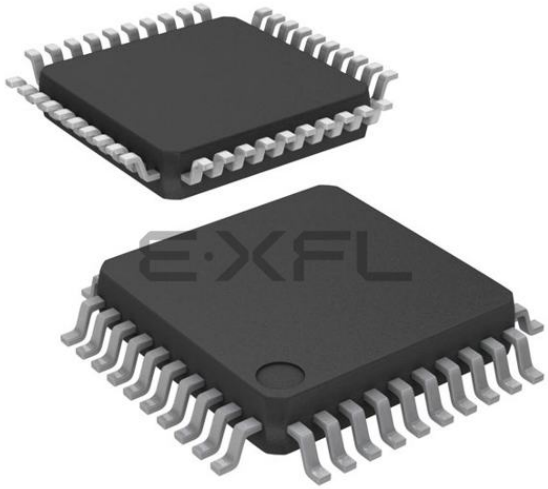


Welcome to E-XFL.COM

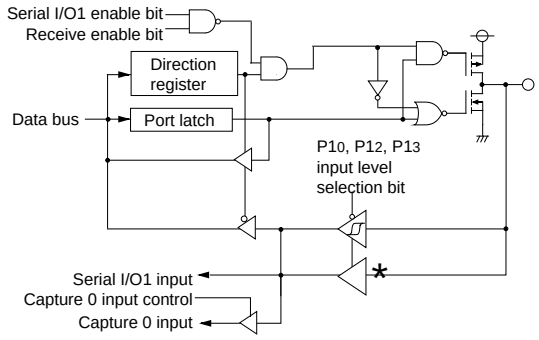
What is "Embedded - Microcontrollers"?

"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.

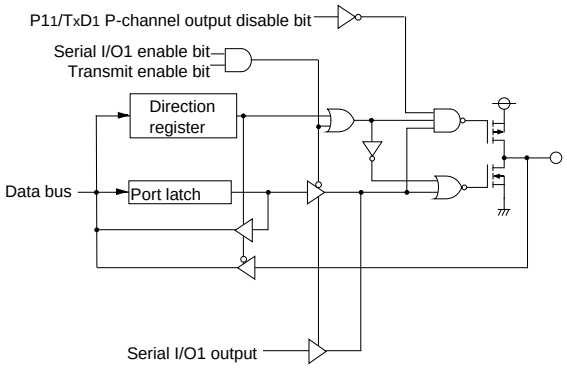
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	740
Core Size	8-Bit
Speed	8MHz
Connectivity	SIO, UART/USART
Peripherals	POR, WDT
Number of I/O	25
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/m37542f8gp-u0

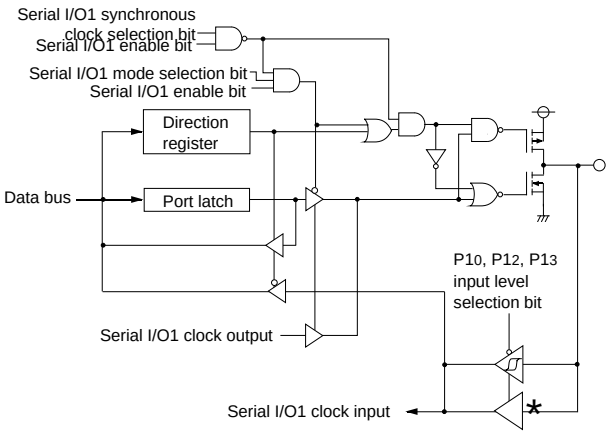
(8) Port P1$_0$



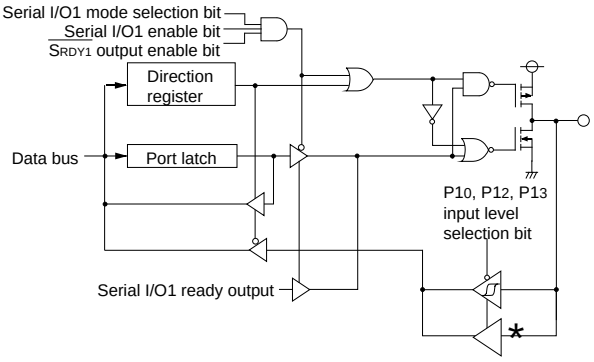
(9) Port P1$_1$



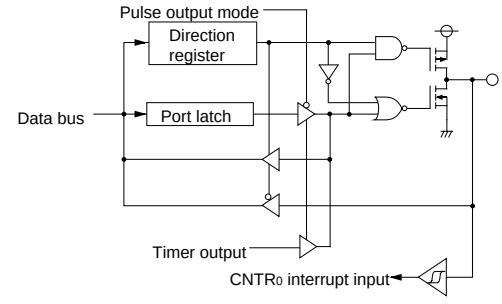
(10) Port P1$_2$



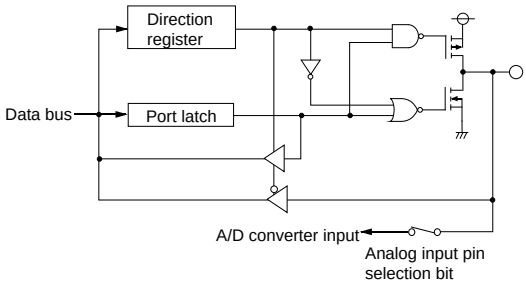
(11) Port P1$_3$



(12) Port P1$_4$



(13) Ports P2$_0$–P2$_7$



* P10, P12, P13, P36, and P37 input level are switched to the CMOS/TTL level by the port P1P3 control register. When the TTL level is selected, there is no hysteresis characteristics.

Fig. 21 Block diagram of ports (2)

• Interrupt Request Generation, Acceptance, and Handling

Interrupts have the following three phases.

(i) Interrupt Request Generation
An interrupt request is generated by an interrupt source (external interrupt signal input, timer underflow, etc.) and the corresponding request bit is set to “1”.

(ii) Interrupt Request Acceptance
Based on the interrupt acceptance timing in each instruction cycle, the interrupt control circuit determines acceptance conditions (interrupt request bit, interrupt enable bit, and interrupt disable flag) and interrupt priority levels for accepting interrupt requests. When two or more interrupt requests are generated simultaneously, the highest priority interrupt is accepted. The value of the interrupt request bit for an unaccepted interrupt remains the same and acceptance is determined at the next interrupt acceptance timing point.

(iii) Handling of Accepted Interrupt Request
The accepted interrupt request is processed.

Figure 25 shows the time up to execution in the interrupt processing routine, and Figure 26 shows the interrupt sequence.
Figure 27 shows the timing of interrupt request generation, interrupt request bit, and interrupt request acceptance.

• Interrupt Handling Execution

When interrupt handling is executed, the following operations are performed automatically.

(1) Once the currently executing instruction is completed, an interrupt request is accepted.
(2) The contents of the program counters and the processor status register at this point are pushed onto the stack area in order from 1 to 3.
1.High-order bits of program counter (PCH)
2.Low-order bits of program counter (PCL)
3.Processor status register (PS)
(3) Concurrently with the push operation, the jump address of the corresponding interrupt (the start address of the interrupt processing routine) is transferred from the interrupt vector to the program counter.
(4) The interrupt request bit for the corresponding interrupt is set to “0”. Also, the interrupt disable flag is set to “1” and multiple interrupts are disabled.
(5) The interrupt routine is executed.
(6) When the RTI instruction is executed, the contents of the registers pushed onto the stack area are popped off in the order from 3 to 1. Then, the routine that was before running interrupt processing resumes.

As described above, it is necessary to set the stack pointer and the jump address in the vector area corresponding to each interrupt to execute the interrupt processing routine.

■ Notes on Interrupts

When setting the followings, the interrupt request bit may be set to “1”.

•When switching external interrupt active edge
Related registers:
Interrupt edge selection register (address $003A_{16}$)
Timer X mode register (address $002B_{16}$)
Capture mode register (address 0020_{16})

When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

① Set the corresponding interrupt enable bit to “0” (disabled).
② Set the interrupt edge select bit (active edge switch bit, trigger mode bit).
③ Set the corresponding interrupt request bit to “0” after 1 or more instructions have been executed.
④ Set the corresponding interrupt enable bit to “1” (enabled).

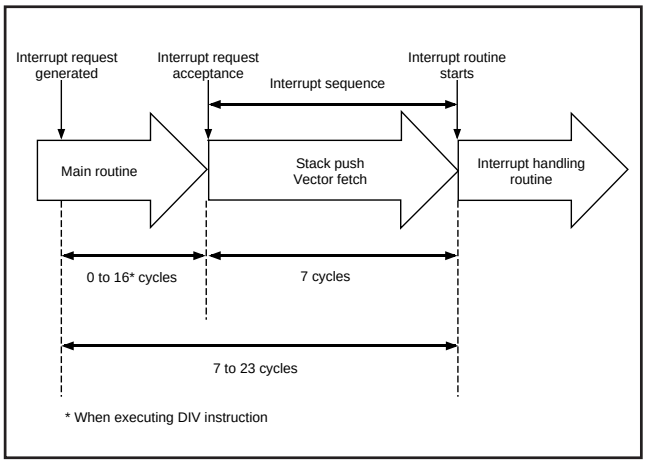


Fig. 25 Time up to execution in interrupt routine

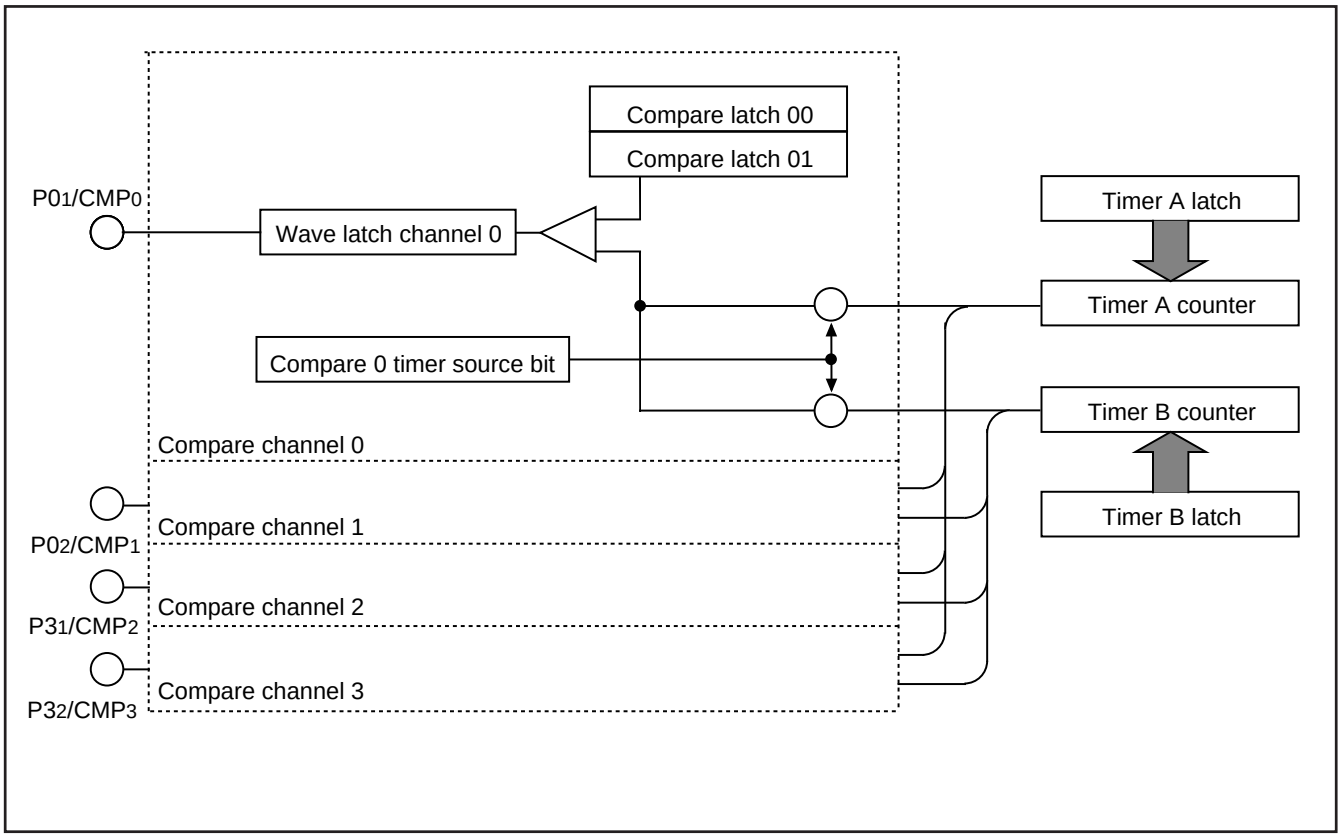


Fig. 42 Block diagram of output compare

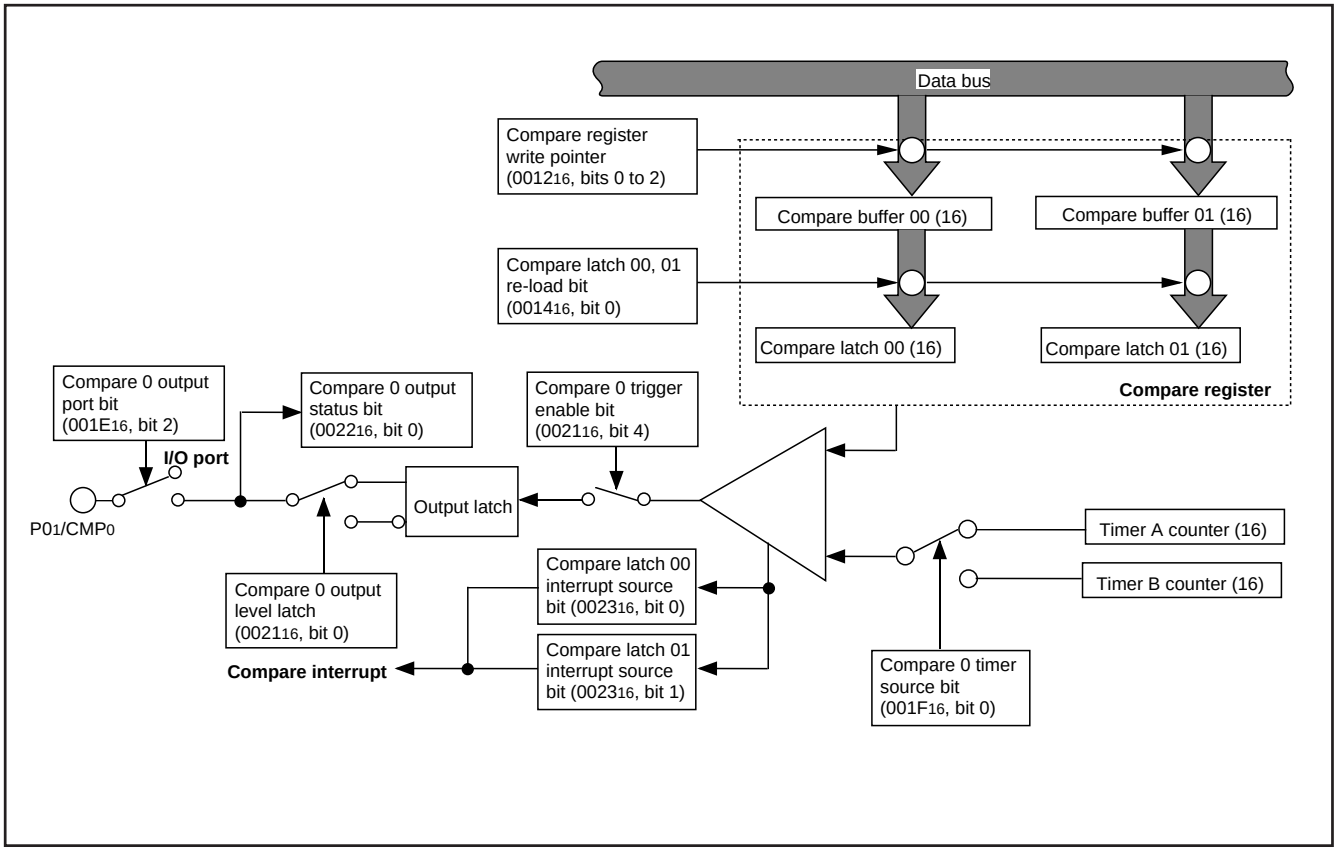


Fig. 43 Block diagram of compare channel 0

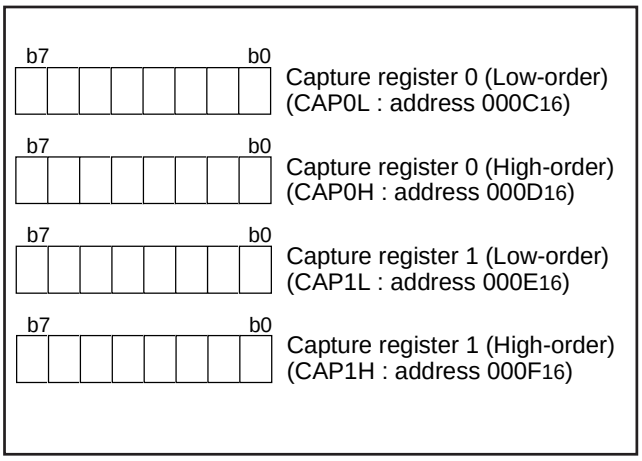


Fig. 49 Structure of capture software trigger register

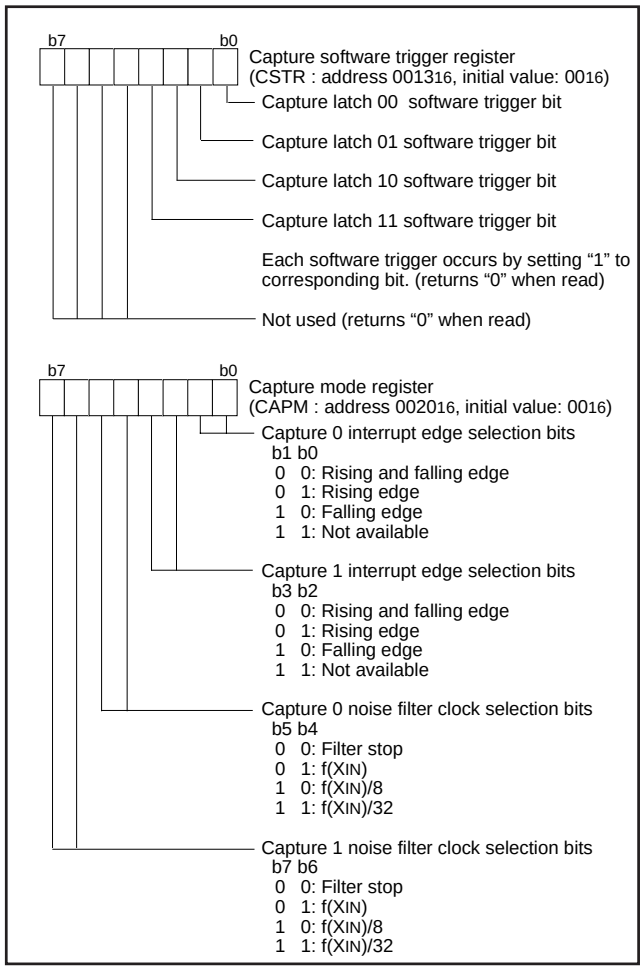


Fig. 50 Structure of capture software trigger register/capture mode register

(2) Asynchronous Serial I/O1 (UART) Mode

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O1 mode selection bit of the serial I/O1 control register to “0”.

Eight serial data transfer formats can be selected, and the transfer formats used by a transmitter and receiver must be identical.

The transmit and receive shift registers each have a buffer, but the two buffers have the same address in memory. Since the shift register cannot be written to or read from directly, transmit data is written to the transmit buffer register, and receive data is read from the receive buffer register.

The transmit buffer register can also hold the next data to be transmitted, and the receive buffer register can hold a character while the next character is being received.

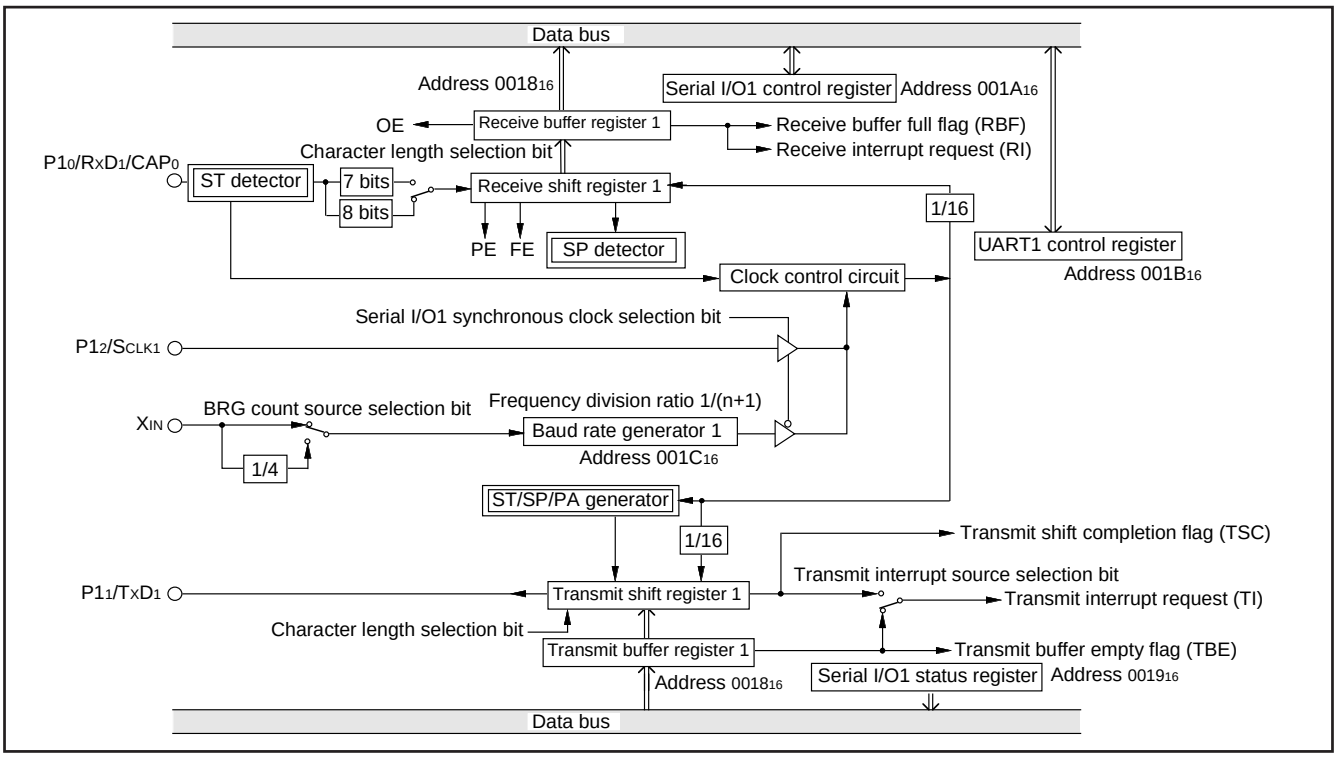
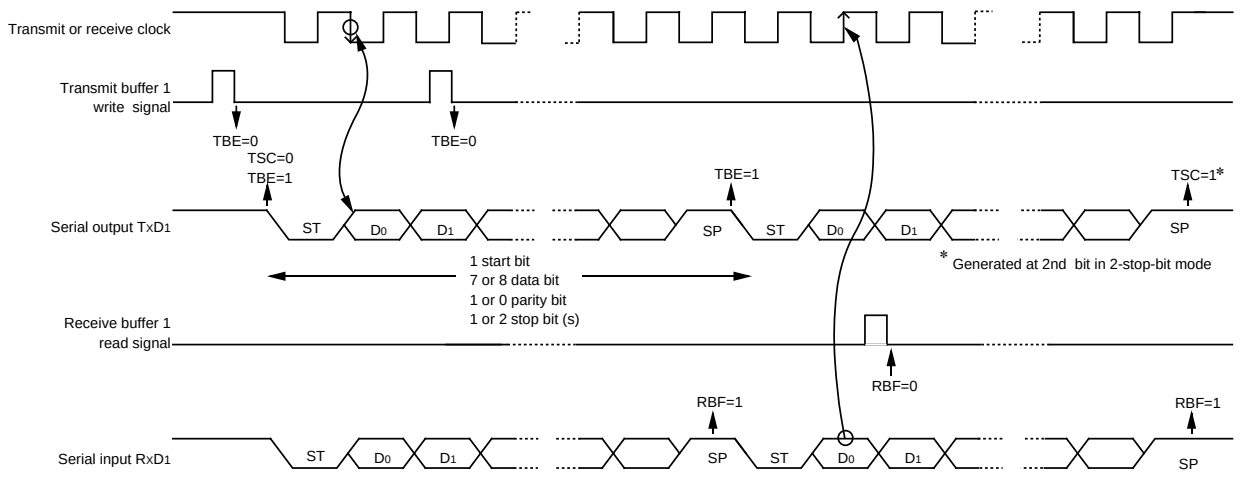


Fig. 57 Block diagram of UART serial I/O1



Notes
1: Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
2: As the transmit interrupt (TI), when either the TBE or TSC flag becomes “1,” can be selected to occur depending on the setting of the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.
3: The receive interrupt (RI) is set when the RBF flag becomes “1.”
4: After data is written to the transmit buffer when TSC=1, 0.5 to 1.5 cycles of the data shift cycle is necessary until changing to TSC=0.

Fig. 58 Operation of UART serial I/O1 function

Bus collision detection (SIO1)

SIO1 can detect a bus collision by setting UART1 bus collision detection interrupt enable bit.

When transmission is started in the clock synchronous or asynchronous (UART) serial I/O mode, the transmit pin TxD_1 is compared with the receive pin RxD_1 in synchronization with rising edge of transmit shift clock. If they do not coincide with each other, a bus collision detection interrupt request occurs.

When a transmit data collision is detected between LSB and MSB of transmit data in the clock synchronous serial I/O mode or between the start bit and stop bit of transmit data in UART mode, a bus collision detection can be performed by both the internal clock and the external clock.

A block diagram is shown in Fig. 61.
A timing diagram is shown in Fig. 62.

Note: Bus collision detection can be used when SIO1 is operating at full-duplex communication. When SIO1 is operating at half-duplex communication, set bus collision detection interrupt to be disabled.

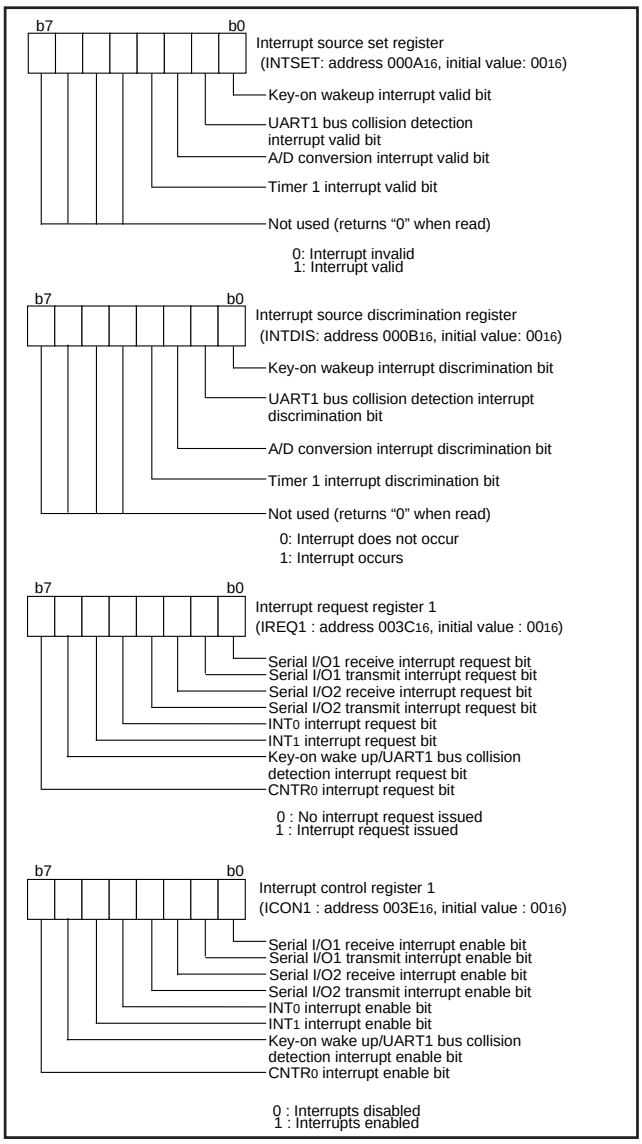


Fig. 60 Bus collision detection circuit related registers

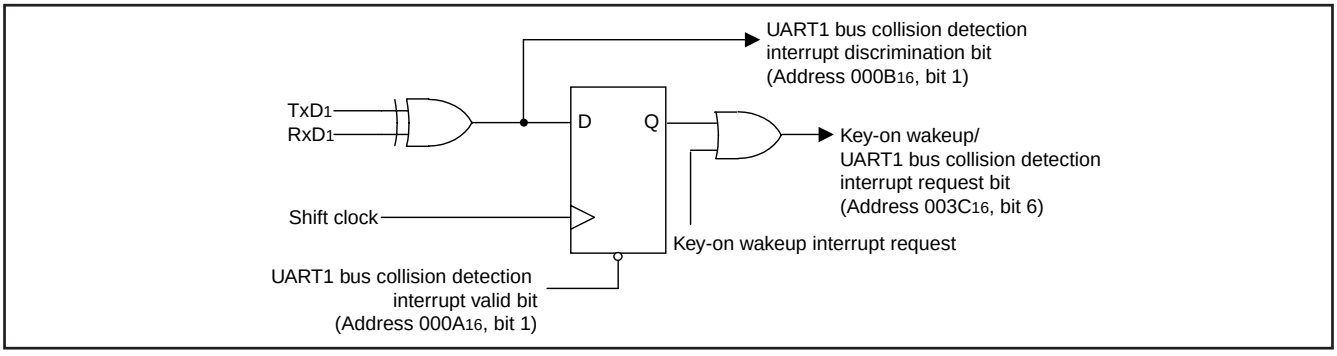


Fig. 61 Block diagram of bus collision detection interrupt circuit

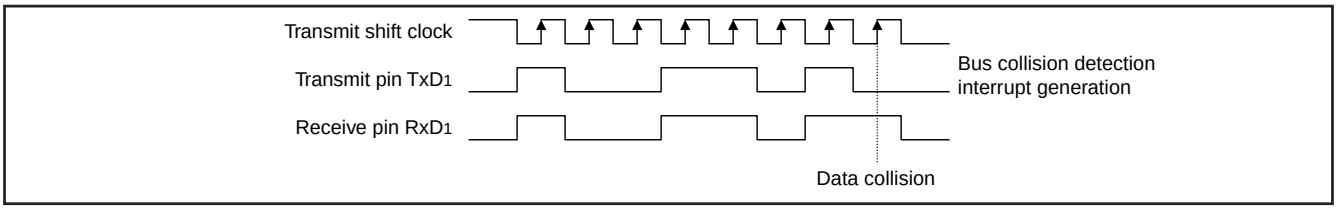


Fig. 62 Timing diagram of bus collision detection interrupt

(2) Asynchronous Serial I/O2 (UART) Mode

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O2 mode selection bit of the serial I/O2 control register to “0”.

Eight serial data transfer formats can be selected, and the transfer formats used by a transmitter and receiver must be identical.

The transmit and receive shift registers each have a buffer, but the two buffers have the same address in memory. Since the shift register cannot be written to or read from directly, transmit data is written to the transmit buffer register, and receive data is read from the receive buffer register.

The transmit buffer register can also hold the next data to be transmitted, and the receive buffer register can hold a character while the next character is being received.

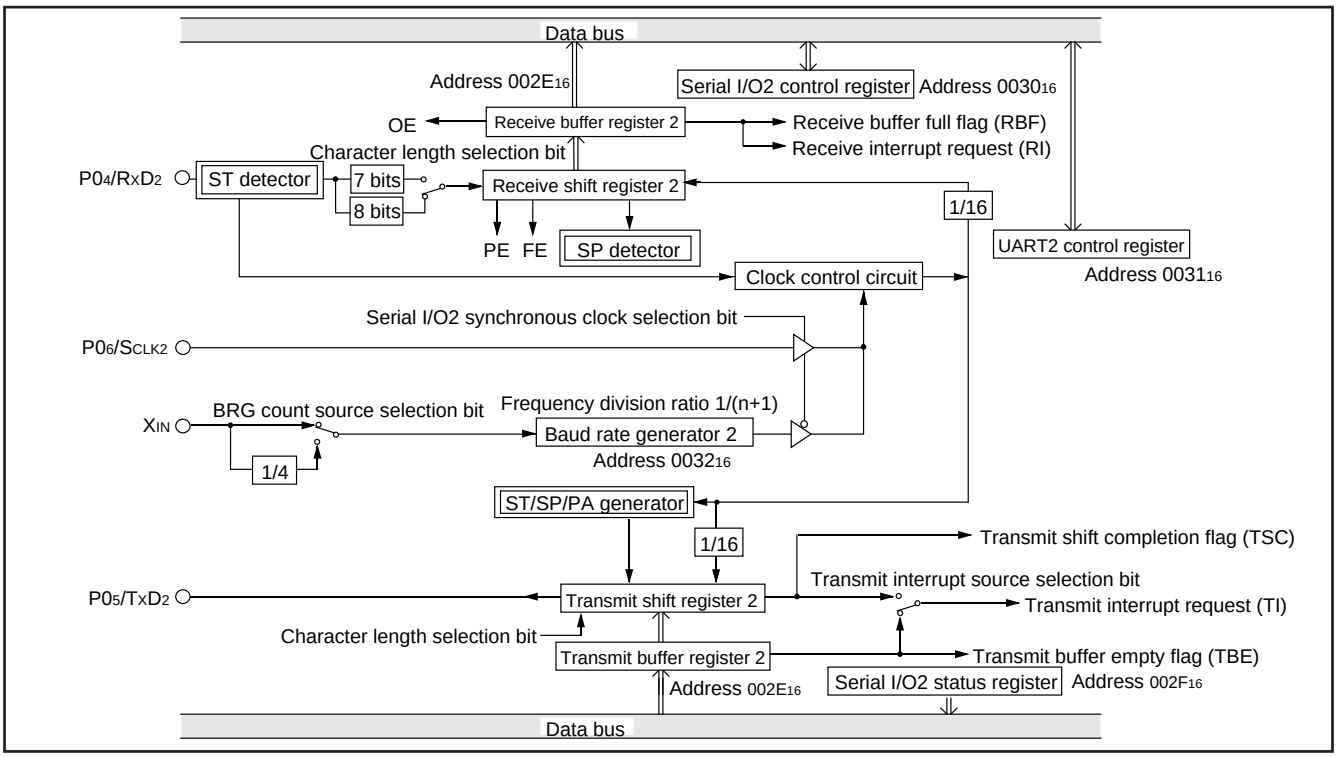
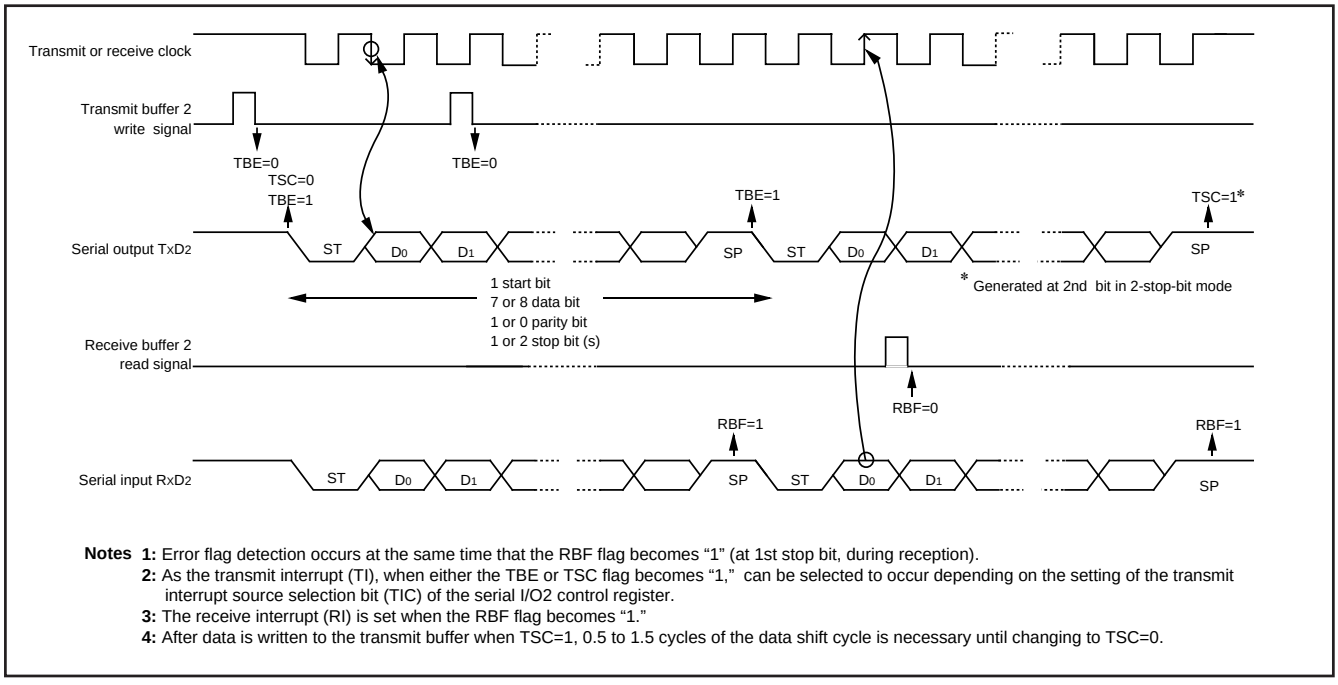


Fig. 65 Block diagram of UART serial I/O2



Notes
1: Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
2: As the transmit interrupt (TI), when either the TBE or TSC flag becomes “1,” can be selected to occur depending on the setting of the transmit interrupt source selection bit (TIC) of the serial I/O2 control register.
3: The receive interrupt (RI) is set when the RBF flag becomes “1.”
4: After data is written to the transmit buffer when TSC=1, 0.5 to 1.5 cycles of the data shift cycle is necessary until changing to TSC=0.

Fig. 66 Operation of UART serial I/O2 function

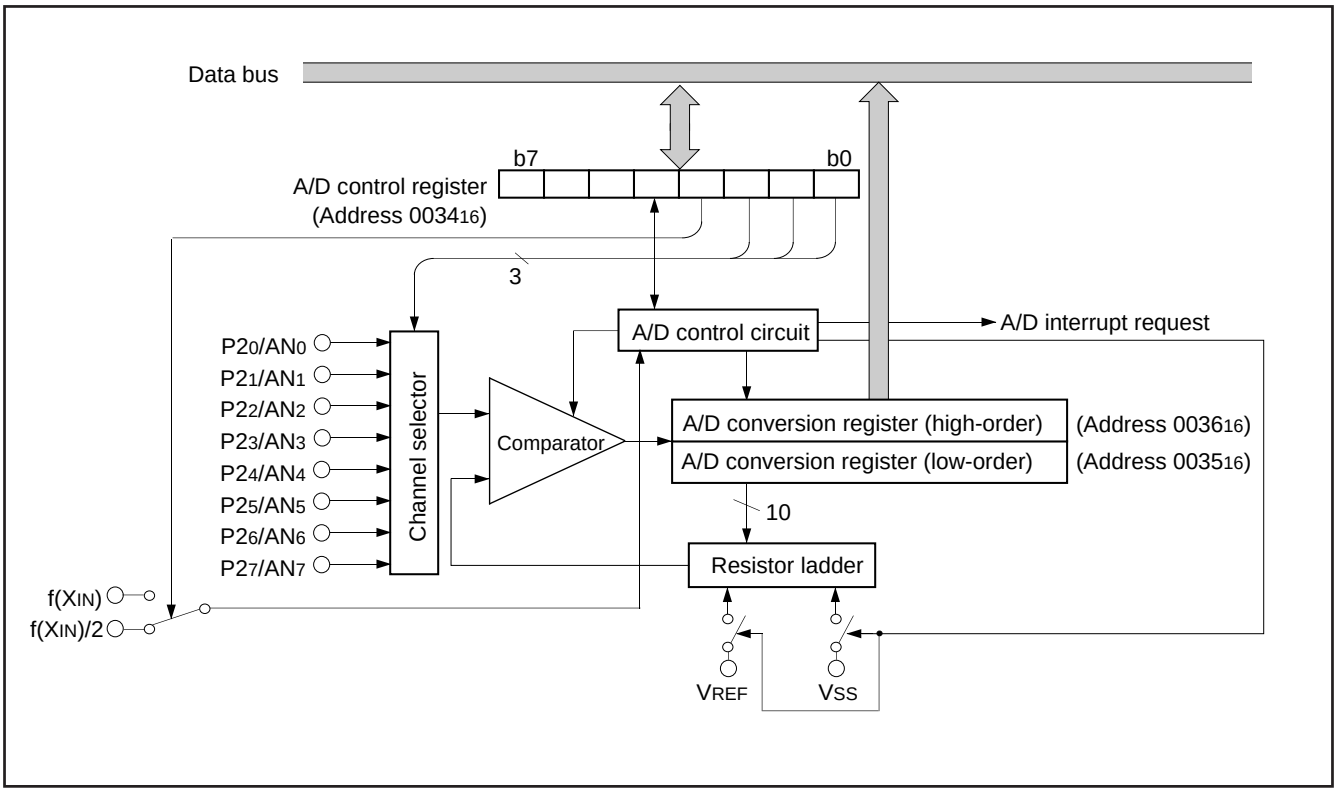


Fig. 70 Block diagram of A/D converter

(1) Oscillation control

• Stop mode

When the STP instruction is executed, the internal clock φ stops at an "H" level and the XIN oscillator stops. At this time, timer 1 is set to "01$_{16}$" and prescaler 1 is set to "FF$_{16}$" when the oscillation stabilization time set bit after release of the STP instruction is "0". On the other hand, timer 1 and prescaler 1 are not set when the above bit is "1". Accordingly, set the wait time fit for the oscillation stabilization time of the oscillator to be used. f(XIN)/16 is forcibly connected to the input of prescaler 1. When an external interrupt is accepted, oscillation is restarted but the internal clock φ remains at "H" until timer 1 underflows. As soon as timer 1 underflows, the internal clock φ is supplied. This is because when a ceramic oscillator is used, some time is required until a start of oscillation. In case oscillation is restarted by reset, no wait time is generated. So apply an "L" level to the $\overline{\text{RESET}}$ pin while oscillation becomes stable, or set the wait time by on-chip oscillator operation after system is released from reset until the oscillation is stabled.

With the FLASH version, the internal power supply circuit is changed to low power consumption mode for consumption current reduction at the time of STP instruction execution.

Although an internal power supply circuit is usually changed to the normal operation mode at the time of the return from an STP instruction, since a certain time is required to start the power supply to FLASH and operation of FLASH to be enabled, set wait time 100 μs or more with the FLASH version by the oscillation stabilization time set function after release of the STP instruction which used the timer 1.

• Wait mode

If the WIT instruction is executed, the internal clock φ stops at an "H" level, but the oscillator does not stop. The internal clock restarts if a reset occurs or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted. To ensure that interrupts will be received to release the STP or WIT state, interrupt enable bits must be set to "1" before the STP or WIT instruction is executed.

■ Notes on Clock Generating Circuit

For use with the oscillation stabilization set bit after release of the STP instruction set to "1", set values in timer 1 and prescaler 1 after fully appreciating the oscillation stabilization time of the oscillator to be used.

• Switch of ceramic and RC oscillations

After releasing reset the operation starts by starting an on-chip oscillator. Then, a ceramic oscillation or an RC oscillation is selected by setting bit 5 of the CPU mode register.

• Double-speed mode

When a ceramic oscillation is selected, a double-speed mode can be used. Do not use it when an RC oscillation is selected.

• CPU mode register

Bits 5, 1 and 0 of CPU mode register are used to select oscillation mode and to control operation modes of the microcomputer. In order to prevent the dead-lock by error-writing (ex. program run-away), these bits can be rewritten only once after releasing reset. After rewriting it is disable to write any data to the bit. (The emulator MCU "M37542RSS" is excluded.)

Also, when the read-modify-write instructions (SEB, CLB) are executed to bits 2 to 4, 6 and 7, bits 5, 1 and 0 are locked.

• Clock division ratio, XIN oscillation control, on-chip oscillator control

The state transition shown in Fig. 84 can be performed by setting the clock division ratio selection bits (bits 7 and 6), XIN oscillation control bit (bit 4), on-chip oscillator oscillation control bit (bit 3) of CPU mode register. Be careful of notes on use in Fig. 84.

- Count source (Timer 1, Timer A, Timer B, Timer X, Serial I/O, Serial I/O2, A/D converter, Watchdog timer)
  A count source of watchdog timer is affected by the clock division selection bit of the CPU mode register.
  The f(XIN) clock is supplied to the watchdog timer when selecting f(XIN) as the CPU clock.
  The on-chip oscillator output is supplied to the watchdog timer when selecting the on-chip oscillator output as the CPU clock.

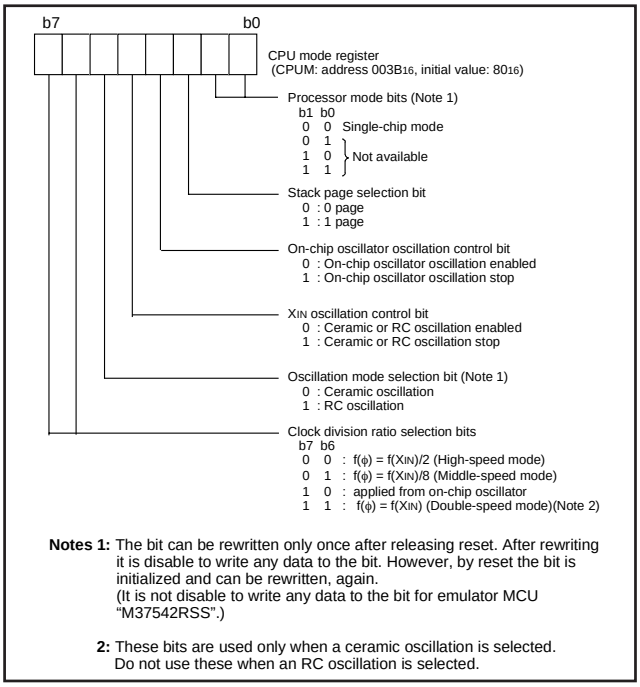
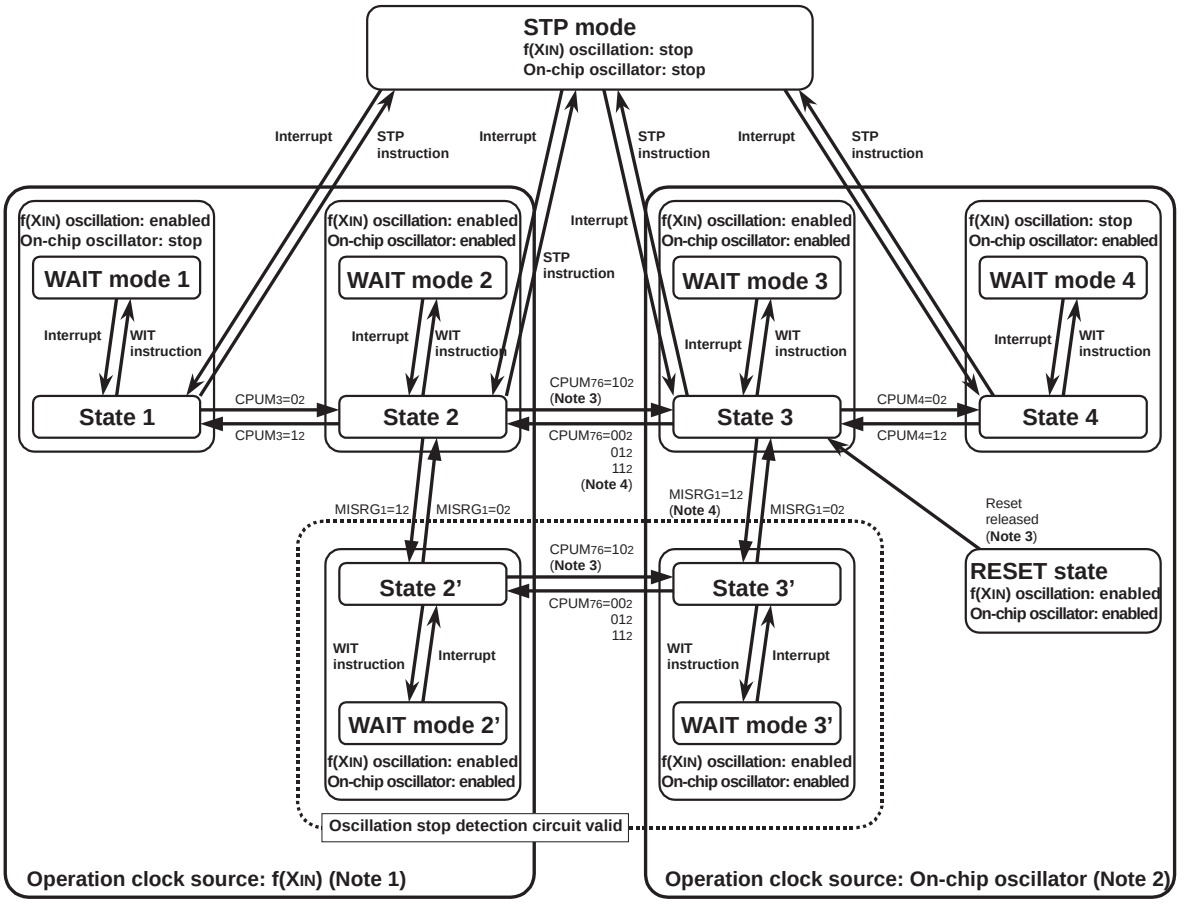


Fig. 80 Structure of CPU mode register



Notes on switch of clock

(1) In operation clock = f(XIN), the following can be selected for the CPU clock division ratio.
f(XIN)/2 (high-speed mode)
f(XIN)/8 (middle-speed mode)
f(XIN) (double-speed mode, only at a ceramic oscillation)

(2) In operation clock = On-chip oscillator, the following can be selected for the CPU clock division ratio.
ROSC/1 (On-chip oscillator double-speed mode)
ROSC/2 (On-chip oscillator high-speed mode)
ROSC/8 (On-chip oscillator middle-speed mode)
ROSC/128 (On-chip oscillator low-speed mode)

(3) After system is released from reset, and state transition of state 2 → state 3 and state transition of state 2' → state 3', ROSC/8 (On-chip oscillator middle-speed mode) is selected for CPU clock.

(4) Executing the state transition state 3 to 2 or state 3 to 3' after stabilizing XIN oscillation.

(5) When the state 2 → state 3 → state 4 is performed, execute the NOP instruction as shown below according to the division ratio of CPU clock.
1. CPUM76 = 102 (state 2 → state 3)
2. NOP instruction
Transition from Double-speed mode: NOP × 3
Transition from High-speed mode: NOP × 1
Transition from Middle-speed mode: NOP × 0
3. CPU4 = 12 (state 3 → state 4)

(6) When the state 3 → state 2 → state 1 is performed, execute the NOP instruction as shown below according to the division ratio of CPU clock.
1. CPUM76 = 002 or 012 or 112 (state 3 → state 2)
2. NOP instruction
Transition from On-chip oscillator double-speed mode: NOP × 4
Transition from On-chip oscillator high-speed mode: NOP × 2
Transition from On-chip oscillator middle-speed mode: NOP × 0
Transition from On-chip oscillator low-speed mode: NOP × 0
3. CPUM3 = 12 (state 2 → state 1)

Fig. 84 State transition

2. Connection of bypass capacitor across Vss line and Vcc line
Connect an approximately 0.1 μF bypass capacitor across the Vss line and the Vcc line as follows:

- Connect a bypass capacitor across the Vss pin and the Vcc pin at equal length.
- Connect a bypass capacitor across the Vss pin and the Vcc pin with the shortest possible wiring.
- Use lines with a larger diameter than other signal lines for Vss line and Vcc line.
- Connect the power source wiring via a bypass capacitor to the Vss pin and the Vcc pin.

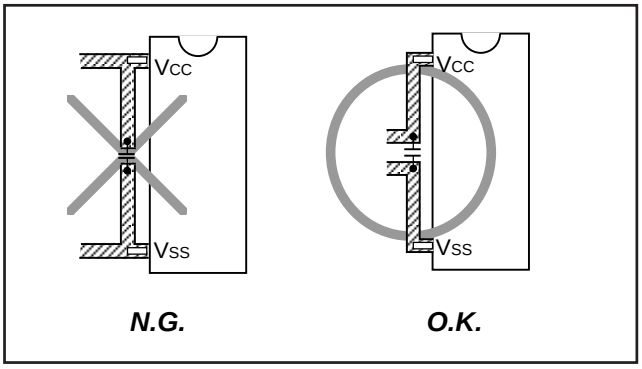


Fig. 91 Bypass capacitor across the Vss line and the Vcc line

3. Wiring to analog input pins

- Connect an approximately 100 Ω to 1 kΩ resistor to an analog signal line which is connected to an analog input pin in series. Besides, connect the resistor to the microcomputer as close as possible.
- Connect an approximately 1000 pF capacitor across the Vss pin and the analog input pin. Besides, connect the capacitor to the Vss pin as close as possible. Also, connect the capacitor across the analog input pin and the Vss pin at equal length.

<Reason>
Signals which is input in an analog input pin (such as an A/D converter/comparator input pin) are usually output signals from sensor. The sensor which detects a change of event is installed far from the printed circuit board with a microcomputer, the wiring to an analog input pin is longer necessarily. This long wiring functions as an antenna which feeds noise into the microcomputer, which causes noise to an analog input pin.

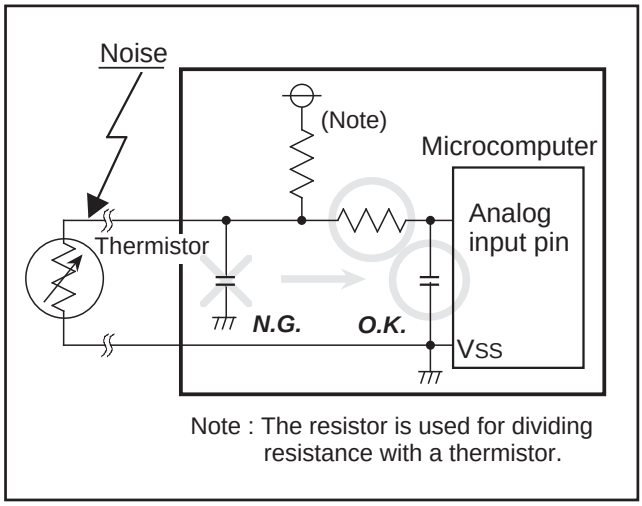


Fig. 92 Analog signal line and a resistor and a capacitor

- The analog input pin is connected to the capacitor of a voltage comparator. Accordingly, sufficient accuracy may not be obtained by the charge/discharge current at the time of A/D conversion when the analog signal source of high-impedance is connected to an analog input pin. In order to obtain the A/D conversion result stabilized more, please lower the impedance of an analog signal source, or add the smoothing capacitor to an analog input pin.

[Flash memory control registers (FMCR0 to FMCR2)]
0FE0$_{16}$ to 0FE2$_{16}$

Figure 98 shows the flash memory control register 0.

Bit 0 of the flash memory control register 0 is the RY/$\overline{BY}$ status flag used exclusively to read the operating status of the flash memory. During programming and erase operations, it is "0" (busy). Otherwise, it is "1" (ready).

Bit 1 of the flash memory control register 0 is the CPU rewrite mode select bit. When this bit is set to "1", the MCU enters CPU rewrite mode. And then, software commands can be accepted. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, use the control program in the internal RAM for write to bit 1. To set this bit 1 to "1", it is necessary to write "0" and then write "1" in succession to bit 1. The bit can be set to "0" by only writing "0".

Bit 2 of the flash memory control register 0 is the 8KB user block E/W mode enable bit. By setting this bit in combination with bit 4 (all user block E/W enable bit) of flash memory control register 2 (address 0FE0$_{16}$), Erase/Write to user block in CPU rewrite mode is disabled.

Bit 3 of the flash memory control register 0 is the flash memory reset bit used to reset the control circuit of internal flash memory. This bit is used when exiting CPU rewrite mode and when flash memory access has failed. When the CPU rewrite mode select bit is "1", setting "1" for this bit resets the control circuit. To release the reset, it is necessary to set this bit to "0".

Bit 5 of the flash memory control register 0 is the User ROM area select bit and is valid only in the boot mode. Setting this bit to "1" in the boot mode switches an accessible area from the boot ROM area to the user ROM area. To use the CPU rewrite mode in the boot mode, set this bit to "1". Note that when the microcomputer is booted up in the user ROM area, only the user ROM area is accessible and bit 5 is invalid; on the other hand, when the microcomputer is in the boot mode, bit 5 is valid independent of the CPU rewrite mode. To rewrite bit 5, execute the user-original reprogramming control software transferred to the internal RAM in advance.

Bit 6 of the flash memory control register 0 is the program status flag. This bit is set to "1" when writing to flash memory is failed. When program error occurs, the block cannot be used.

Bit 7 of the flash memory control register 0 is the erase status flag. This bit is set to "1" when erasing flash memory is failed. When erase error occurs, the block cannot be used.

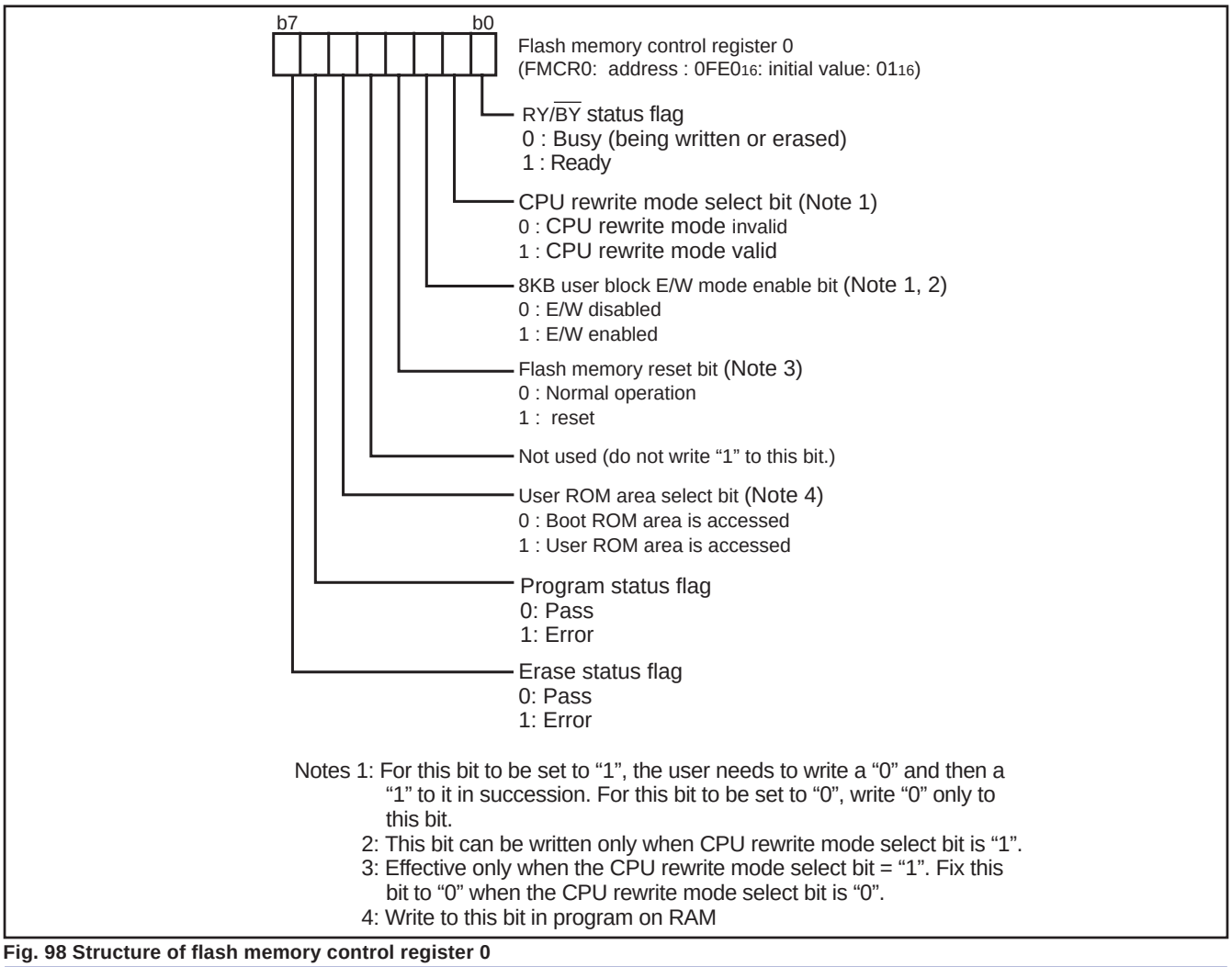


Fig. 98 Structure of flash memory control register 0

● Software Commands

Table 11 lists the software commands.
After setting the CPU rewrite mode select bit to "1", execute a software command to specify an erase or program operation.
Each software command is explained below.

• Read Array Command (FF_{16})

The read array mode is entered by writing the command code "FF_{16}" in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the contents of the specified address are read out at the data bus (D_0 to D_7).
The read array mode is retained until another command is written.

• Read Status Register Command (70_{16})

When the command code "70_{16}" is written in the first bus cycle, the contents of the status register are read out at the data bus (D_0 to D_7) by a read in the second bus cycle.
The status register is explained in the next section.

• Clear Status Register Command (50_{16})

This command is used to clear the bits SR4 and SR5 of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code "50_{16}" in the first bus cycle.

• Program Command (40_{16})

Program operation starts when the command code "40_{16}" is written in the first bus cycle. Then, if the address and data to program are written in the 2nd bus cycle, program operation (data programming and verification) will start.
Whether the write operation is completed can be confirmed by read status register or the RY/$\overline{BY}$ status flag. When the program starts, the read status register mode is entered automatically and the contents of the status register is read at the data bus (D_0 to D_7). The status register bit 7 (SR7) is set to "0" at the same time the write operation starts and is returned to "1" upon completion of the write operation. In this case, the read status register mode remains active until the read array command (FF_{16}) is written.

The RY/$\overline{BY}$ status flag of the flash memory control register is "0" during write operation and "1" when the write operation is completed as is the status register bit 7.
At program end, program results can be checked by reading the status register.

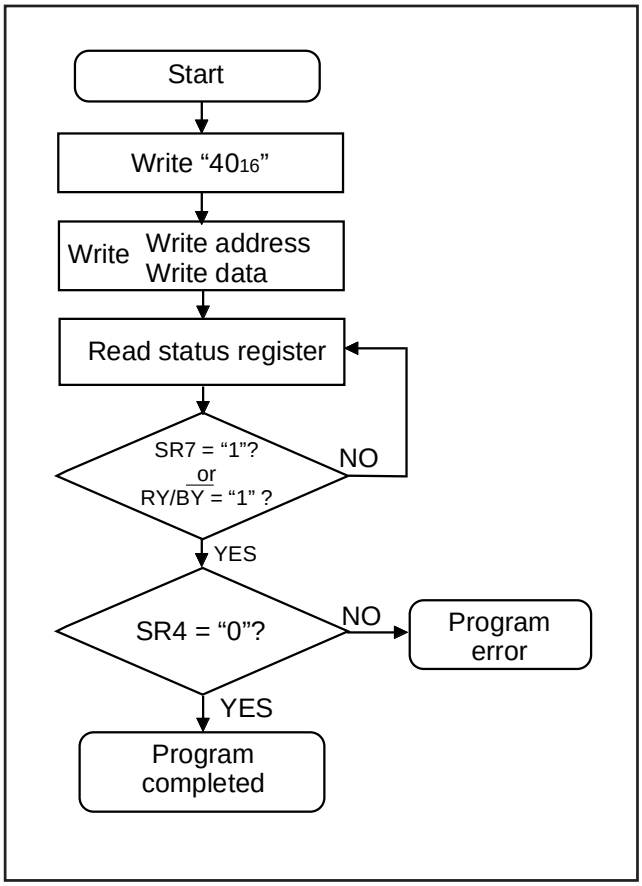


Fig. 102 Program flowchart

Table 11 List of software commands (CPU rewrite mode)

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read array	Write	× (Note 4)	FF_{16}			
Read status register	Write	×	70_{16}	Read	×	SRD (Note 1)
Clear status register	Write	×	50_{16}			
Program	Write	×	40_{16}	Write	WA (Note 2)	WD (Note 2)
Block erase	Write	×	20_{16}	Write	BA (Note 3)	$D0_{16}$

SRD = Status Register Data
WA = Write Address, WD = Write Data
BA = Block Address to be erased (Input the maximum address of each block.)
× = × denotes a given address in the user ROM area.

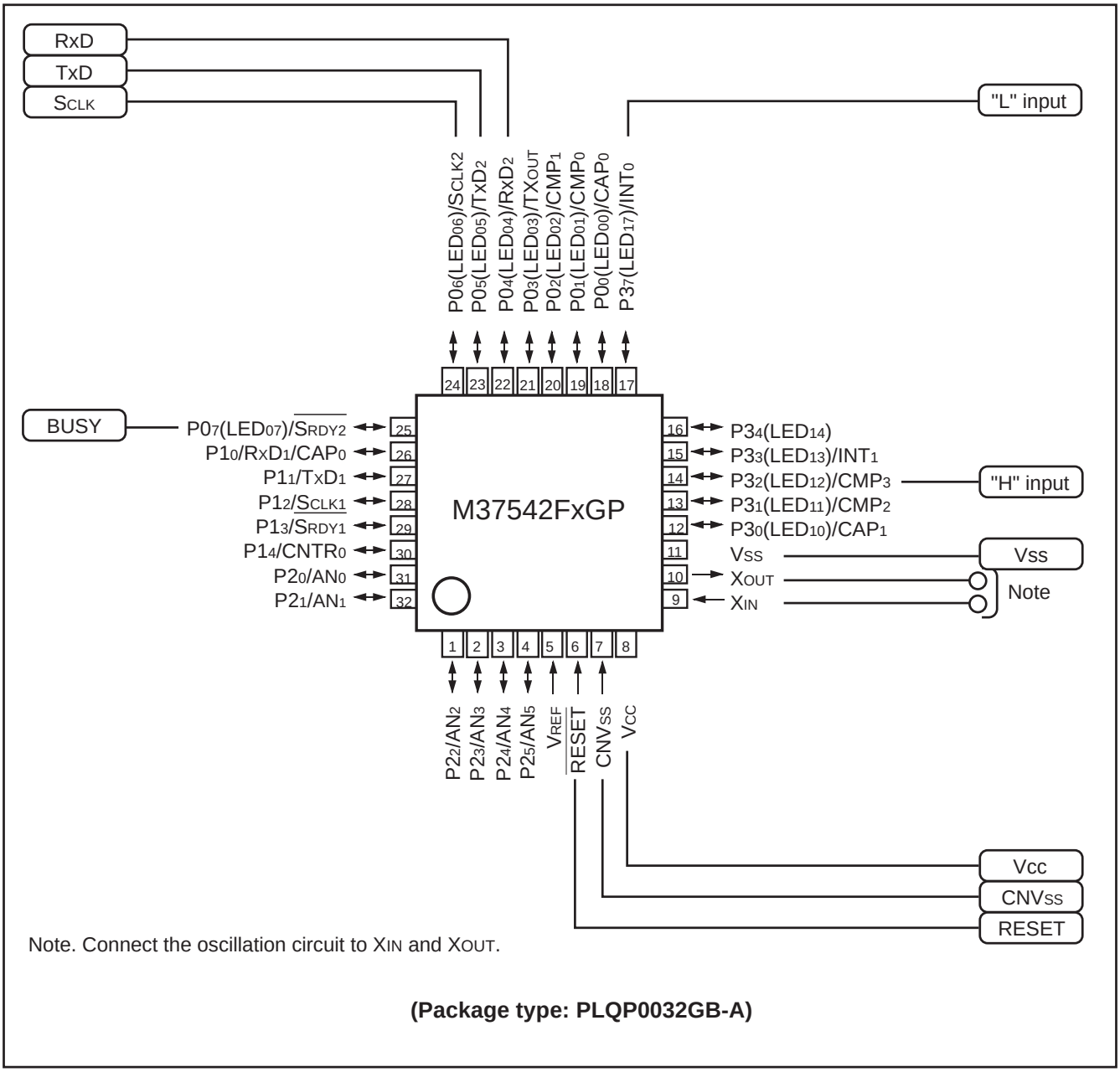
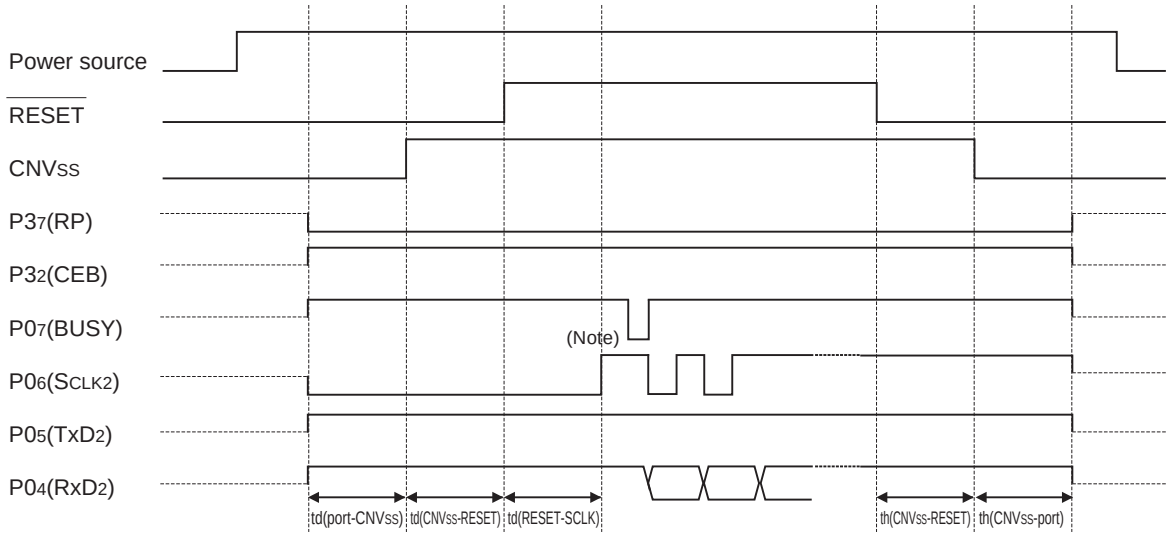


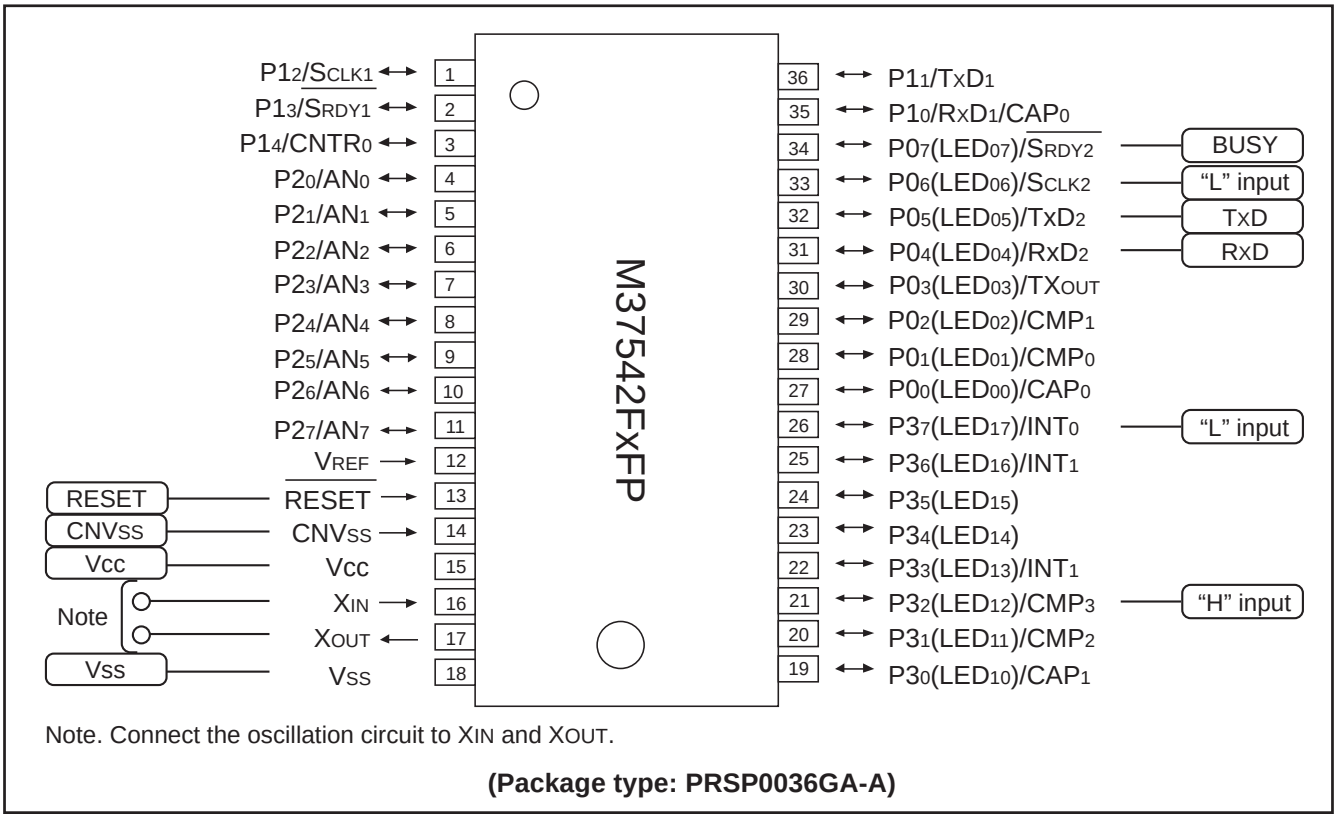
Fig. 107 Pin connection diagram in standard serial I/O mode 1 (PLQP0032GB-A package)



Symbol	Ratings			Unit
	Min.	Typ.	Max.	
td(port-CNVss)	1	-	-	ms
td(CNVss-RESET)	1	-	-	ms
td(RESET-SCLK)	0.05	-	0.5	ms
th(RESET-CNVss)	1	-	-	ms
th(CNVss-port)	1	-	-	ms

Note: Keep input of P06 "H" until P07 turns "L".

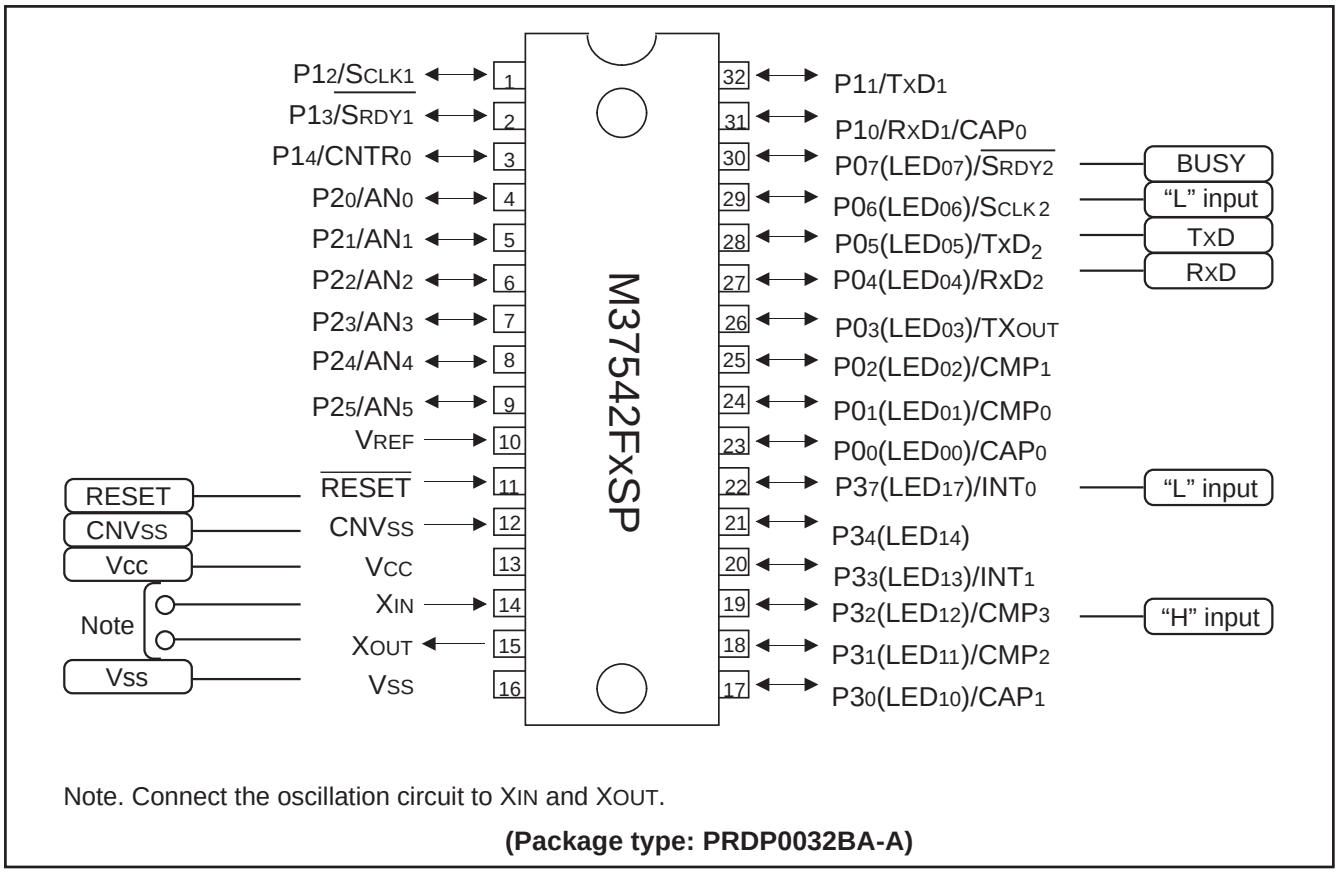
Fig. 111 Timing diagram in standard serial I/O mode 1



Note. Connect the oscillation circuit to XIN and XOUT.

(Package type: PRSP0036GA-A)

Fig. 113 Pin connection diagram in standard serial I/O mode 2 (PRSP0036GA-A package)



Note. Connect the oscillation circuit to XIN and XOUT.

(Package type: PRDP0032BA-A)

Fig. 114 Pin connection diagram in standard serial I/O mode 2 (PRDP0032BA-A package)

Table 24 Timing requirements (3)

(Mask ROM version: Vcc = 2.2 to 5.5 V, Vss = 0 V, Ta = –20 to 85 °C, unless otherwise noted) (This is only for the mask ROM version.)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	2			μs
$t_{C(XIN)}$	External clock input cycle time	500			ns
$t_{WH(XIN)}$	External clock input “H” pulse width	200			ns
$t_{WL(XIN)}$	External clock input “L” pulse width	200			ns
$t_{C(CNTR0)}$	CNTR0 input cycle time	1000			ns
$t_{WH(CNTR0)}$	CNTR0, INT0, INT1, CAP0, CAP1 input “H” pulse width (Note 1)	460			ns
$t_{WL(CNTR0)}$	CNTR0, INT0, INT1, CAP0, CAP1 input “L” pulse width (Note 1)	460			ns
$t_{C(SCLK1)}$	Serial I/O1, serial I/O2 clock input cycle time (Note 2)	4000			ns
$t_{WH(SCLK1)}$	Serial I/O1, serial I/O2 clock input “H” pulse width (Note 2)	1900			ns
$t_{WL(SCLK1)}$	Serial I/O1, serial I/O2 clock input “L” pulse width (Note 2)	1900			ns
$t_{su(RxD1–SCLK1)}$	Serial I/O1, serial I/O2 input set up time	800			ns
$t_{h(SCLK1–RxD1)}$	Serial I/O1, serial I/O2 input hold time	400			ns

Notes 1: As for CAP0, CAP1, it is the value when noise filter is not used.

2: In this time, bit 6 of the serial I/O1 control register (address $001A_{16}$) is set to “1” (clock synchronous serial I/O is selected).
When bit 6 of the serial I/O1 control register is “0” (clock asynchronous serial I/O1 is selected), the rating values are divided by 4.
In this time, bit 6 of the serial I/O2 control register (address 0030_{16}) is set to “1” (clock synchronous serial I/O is selected).
When bit 6 of the serial I/O2 control register is “0” (clock asynchronous serial I/O is selected), the rating values are divided by 4.

3. Interrupt discrimination bit
Use an LDM instruction to clear to “0” an interrupt discrimination bit.
LDM #%0000XXXX, $0B
Set the following values to “X”
“0”: an interrupt discrimination bit to clear
“1”: other interrupt discrimination bits
Ex.) When a key-on wakeup interrupt discrimination bit is cleared;
LDM #%00001110 and $0B.

4. Interrupt discrimination bit and interrupt request bit
For key-on wakeup, UART1 bus collision detection, A/D conversion and Timer 1 interrupt, even if each interrupt valid bit (interrupt source set register (address $0A_{16}$)) is set “0: Invalid”, each interrupt discrimination bit (interrupt source discrimination register (address $0B_{16}$)) is set to “1: interrupt occurs” when corresponding interrupt request occurs.
But corresponding interrupt request bit (interrupt request registers 1, 2 (addresses $3C_{16}$, $3D_{16}$) is not affected.

Notes on Timers

1. When n (0 to 255) is written to a timer latch, the frequency division ratio is 1/(n+1).

2. When a count source of timer X, timer A or timer B is switched, stop a count of the timer.

Notes on Timer X

1. $CNTR_0$ interrupt active edge selection
$CNTR_0$ interrupt active edge depends on the $CNTR_0$ active edge switch bit (bit 2 of timer X mode register (address $2B_{16}$)).
When this bit is “0”, the $CNTR_0$ interrupt request bit is set to “1” at the falling edge of $CNTR_0$ pin input signal. When this bit is “1”, the $CNTR_0$ interrupt request bit is set to “1” at the rising edge of $CNTR_0$ pin input signal.

2. Timer X count source selection
The f(X_{IN}) (frequency not divided) can be selected by the timer X count source selection bits (bits 1 and 0 of timer count source set register (address $2A_{16}$)) only when the ceramic oscillation or the on-chip oscillator is selected.
Do not select it for the timer X count source at the RC oscillation.

3. Pulse output mode
Set the direction register of port $P1_4$, which is also used as $CNTR_0$ pin, to output.
When the TX_{OUT} pin is used, set the direction register of port $P0_3$, which is also used as TX_{OUT} pin, to output.

4. Pulse width measurement mode
Set the direction register of port $P1_4$, which is also used as $CNTR_0$ pin, to input.

Notes on Timer A, B

1. Setting of timer value
When “1: Write to only latch” is set to the timer A (B) write control bit, written data to timer register is set to only latch even if timer is stopped or operating. Accordingly, in order to set the initial value for timer when it is stopped, set “0: Write to latch and timer simultaneously” to timer A (B) write control bit.

2. Read/write of timer A
Stop timer A to read/write its data in the following state;
X_{IN} oscillation selected by clock division ratio selection bits (bits 7 and 6 of CPU mode register (address $3B_{16}$)), and the on-chip oscillator output is selected as the timer A count source.

3. Read/write of timer B
Stop timer B to read/write its data in the following state;
X_{IN} oscillation selected by clock division ratio selection bits, the timer A underflow is selected as the timer B count source, and the on-chip oscillator output is selected as the timer A count source.

Notes on Output Compare

1. When the selected source timer of each compare channel is stopped, written data to compare register is loaded to the compare latch simultaneously.

2. Do not write the same data to both of compare latch x0 (x=0, 1, 2, 3) and x1.

3. When setting value of the compare register is larger than timer setting value, compare match signal is not generated. Accordingly, the output waveform is fixed to “L” or “H” level.
   However, when setting value of another compare register is smaller than timer setting value, this compare match signal is generated. Accordingly, if the corresponding compare latch y (y=00, 01, 10, 11, 20, 21, 30, 31) interrupt source bit is set to “1” (valid), compare match interrupt request occurs.

4. When the compare x trigger enable bit is cleared to “0” (disabled), the match trigger to the waveform output circuit is disabled. Accordingly, the output waveform can be fixed to “L” or “H” level.
   However, in this case, the compare match signal is generated. Accordingly, if the corresponding compare latch y (y=00, 01, 10, 11, 20, 21, 30, 31) interrupt source bit is set to “1” (valid),compare match interrupt request occurs.

Notes on Input Capture

1. If the capture trigger is input while the capture register (low-order and high-order) is in read, captured value is changed between high-order reading and low-order reading. Accordingly, some countermeasure by software is recommended, for example comparing the values that twice of read.

2. Timer A cannot be used for the capture source timer in the following state;
   - X_{IN} oscillation selected by clock division ratio selection bits (bits 7 and 6 of CPU mode register (address $3B_{16}$))
   - Timer A count source: On-chip oscillator output.

   Timer B cannot be used for the capture source timer in the following state;
   - X_{IN} oscillation selected by clock division ratio selection bits
   - Timer B count source: Timer A underflow
   - Timer A count source: On-chip oscillator output.

3. As shown below, when the capture input is performed to both capture latch 00 and 01 at the same time, the value of capture 0 status bit (bit 4 of capture/compare status register (address 22_{16})) is undefined (same as capture 1).
- When “1” is written to capture latch 00 software trigger bit (bit 0 of capture software trigger register (address 13_{16})) and capture latch 01 software trigger bit (bit 1 of capture software trigger register) at the same time
- When external trigger of capture latch 00 and software trigger of capture latch 01 occur at the same time
- When external trigger of capture latch 01 and software trigger of capture latch 00 occur at the same time

4. When the capture interrupt is used as the interrupt for return from stop mode, set the capture 0 noise filter clock selection bits (bits 5 and 4 of capture mode register (address 20_{16})) to “00 (Filter stop)” (same as capture 1).

REVISION HISTORY	7542 Group Datasheet

Rev.	Date	Description	
		Page	Summary
3.00	Jun 01, 2005	1	ROM size of Flash memory version revised.
		2	Fig.1 M37542F8GP → M37542FxGP
			Fig.2 M37542F8FP → M37542FxFP
		3	Fig.3 M37542F8SP → M37542FxSP
		5	Table 1 Performance overview added.
		10	Table 2 Function of Vcc, Vss revised.
		11	Flash memory size revised, and Fig.10 M37542F4 added.
		12	Table 3 M37542F4GP,M37542F4FP,M37542F4SP added.
		21	Fig. 20 (5) Port $P0_5$ revised.
		24	Table 7 Termination of unused pins added.
		46	Description of Serial I/O revised.
		53	[UART2 control register (UART2CON)] revised.
		54	Fig. 64 UART2 contorl register revised.
		60	Description of Clock Generating Circuit revised. Fig. 74 revised.
		63	Fig. 79 revised.
		72	Table 9 Temperature at program/erase added.
		73	Fig.94 16 Kbyte ROM Product added.
		77	Table 11 List of software commands (CPU rewrite mode) revised.
		86	Fig.104 M37542F8GP → M37542FxGP
		87	Fig.105 M37542F8SP → M37542FxSP
			Fig.106 M37542F8FP → M37542FxFP
		91	Fig.109 M37542F8GP → M37542FxGP
		92	Fig.110 M37542F8SP → M37542FxSP
			Fig.111 M37542F8FP → M37542FxFP
		95	M37542F4GP,M37542F4FP,M37542F4SP added.
			Table 15 Conditions: Description added.
		98	Table 18 Note 1 added.
		100	Table 20, Fig. 104 added.
		105	Table 28 Conditions: Description added.
		108	Table 31 Note 1 added.
		110	Table 34, Fig. 106 added.
		114-122	Extended operating temperature 125 °C version added.
		125	(2) How to reference the processor status register revised.
			Fig. 2 revised.
		-	Package revised.
3.01	Nov 02, 2005	-	Bit name revised: STP instruction disable bit → STP instruction function selection bit
		57	- Description for “Operation of STP instruction function selection bit” revised.
			- Notes on Watchdog Timer added.
			- Fig.68: Blodk diagram of watchdog timer revised.
			- Fig.69: Bit 6 and Bit 7 of WDTCON revised.
			Bit 6: Bit name and its description revised. (Bit function is not changed.)
			→ STP instruction function selection bit
			0 : System enters into the stop mode at the STP instruction execution
			1 : Internal reset occurs at the STP instruction execution
		61	-Notes on Clock Generating Circuit : Note on Count source added.
		132	Notes on Watchdog Timer : Note on Count source added.
		133	Notes on Clock Generating Circuit : Note on Count source added.