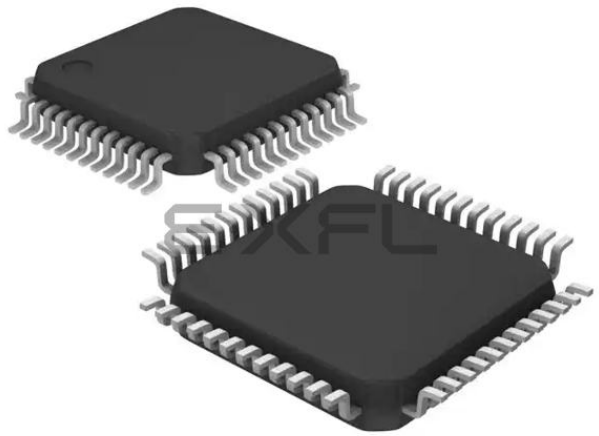


**Welcome to E-XFL.COM**

## 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 | H8/300H |
| Core Size | 16-Bit |
| Speed | 20MHz |
| Connectivity | SCI |
| Peripherals | LVD, POR, 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 | 48-LQFP |
| Supplier Device Package | 48-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/df36014gfxv |

## 2.2 Register Configuration

The H8/300H CPU has the internal registers shown in figure 2.2. There are two types of registers; general registers and control registers. The control registers are a 24-bit program counter (PC), and an 8-bit condition code register (CCR).

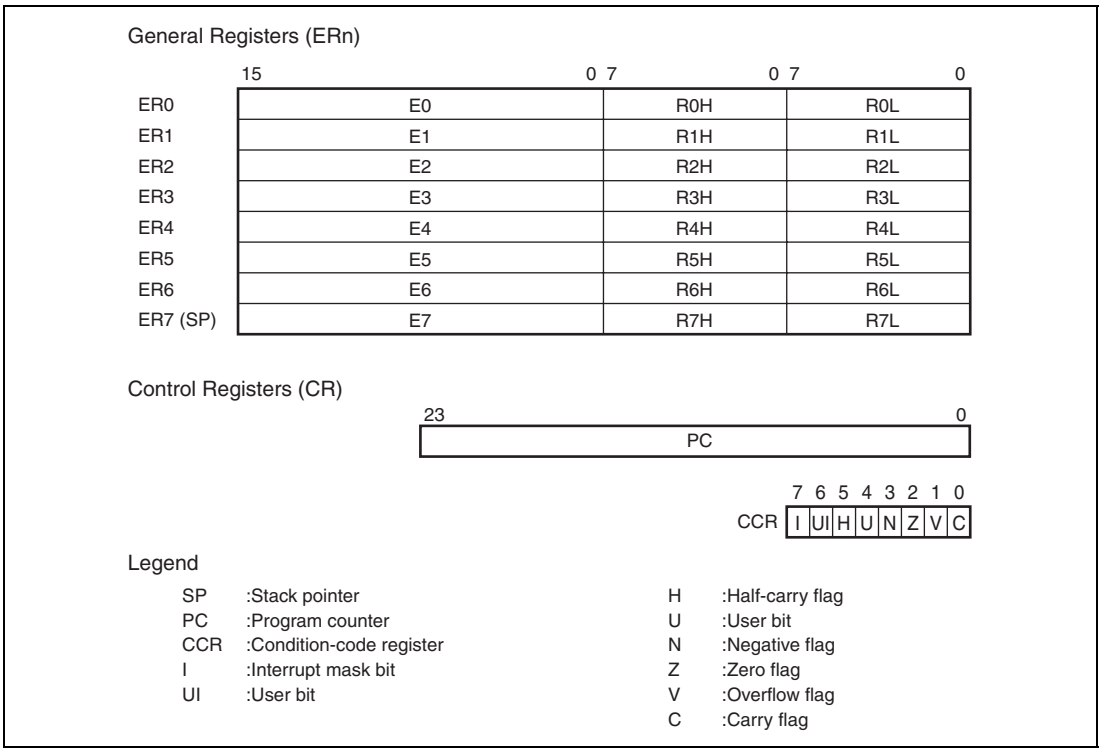



**Figure 2.2 CPU Registers**

## 2.3 Data Formats

The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

### 2.3.1 General Register Data Formats

Figure 2.5 shows the data formats in general registers.

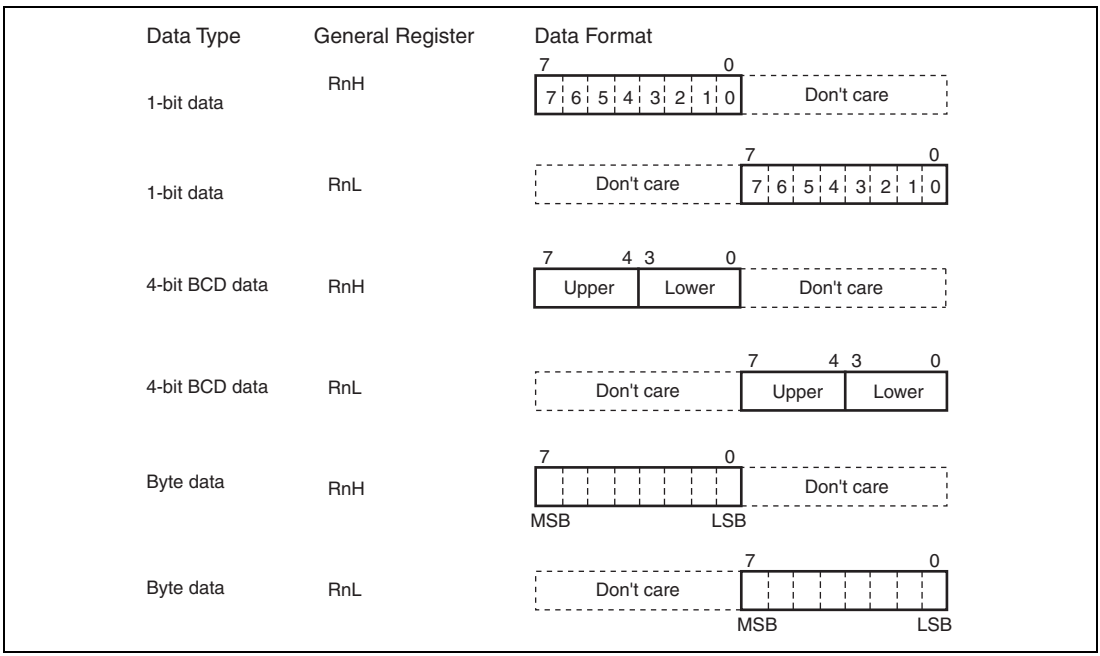



**Figure 2.5 General Register Data Formats (1)**

### 6.1.2 System Control Register 2 (SYSCR2)

SYSCR2 controls the power-down modes, as well as SYSCR1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | SMSEL | 0 | R/W | Sleep Mode Selection<br>This bit selects the mode to transit after the execution of a SLEEP instruction, as well as bit SSBY of SYSCR1.<br>For details, see table 6.2. |
| 6 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 5 | DTON | 0 | R/W | Direct Transfer on Flag<br>This bit selects the mode to transit after the execution of a SLEEP instruction, as well as bit SSBY of SYSCR1.<br>For details, see table 6.2. |
| 4 | MA2 | 0 | R/W | Active Mode Clock Select 2 to 0 |
| 3 | MA1 | 0 | R/W | These bits select the operating clock frequency in active and sleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed. |
| 2 | MA0 | 0 | R/W | 0XX: $\phi_{osc}$<br>100: $\phi_{osc}$/8<br>101: $\phi_{osc}$/16<br>110: $\phi_{osc}$/32<br>111: $\phi_{osc}$/64 |
| 1, 0 | — | All 0 | — | Reserved<br>These bits are always read as 0. |

Legend: X : Don't care.

### 10.3.3 Timer Control Register V0 (TCRV0)

TCRV0 selects the input clock signals of TCNTV, specifies the clearing conditions of TCNTV, and controls each interrupt request.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CMIEB | 0 | R/W | Compare Match Interrupt Enable B<br>When this bit is set to 1, interrupt request from the CMFB bit in TCSRV is enabled. |
| 6 | CMIEA | 0 | R/W | Compare Match Interrupt Enable A<br>When this bit is set to 1, interrupt request from the CMFA bit in TCSRV is enabled. |
| 5 | OVIE | 0 | R/W | Timer Overflow Interrupt Enable<br>When this bit is set to 1, interrupt request from the OVF bit in TCSRV is enabled. |
| 4 | CCLR1 | 0 | R/W | Counter Clear 1 and 0 |
| 3 | CCLR0 | 0 | R/W | These bits specify the clearing conditions of TCNTV.<br>00: Clearing is disabled<br>01: Cleared by compare match A<br>10: Cleared by compare match B<br>11: Cleared on the rising edge of the TMRIV pin. The operation of TCNTV after clearing depends on TRGE in TCRV1. |
| 2 | CKS2 | 0 | R/W | Clock Select 2 to 0<br>These bits select clock signals to input to TCNTV and the counting condition in combination with ICKS0 in TCRV1.<br>Refer to table 10.2. |
| 1 | CKS1 | 0 | R/W | |
| 0 | CKS0 | 0 | R/W | |

## 10.5 Timer V Application Examples

### 10.5.1 Pulse Output with Arbitrary Duty Cycle

Figure 10.9 shows an example of output of pulses with an arbitrary duty cycle.

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORA.
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 CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
4. With these settings, a waveform is output without further software intervention, with a period determined by TCORA and a pulse width determined by TCORB.

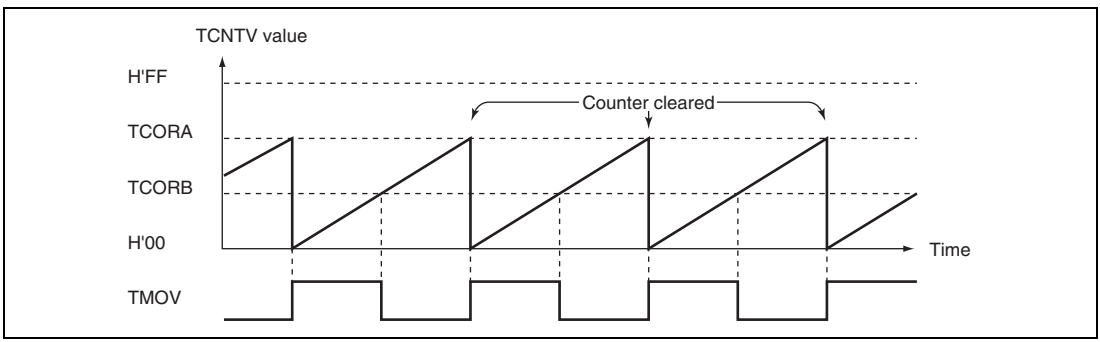



**Figure 10.9 Pulse Output Example**

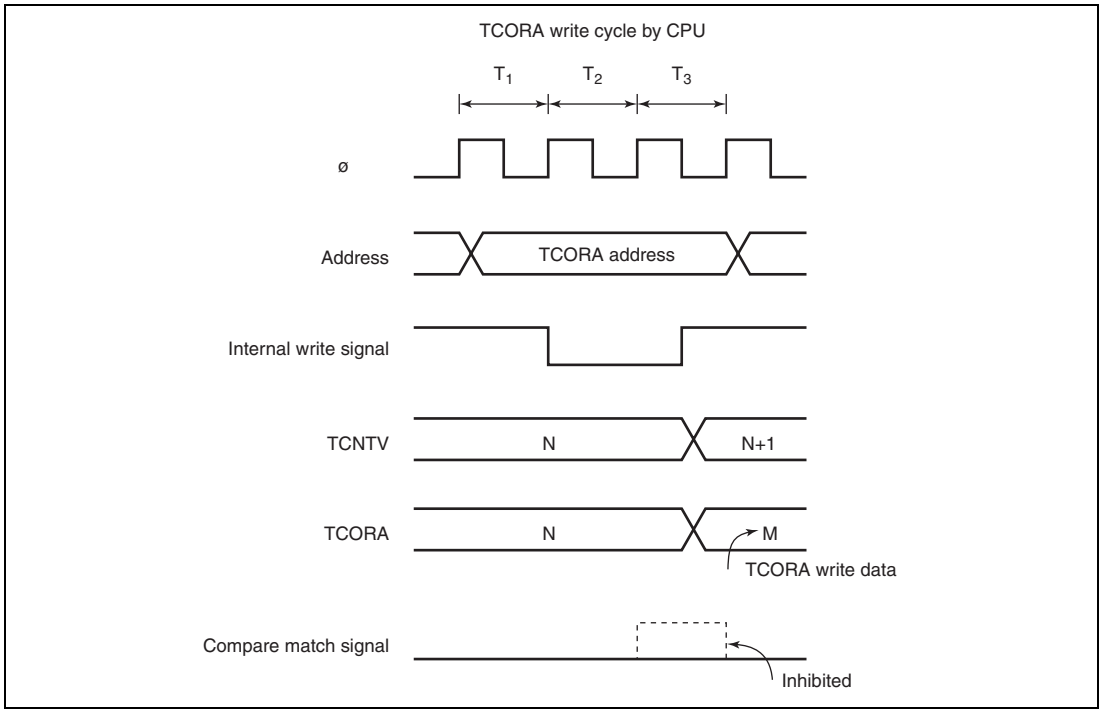



**Figure 10.12 Contention between TCORA Write and Compare Match**

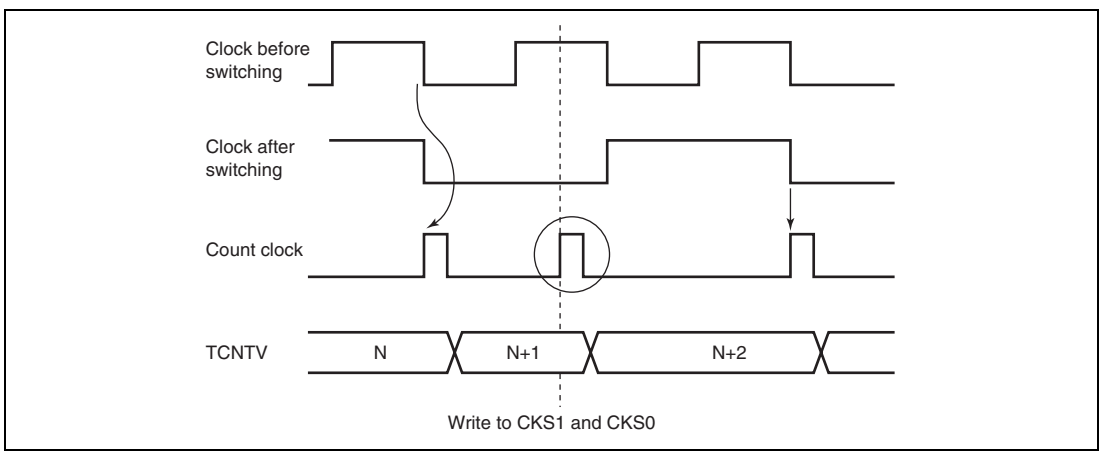



**Figure 10.13 Internal Clock Switching and TCNTV Operation**

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.

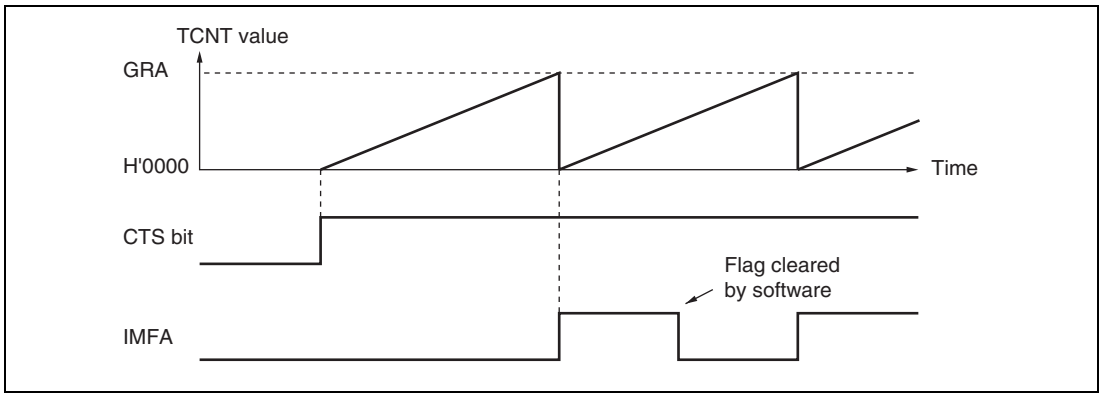



**Figure 11.3 Periodic Counter Operation**

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.

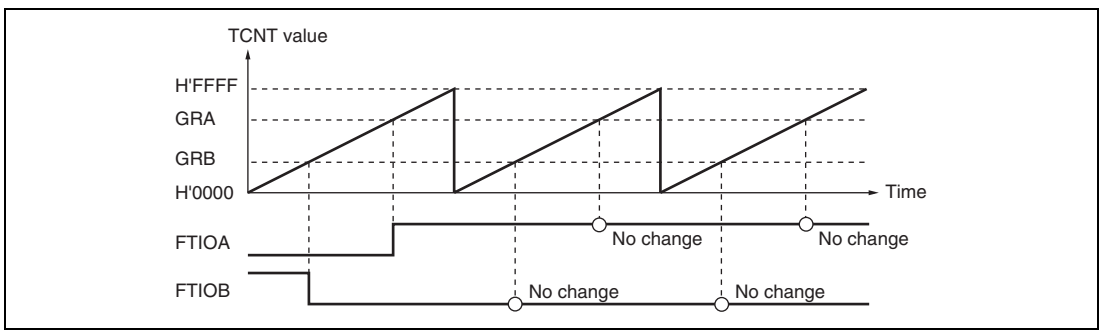



**Figure 11.4 0 and 1 Output Example (TOA = 0, TOB = 1)**

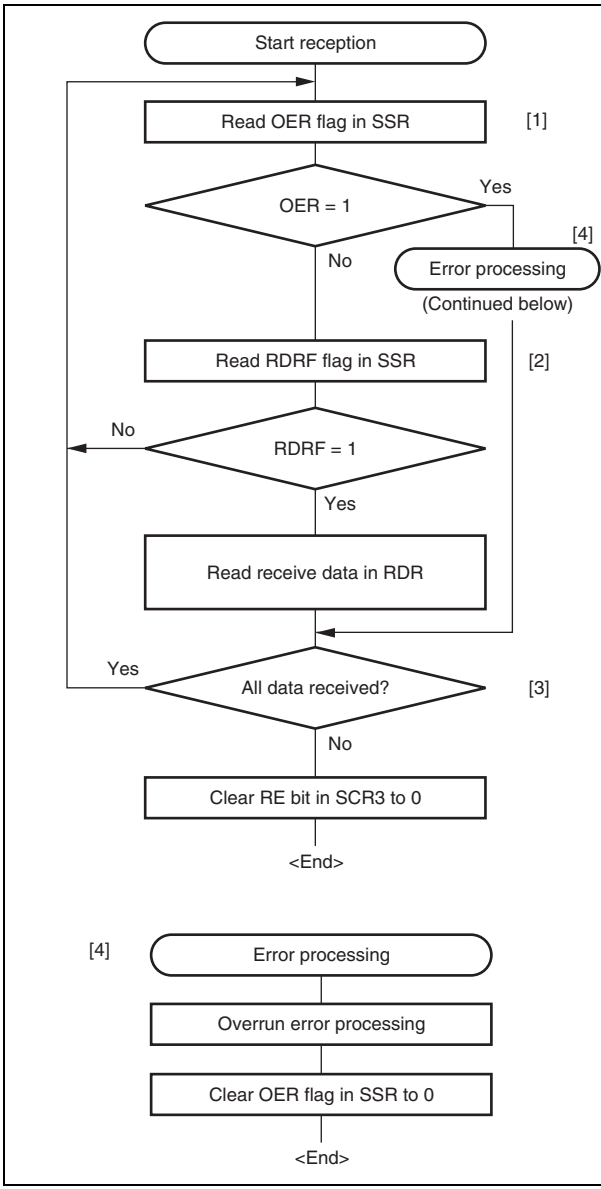
**Table 13.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)**

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | | | 2.097152 | | | 2.4576 | | | 3 | | |
| Bit Rate (bits/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 1 | 141 | 0.03 | 1 | 148 | –0.04 | 1 | 174 | –0.26 | 1 | 212 | 0.03 |
| 150 | 1 | 103 | 0.16 | 1 | 108 | 0.21 | 1 | 127 | 0.00 | 1 | 155 | 0.16 |
| 300 | 0 | 207 | 0.16 | 0 | 217 | 0.21 | 0 | 255 | 0.00 | 1 | 77 | 0.16 |
| 600 | 0 | 103 | 0.16 | 0 | 108 | 0.21 | 0 | 127 | 0.00 | 0 | 155 | 0.16 |
| 1200 | 0 | 51 | 0.16 | 0 | 54 | –0.70 | 0 | 63 | 0.00 | 0 | 77 | 0.16 |
| 2400 | 0 | 25 | 0.16 | 0 | 26 | 1.14 | 0 | 31 | 0.00 | 0 | 38 | 0.16 |
| 4800 | 0 | 12 | 0.16 | 0 | 13 | –2.48 | 0 | 15 | 0.00 | 0 | 19 | –2.34 |
| 9600 | 0 | 6 | –6.99 | 0 | 6 | –2.48 | 0 | 7 | 0.00 | 0 | 9 | –2.34 |
| 19200 | 0 | 2 | 8.51 | 0 | 2 | 13.78 | 0 | 3 | 0.00 | 0 | 4 | –2.34 |
| 31250 | 0 | 1 | 0.00 | 0 | 1 | 4.86 | 0 | 1 | 22.88 | 0 | 2 | 0.00 |
| 38400 | 0 | 1 | –18.62 | 0 | 1 | –14.67 | 0 | 1 | 0.00 | — | — | — |

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 3.6864 | | | 4 | | | 4.9152 | | | 5 | | |
| Bit Rate (bits/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 64 | 0.70 | 2 | 70 | 0.03 | 2 | 86 | 0.31 | 2 | 88 | –0.25 |
| 150 | 1 | 191 | 0.00 | 1 | 207 | 0.16 | 1 | 255 | 0.00 | 2 | 64 | 0.16 |
| 300 | 1 | 95 | 0.00 | 1 | 103 | 0.16 | 1 | 127 | 0.00 | 1 | 129 | 0.16 |
| 600 | 0 | 191 | 0.00 | 0 | 207 | 0.16 | 0 | 255 | 0.00 | 1 | 64 | 0.16 |
| 1200 | 0 | 95 | 0.00 | 0 | 103 | 0.16 | 0 | 127 | 0.00 | 0 | 129 | 0.16 |
| 2400 | 0 | 47 | 0.00 | 0 | 51 | 0.16 | 0 | 63 | 0.00 | 0 | 64 | 0.16 |
| 4800 | 0 | 23 | 0.00 | 0 | 25 | 0.16 | 0 | 31 | 0.00 | 0 | 32 | –1.36 |
| 9600 | 0 | 11 | 0.00 | 0 | 12 | 0.16 | 0 | 15 | 0.00 | 0 | 15 | 1.73 |
| 19200 | 0 | 5 | 0.00 | 0 | 6 | –6.99 | 0 | 7 | 0.00 | 0 | 7 | 1.73 |
| 31250 | — | — | — | 0 | 3 | 0.00 | 0 | 4 | –1.70 | 0 | 4 | 0.00 |
| 38400 | 0 | 2 | 0.00 | 0 | 2 | 8.51 | 0 | 3 | 0.00 | 0 | 3 | 1.73 |

Legend

—: A setting is available but error occurs




[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)**

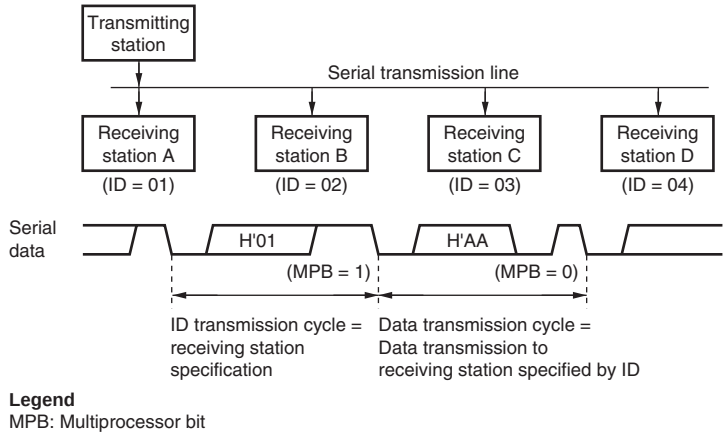



**Figure 13.15 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)**

### 13.6.1 Multiprocessor Serial Data Transmission

Figure 13.16 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI3 operations are the same as those in asynchronous mode.

## 14.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes; single mode and scan mode. When changing the operating mode or analog input channel, in order to prevent incorrect operation, first clear the bit ADST to 0 in ADCSR. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

### 14.4.1 Single Mode

In single mode, A/D conversion is performed once for the analog input on the specified single channel as follows:

1. A/D conversion is started from the first channel when the ADST bit in ADCSR is set to 1, according to software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the wait state.

### 14.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially for the analog input on the specified channels (four channels maximum) as follows:

1. When the ADST bit is set to 1 by software, or external trigger input, A/D conversion starts on the first channel in the group.
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. Conversion of the first channel in the group starts again.
4. The ADST bit is not automatically cleared to 0. Steps [2] to [3] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops.

### 15.3.2 Low-Voltage Detection Circuit

#### (1) LVDR (Reset by Low Voltage Detect) Circuit

Figure 15.3 shows the timing of the LVDR function. The LVDR enters the module-standby state after a power-on reset is canceled. To operate the LVDR, set the LVDE bit in LVDCR to 1, wait for 50 µs ($t_{LVDON}$) until the reference voltage and the low-voltage-detection power supply have stabilized by a software timer, etc., then set the LVDRE bit in LVDCR to 1. After that, the output settings of ports must be made. To cancel the low-voltage detection circuit, first the LVDRE bit should be cleared to 0 and then the LVDE bit should be cleared to 0. The LVDE and LVDRE bits must not be cleared to 0 simultaneously because incorrect operation may occur.

When the power-supply voltage falls below the Vreset voltage (typ. = 2.3 V or 3.6 V), the LVDR clears the $\overline{\text{LVDRES}}$ signal to 0, and resets the prescaler S. The low-voltage detection reset state remains in place until a power-on reset is generated. When the power-supply voltage rises above the Vreset voltage again, the prescaler S starts counting. It counts 131,072 clock (ϕ) cycles, and then releases the internal reset signal. In this case, the LVDE, LVDSEL, and LVDRE bits in LVDCR are not initialized.

Note that if the power supply voltage (Vcc) falls below $V_{LVDRmin}$ = 1.0 V and then rises from that point, the low-voltage detection reset may not occur.

If the power supply voltage (Vcc) falls below Vpor = 100 mV, a power-on reset occurs.

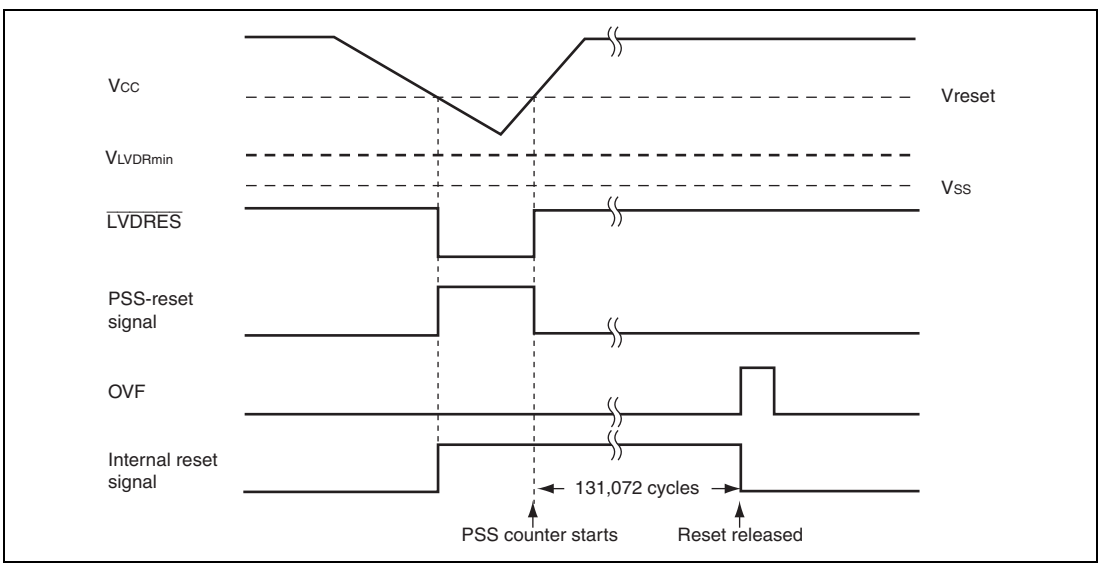



**Figure 15.3 Operational Timing of LVDR Circuit**

| Item | Symbol | Applicable Pins | Test Condition | Values Min | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Output high voltage | $V_{OH}$ | P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P70, P84 to P80 | $V_{CC}$ = 4.0 V to 5.5 V<br>$-I_{OH}$ = 1.5 mA | $V_{CC}$ – 1.0 | — | — | V | |
| | | | $-I_{OH}$ = 0.1 mA | $V_{CC}$ – 0.5 | — | — | | |
| Output low voltage | $V_{OL}$ | P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P70 | $V_{CC}$ = 4.0 V to 5.5 V<br>$I_{OL}$ = 1.6 mA | — | — | 0.6 | V | |
| | | | $I_{OL}$ = 0.4 mA | — | — | 0.4 | | |
| | | P84 to P80 | $V_{CC}$ = 4.0 V to 5.5 V<br>$I_{OL}$ = 20.0 mA | — | — | 1.5 | V | |
| | | | $V_{CC}$ = 4.0 V to 5.5 V<br>$I_{OL}$ = 10.0 mA | — | — | 1.0 | | |
| | | | $V_{CC}$ = 4.0 V to 5.5 V<br>$I_{OL}$ = 1.6 mA | — | — | 0.4 | | |
| | | | $I_{OL}$ = 0.4 mA | — | — | 0.4 | | |
| Input/ output leakage current | $\lvert I_{IL} \rvert$ | OSC1, $\overline{\text{RES}}$, $\overline{\text{NMI}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, $\overline{\text{IRQ0}}$, $\overline{\text{IRQ3}}$, $\overline{\text{ADTRG}}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, RXD_2, RXD_3*[1], SCK3, SCK3_2, SCK3_3*[1] | $V_{IN}$ = 0.5 V to ($V_{CC}$ – 0.5 V) | — | — | 1.0 | μA | |
| | | P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P70, P84 to P80 | $V_{IN}$ = 0.5 V to ($V_{CC}$ – 0.5 V) | — | — | 1.0 | μA | |
| | | PB3 to PB0 | $V_{IN}$ = 0.5 V to ($AV_{CC}$ – 0.5 V) | — | — | 1.0 | μA | |

## 3. Logic instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | — | | I | H | N | Z | V | C | Normal | Advanced |
| AND | AND.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8∧#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.B Rs, Rd | B | | 2 | | | | | | | | Rd8∧Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16∧#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | AND.W Rs, Rd | W | | 2 | | | | | | | | Rd16∧Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32∧#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | AND.L ERs, ERd | L | | 4 | | | | | | | | ERd32∧ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| OR | OR.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8∨#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.B Rs, Rd | B | | 2 | | | | | | | | Rd8∨Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16∨#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | OR.W Rs, Rd | W | | 2 | | | | | | | | Rd16∨Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32∨#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | OR.L ERs, ERd | L | | 4 | | | | | | | | ERd32∨ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| XOR | XOR.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8⊕#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.B Rs, Rd | B | | 2 | | | | | | | | Rd8⊕Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16⊕#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | XOR.W Rs, Rd | W | | 2 | | | | | | | | Rd16⊕Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32⊕#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | XOR.L ERs, ERd | L | | 4 | | | | | | | | ERd32⊕ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| NOT | NOT.B Rd | B | | 2 | | | | | | | | ¬ Rd8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | NOT.W Rd | W | | 2 | | | | | | | | ¬ Rd16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | NOT.L ERd | L | | 2 | | | | | | | | ¬ Rd32 → Rd32 | — | — | ↕ | ↕ | 0 | — | 2 | |

## 5. Bit manipulation instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | — | | I | H | N | Z | V | C | Normal | Advanced |
| BSET | BSET #xx:3, Rd | B | | 2 | | | | | | | | (#xx:3 of Rd8) ← 1 | — | — | — | — | — | — | 2 | |
| | BSET #xx:3, @ERd | B | | | 4 | | | | | | | (#xx:3 of @ERd) ← 1 | — | — | — | — | — | — | 8 | |
| | BSET #xx:3, @aa:8 | B | | | | | | 4 | | | | (#xx:3 of @aa:8) ← 1 | — | — | — | — | — | — | 8 | |
| | BSET Rn, Rd | B | | 2 | | | | | | | | (Rn8 of Rd8) ← 1 | — | — | — | — | — | — | 2 | |
| | BSET Rn, @ERd | B | | | 4 | | | | | | | (Rn8 of @ERd) ← 1 | — | — | — | — | — | — | 8 | |
| | BSET Rn, @aa:8 | B | | | | | | 4 | | | | (Rn8 of @aa:8) ← 1 | — | — | — | — | — | — | 8 | |
| BCLR | BCLR #xx:3, Rd | B | | 2 | | | | | | | | (#xx:3 of Rd8) ← 0 | — | — | — | — | — | — | 2 | |
| | BCLR #xx:3, @ERd | B | | | 4 | | | | | | | (#xx:3 of @ERd) ← 0 | — | — | — | — | — | — | 8 | |
| | BCLR #xx:3, @aa:8 | B | | | | | | 4 | | | | (#xx:3 of @aa:8) ← 0 | — | — | — | — | — | — | 8 | |
| | BCLR Rn, Rd | B | | 2 | | | | | | | | (Rn8 of Rd8) ← 0 | — | — | — | — | — | — | 2 | |
| | BCLR Rn, @ERd | B | | | 4 | | | | | | | (Rn8 of @ERd) ← 0 | — | — | — | — | — | — | 8 | |
| | BCLR Rn, @aa:8 | B | | | | | | 4 | | | | (Rn8 of @aa:8) ← 0 | — | — | — | — | — | — | 8 | |
| BNOT | BNOT #xx:3, Rd | B | | 2 | | | | | | | | (#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8) | — | — | — | — | — | — | 2 | |
| | BNOT #xx:3, @ERd | B | | | 4 | | | | | | | (#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd) | — | — | — | — | — | — | 8 | |
| | BNOT #xx:3, @aa:8 | B | | | | | | 4 | | | | (#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8) | — | — | — | — | — | — | 8 | |
| | BNOT Rn, Rd | B | | 2 | | | | | | | | (Rn8 of Rd8) ← ¬ (Rn8 of Rd8) | — | — | — | — | — | — | 2 | |
| | BNOT Rn, @ERd | B | | | 4 | | | | | | | (Rn8 of @ERd) ← ¬ (Rn8 of @ERd) | — | — | — | — | — | — | 8 | |
| | BNOT Rn, @aa:8 | B | | | | | | 4 | | | | (Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8) | — | — | — | — | — | — | 8 | |
| BTST | BTST #xx:3, Rd | B | | 2 | | | | | | | | ¬ (#xx:3 of Rd8) → Z | — | — | — | ↕ | — | — | 2 | |
| | BTST #xx:3, @ERd | B | | | 4 | | | | | | | ¬ (#xx:3 of @ERd) → Z | — | — | — | ↕ | — | — | 6 | |
| | BTST #xx:3, @aa:8 | B | | | | | | 4 | | | | ¬ (#xx:3 of @aa:8) → Z | — | — | — | ↕ | — | — | 6 | |
| | BTST Rn, Rd | B | | 2 | | | | | | | | ¬ (Rn8 of @Rd8) → Z | — | — | — | ↕ | — | — | 2 | |
| | BTST Rn, @ERd | B | | | 4 | | | | | | | ¬ (Rn8 of @ERd) → Z | — | — | — | ↕ | — | — | 6 | |
| | BTST Rn, @aa:8 | B | | | | | | 4 | | | | ¬ (Rn8 of @aa:8) → Z | — | — | — | ↕ | — | — | 6 | |
| BLD | BLD #xx:3, Rd | B | | 2 | | | | | | | | (#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |

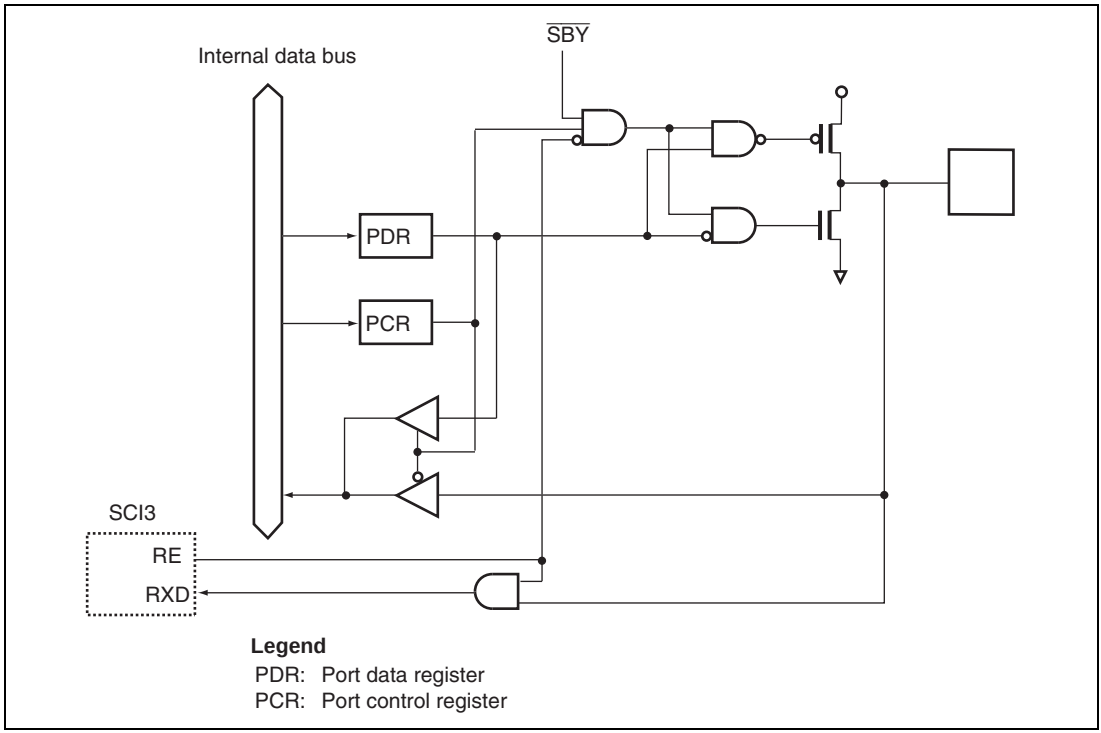
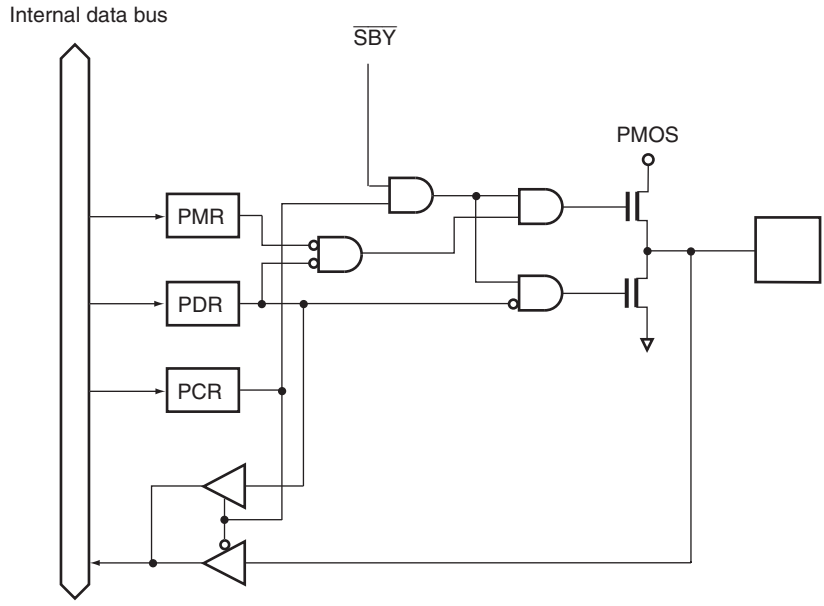



**Figure B.7 Port 2 Block Diagram (P21)**




**Legend**
PMR: Port mode register
PDR: Port data register
PCR: Port control register

**Figure B.9 Port 5 Block Diagram (P57, P56) (H8/36014)**

| Product Type | | | Product Code | Model Marking | Package Code |
|---|---|---|---|---|---|
| H8/36010 | Masked ROM version | Standard product | HD64336010FP | HD64336010(***)FP | LQFP-64 (FP-64E) |
| | | | HD64336010FX | HD64336010(***)FX | LQFP-48 (FP-48F) |
| | | | HD64336010FY | HD64336010(***)FY | LQFP-48 (FP-48B) |
| | | | HD64336010FT | HD64336010(***)FT | QFN-48(TNP-48) |
| | | Product with POR & LVDC | HD64336010GFP | HD64336010G(***)FP | LQFP-64 (FP-64E) |
| | | | HD64336010GFX | HD64336010G(***)FX | LQFP-48 (FP-48F) |
| | | | HD64336010GFY | HD64336010G(***)FY | LQFP-48 (FP-48B) |
| | | | HD64336010GFT | HD64336010G(***)FT | QFN-48(TNP-48) |

**Legend**

POR & LVDC: Power-on reset and low-voltage detection circuits

(***): ROM code

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 12 Watchdog Timer<br>12.2.1 Timer Control/Status Register WD (TCSRWD) | 168 | |

| Bit | Bit Name | Description |
|---|---|---|
| 4 | TCSRWE | Timer Control/Status Register WD Write Enable |

| | | |
|---|---|---|
| Section 14 A/D Converter<br>14.3.1 A/D Data Registers A to D (ADDRA to ADDRD) | 220 | Therefore byte access to ADDR should be done by reading the upper byte first then the lower one. Word access is also possible. ADDR is initialized to H'0000. |
| Section 18 Electrical Characteristics<br>Table 18.2 DC Characteristics (1) | 254 | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min |
|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | PB3 to PB0 | $V_{CC}$ = 4.0 V to 5.5 V | $V_{CC}$ × 0.7 |
| | | | | $V_{CC}$ × 0.8 |
| Input low voltage | $V_{IL}$ | RXD, RXD_2, RXD_3*[1], P12 to P10, P17 to P14, : PB3 to PB0 | $V_{CC}$ = 4.0 V to 5.5 V | –0.3 |

| | | |
|---|---|---|
| Table 18.2 DC Characteristics (1) | 257 | |

| Mode | $\overline{RES}$ Pin | Internal State |
|---|---|---|
| Active mode 1 | $V_{CC}$ | Operates |
| Active mode 2 | | Operates (φOSC/64) |
| Sleep mode 1 | $V_{CC}$ | Only timers operate |
| Sleep mode 2 | | Only timers operate (φOSC/64) |