program mode. At reset-start in reset mode, this LSI changes to a mode depending on the TEST pin settings, $\overline{\text{NMI}}$ pin settings, and input level of each port, as shown in table 7.1. The input level of each pin must be defined four states before the reset ends.

When changing to boot mode, the boot program built into this LSI is initiated. The boot program transfers the programming control program from the externally-connected host to on-chip RAM via SCI3. After erasing the entire flash memory, the programming control program is executed. This can be used for programming initial values in the on-board state or for a forcible return when programming/erasing can no longer be done in user program mode. In user program mode, individual blocks can be erased and programmed by branching to the user program/erase control program prepared by the user.

**Table 7.1 Setting Programming Modes**

| TEST | $\overline{\text{NMI}}$ | E10T_0 | PB0 | PB1 | PB2 | LSI State after Reset End |
|---|---|---|---|---|---|---|
| 0 | 1 | X | X | X | X | User Mode |
| 0 | 0 | 1 | X | X | X | Boot Mode |

**Legend**: X: Don't care.

### 7.3.1 Boot Mode

Table 7.2 shows the boot mode operations between reset end and branching to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 7.4, Flash Memory Programming/Erasing.
2. SCI3 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity.
3. When the boot program is initiated, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI3 bit rate to match that of the host. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary. After the reset is complete, it takes approximately 100 states before the chip is ready to measure the low-level period.

## 9.5 Port 8

Port 8 is a general I/O port also functioning as a timer W I/O pin. Each pin of the port 8 is shown in figure 9.5. The register setting of the timer W has priority for functions of the pins P84/FTIOD, P83/FTIOC, P82/FTIOB, and P81/FTIOA. The P80/FTCI pin also functions as a timer W input port that is connected to the timer W regardless of the register setting of port 8.

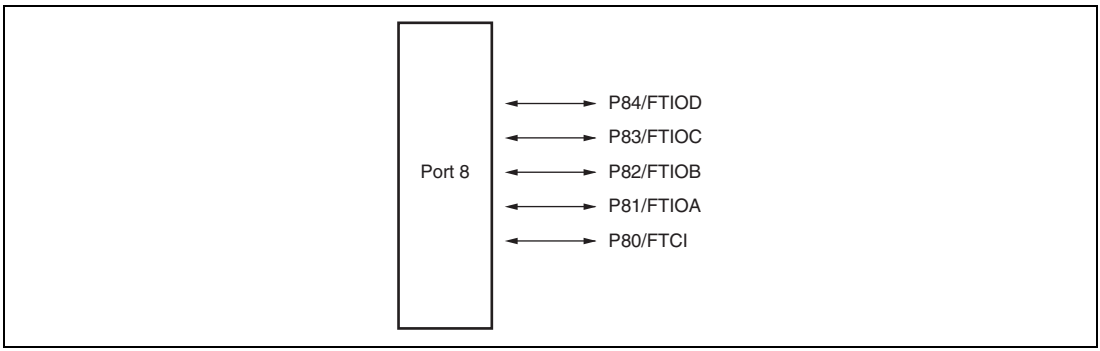



**Figure 9.5 Port 8 Pin Configuration**

Port 8 has the following registers.

- Port control register 8 (PCR8)
- Port data register 8 (PDR8)

### 9.5.1 Port Control Register 8 (PCR8)

PCR8 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 8.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 5 | — | — | — | Reserved |
| 4 | PCR84 | 0 | W | When each of the port 8 pins P84 to P80 functions as an general I/O port, setting a PCR8 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 3 | PCR83 | 0 | W | |
| 2 | PCR82 | 0 | W | |
| 1 | PCR81 | 0 | W | |
| 0 | PCR80 | 0 | W | |

### 10.5.2 Pulse Output with Arbitrary Pulse Width and Delay from TRGV Input

The trigger function can be used to output a pulse with an arbitrary pulse width at an arbitrary delay from the TRGV input, as shown in figure 10.10. To set up this output:

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORB.
2. Set bits OS3 to OS0 in TCSRV so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
3. Set bits TVEG1 and TVEG0 in TCRV1 and set TRGE to select the falling edge of the TRGV input.
4. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
5. After these settings, a pulse waveform will be output without further software intervention, with a delay determined by TCORA from the TRGV input, and a pulse width determined by (TCORB – TCORA).

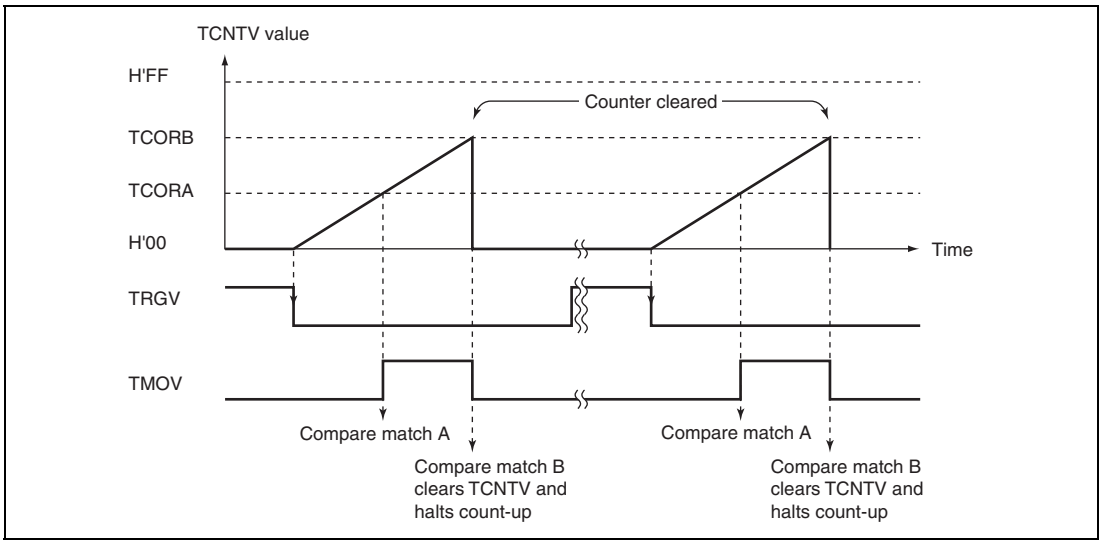



**Figure 10.10 Example of Pulse Output Synchronized to TRGV Input**

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | IMFB | 0 | R/W | Input Capture/Compare Match Flag B<br>[Setting conditions]<br>• TCNT = GRB when GRB functions as an output compare register<br>• The TCNT value is transferred to GRB by an input capture signal when GRB functions as an input capture register<br>[Clearing condition]<br>Read IMFB when IMFB = 1, then write 0 in IMFB |
| 0 | IMFA | 0 | R/W | Input Capture/Compare Match Flag A<br>[Setting conditions]<br>• TCNT = GRA when GRA functions as an output compare register<br>• The TCNT value is transferred to GRA by an input capture signal when GRA functions as an input capture register<br>[Clearing condition]<br>Read IMFA when IMFA = 1, then write 0 in IMFA |

### 11.3.5 Timer I/O Control Register 0 (TIOR0)

TIOR0 selects the functions of GRA and GRB, and specifies the functions of the FTIOA and FTIOB pins.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 6 | IOB2 | 0 | R/W | I/O Control B2<br>Selects the GRB function.<br>0: GRB functions as an output compare register<br>1: GRB functions as an input capture register |
| 5 | IOB1 | 0 | R/W | I/O Control B1 and B0 |
| 4 | IOB0 | 0 | R/W | When IOB2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOB pin at GRB compare match<br>10: 1 output to the FTIOB pin at GRB compare match<br>11: Output toggles to the FTIOB pin at GRB compare match<br>When IOB2 = 1,<br>00: Input capture at rising edge at the FTIOB pin<br>01: Input capture at falling edge at the FTIOB pin<br>1X: Input capture at rising and falling edges of the FTIOB pin |
| 3 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 2 | IOA2 | 0 | R/W | I/O Control A2<br>Selects the GRA function.<br>0: GRA functions as an output compare register<br>1: GRA functions as an input capture register |

Periodic counting operation can be performed when GRA is set as an output compare register and bit CCLR in TCRW is set to 1. When the count matches GRA, TCNT is cleared to H'0000, the IMFA flag in TSRW is set to 1. If the corresponding IMIEA bit in TIERW is set to 1, an interrupt request is generated. TCNT continues counting from H'0000. Figure 11.3 shows periodic counting.

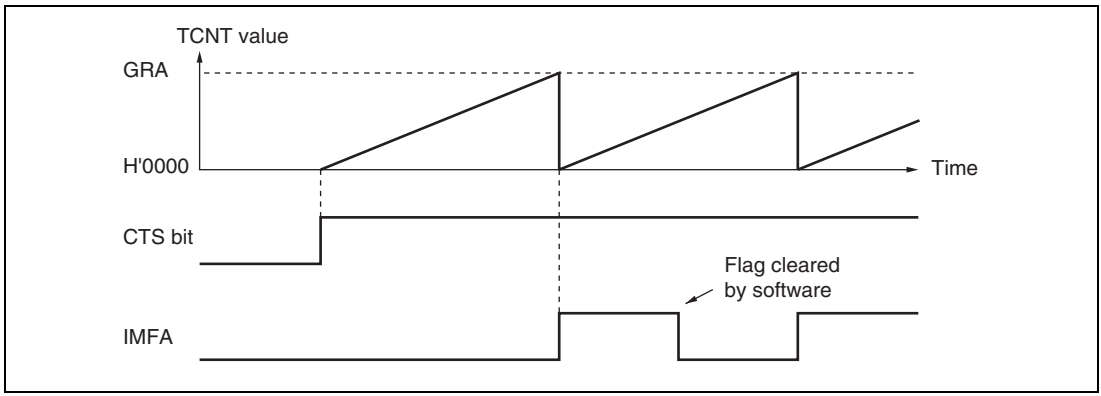



**Figure 11.3 Periodic Counter Operation**

By setting a general register as an output compare register, compare match A, B, C, or D can cause the output at the FTIOA, FTIOB, FTIOC, or FTIOD pin to output 0, output 1, or toggle. Figure 11.4 shows an example of 0 and 1 output when TCNT operates as a free-running counter, 1 output is selected for compare match A, and 0 output is selected for compare match B. When signal is already at the selected output level, the signal level does not change at compare match.

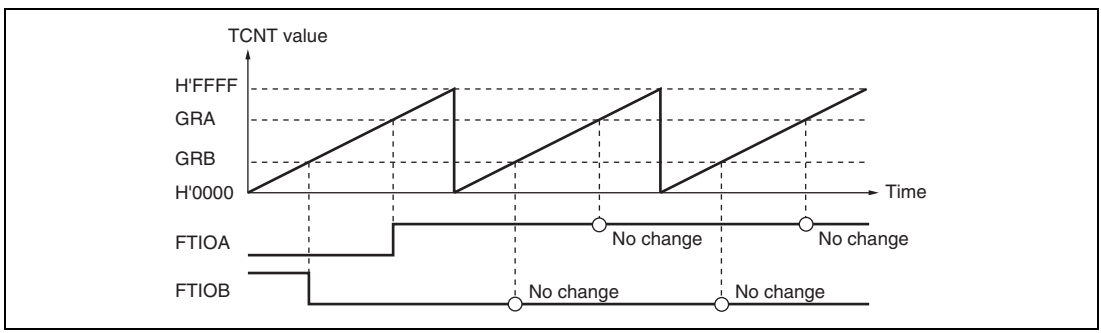



**Figure 11.4 0 and 1 Output Example (TOA = 0, TOB = 1)**

The TCNT value can be captured into a general register (GRA, GRB, GRC, or GRD) when a signal level changes at an input-capture pin (FTIOA, FTIOB, FTIOC, or FTIOD). Capture can take place on the rising edge, falling edge, or both edges. By using the input-capture function, the pulse width and periods can be measured. Figure 11.7 shows an example of input capture when both edges of FTIOA and the falling edge of FTIOB are selected as capture edges. TCNT operates as a free-running counter.

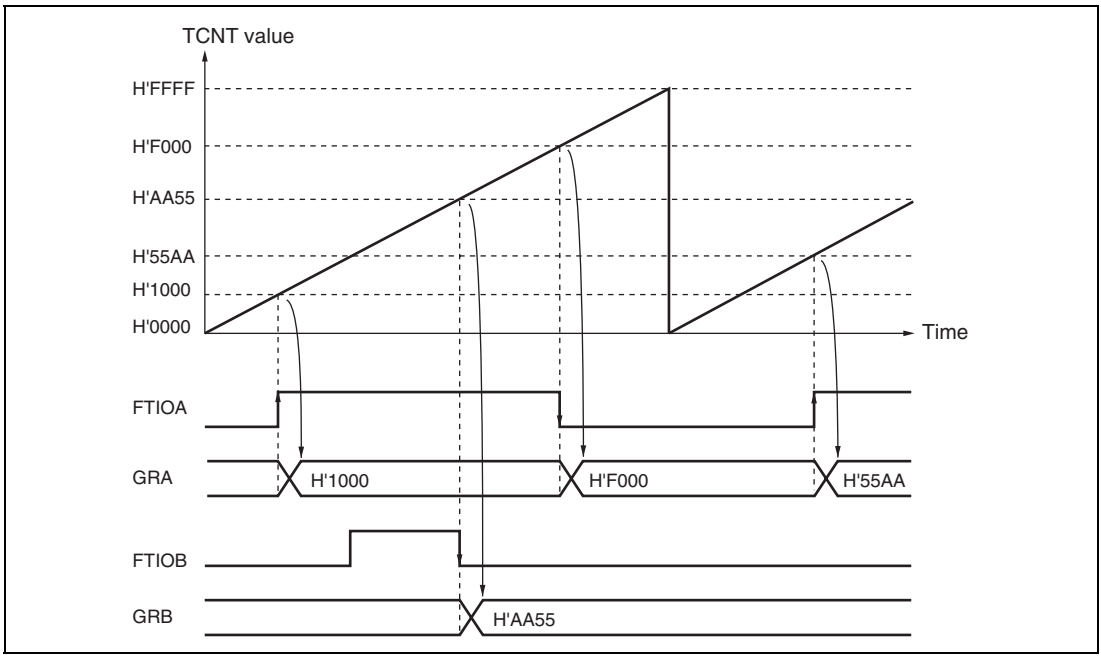



**Figure 11.7 Input Capture Operating Example**

Figure 11.11 shows an example of buffer operation when the FTIOB pin is set to PWM mode and GRD is set as the buffer register for GRB. TCNT is cleared by compare match A, and FTIOB outputs 1 at compare match B and 0 at compare match A.

Due to the buffer operation, the FTIOB output level changes and the value of buffer register GRD is transferred to GRB whenever compare match B occurs. This procedure is repeated every time compare match B occurs.

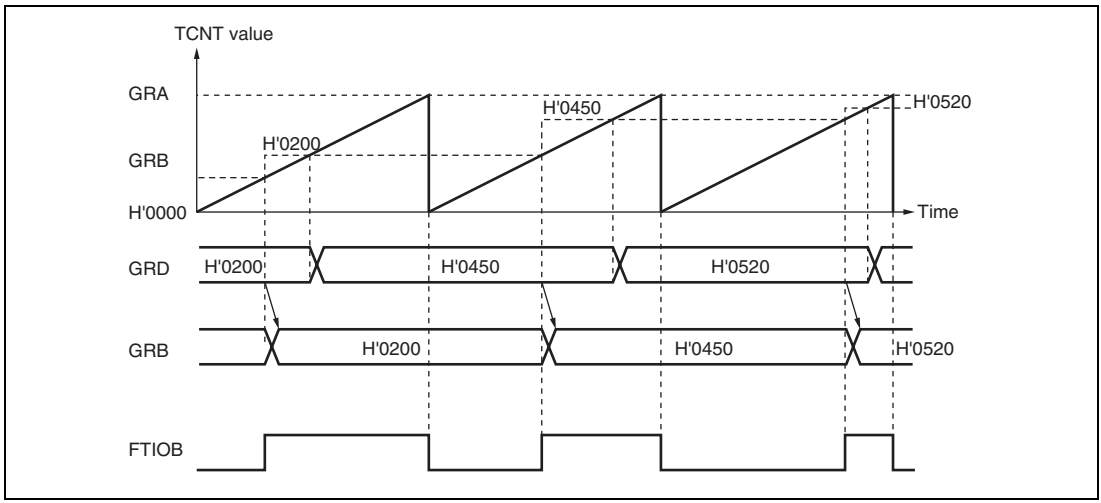



**Figure 11.11 Buffer Operation Example (Output Compare)**

## 12.3 Operation

The watchdog timer is provided with an 8-bit counter. If 1 is written to WDON while writing 0 to B2WI when the TCSRWE bit in TCSRWD is set to 1, TCWD begins counting up. (To operate the watchdog timer, two write accesses to TCSRWD are required.) When a clock pulse is input after the TCWD count value has reached H'FF, the watchdog timer overflows and an internal reset signal is generated. The internal reset signal is output for a period of 256 $\phi_{osc}$ clock cycles. TCWD is a writable counter, and when a value is set in TCWD, the count-up starts from that value. An overflow period in the range of 1 to 256 input clock cycles can therefore be set, according to the TCWD set value.

Figure 12.2 shows an example of watchdog timer operation.

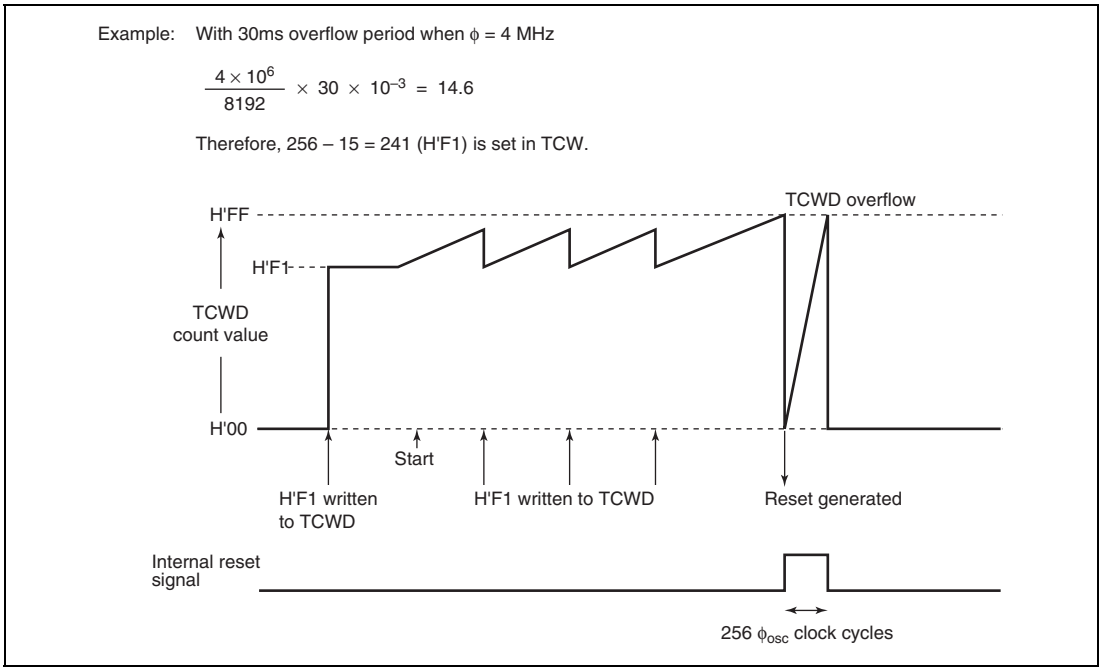



**Figure 12.2 Watchdog Timer Operation Example**

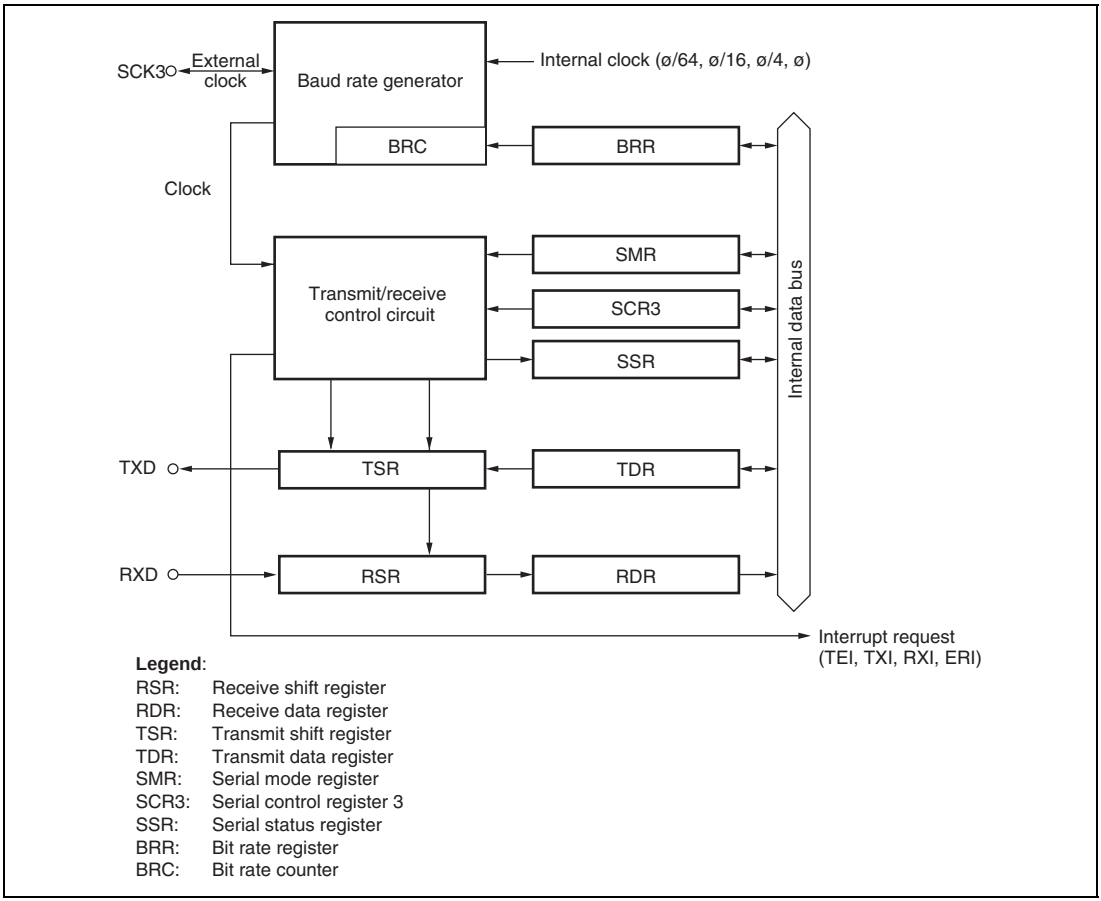
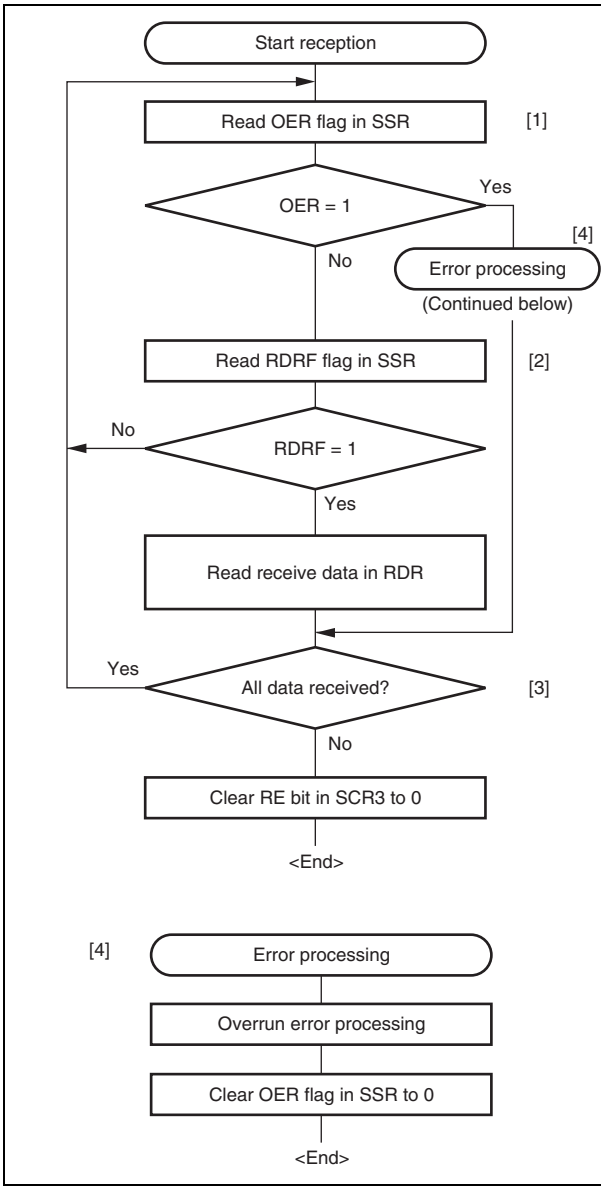



**Figure 13.1 Block Diagram of SCI3**




[1] Read the OER flag in SSR to determine if there is an error. If an overrun error has occurred, execute overrun error processing.

[2] Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[3] To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag and reading RDR should be finished. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[4] If an overrun error occurs, read the OER flag in SSR, and after performing the appropriate error processing, clear the OER flag to 0. Reception cannot be resumed if the OER flag is set to 1.

**Figure 13.13 Sample Serial Reception Flowchart (Clocked Synchronous Mode)**

## 13.7 Interrupts

SCI3 creates the following six interrupt requests: transmission end, transmit data empty, receive data full, and receive errors (overrun error, framing error, and parity error). Table 13.7 shows the interrupt sources.

**Table 13.7 SCI3 Interrupt Requests**

| Interrupt Requests | Abbreviation | Interrupt Sources |
|---|---|---|
| Receive Data Full | RXI | Setting RDRF in SSR |
| Transmit Data Empty | TXI | Setting TDRE in SSR |
| Transmission End | TEI | Setting TEND in SSR |
| Receive Error | ERI | Setting OER, FER, and PER in SSR |

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TXI interrupt request is generated even if the transmit data is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TEI interrupt request is generated even if the transmit data has not been sent. It is possible to make use of the most of these interrupt requests efficiently by transferring the transmit data to TDR in the interrupt routine. To prevent the generation of these interrupt requests (TXI and TEI), set the enable bits (TIE and TEIE) that correspond to these interrupt requests to 1, after transferring the transmit data to TDR.

### 18.2.6 Flash Memory Characteristics

**Table 18.7 Flash Memory Characteristics**

$V_{CC}$ = 3.0 V to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20°C to +75°C, unless otherwise specified.

| Item | | Symbol | Test Condition | Values Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Programming time (per 128 bytes)*1*2*4 | | $t_P$ | | — | 7 | 200 | ms |
| Erase time (per block) *1*3*6 | | $t_E$ | | — | 100 | 1200 | ms |
| Reprogramming count | | $N_{WEC}$ | | 1000 | 10000 | — | Times |
| Programming | Wait time after SWE bit setting*1 | x | | 1 | — | — | µs |
| | Wait time after PSU bit setting*1 | y | | 50 | — | — | µs |
| | Wait time after P bit setting *1*4 | z1 | 1 ≤ n ≤ 6 | 28 | 30 | 32 | µs |
| | | z2 | 7 ≤ n ≤ 1000 | 198 | 200 | 202 | µs |
| | | z3 | Additional-programming | 8 | 10 | 12 | µs |
| | Wait time after P bit clear*1 | α | | 5 | — | — | µs |
| | Wait time after PSU bit clear*1 | β | | 5 | — | — | µs |
| | Wait time after PV bit setting*1 | γ | | 4 | — | — | µs |
| | Wait time after dummy write*1 | ε | | 2 | — | — | µs |
| | Wait time after PV bit clear*1 | η | | 2 | — | — | µs |
| | Wait time after SWE bit clear*1 | θ | | 100 | — | — | µs |
| | Maximum programming count*1*4*5 | N | | — | — | 1000 | Times |

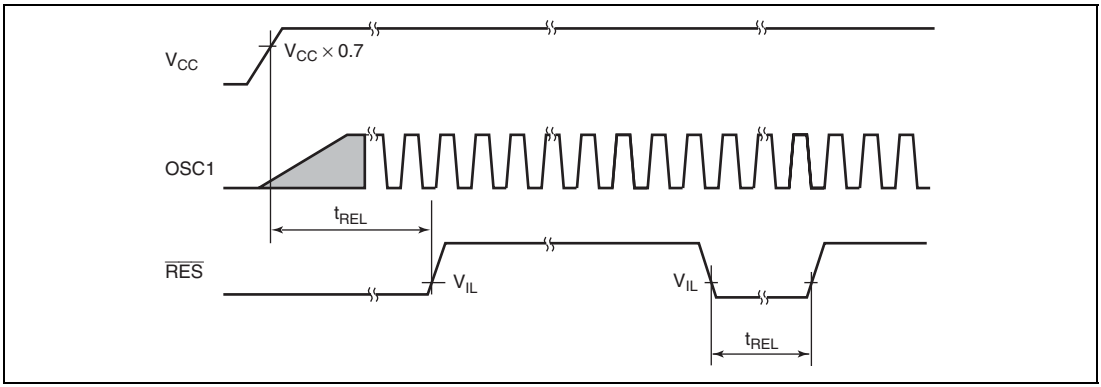



**Figure 18.2 $\overline{RES}$ Low Width Timing**

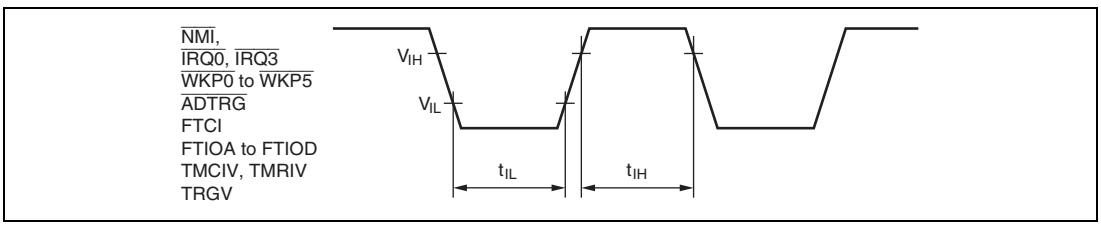



**Figure 18.3 Input Timing**

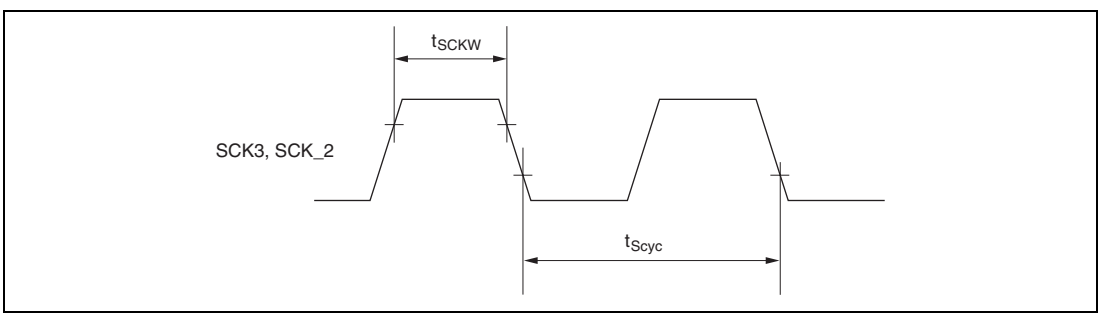



**Figure 18.4 SCK3 Input Clock Timing**