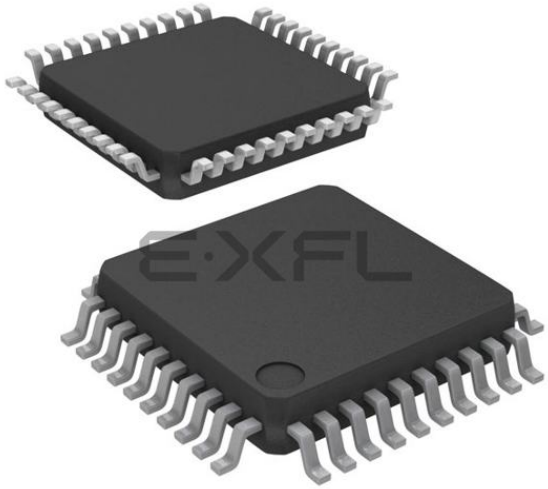
Renesas Electronics America Inc - M37546G4GP#U0 Datasheet



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	LVD, POR, WDT
Number of I/O	25
Program Memory Size	16KB (16K x 8)
Program Memory Type	QzROM
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 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/m37546g4gp-u0

7546 Group

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

REJ03B0160-0122
Rev.1.22
Mar 13, 2009

DESCRIPTION

The 7546 Group is the QzROM version of 7542 Group.
The 7546 Group has the pin-compatibility with the 7542 Group. As new functions, the power-on reset, the low voltage detection circuit, and the function set ROM are added.

FEATURES

- Basic machine-language instructions 71
- The minimum instruction execution time ..... 0.25 μs (at 8 MHz oscillation frequency, double-speed mode for the shortest instruction)
- Memory size
  - ROM ..... 8K, 16K bytes
  - RAM ..... 384, 512 bytes
- Programmable I/O ports 25
- Interrupts ..... 18 sources, 16 vectors
- Timers ..... 8-bit × 2
  ..... 16-bit × 2
- Output compare 4-channel
- Input capture 2-channel
- Serial interface ..... 8-bit × 2 (UART or Clock-synchronized)
- A/D converter ..... 10-bit × 6 channels
- Clock generating circuit ..... Built-in type
  (low-power dissipation by an on-chip oscillator)
  (connected to external ceramic resonator or quartz-crystal oscillator permitting RC oscillation)
- Watchdog timer ..... 16-bit × 1
- Power-on reset circuit ..... Built-in type
- Low voltage detection circuit ..... Built-in type
- Power source voltage
  - XIN oscillation frequency at ceramic oscillation, in double-speed mode
    - At 8 MHz ..... 4.5 to 5.5 V
    - At 6.5 MHz ..... 4.0 to 5.5 V
    - At 2 MHz ..... 2.4 to 5.5 V
    - At 1 MHz ..... 2.2 to 5.5 V
  - XIN oscillation frequency at ceramic oscillation, in high-speed mode or middle-speed mode
    - At 8 MHz ..... 4.0 to 5.5 V
    - At 4 MHz ..... 2.4 to 5.5 V
    - At 2 MHz ..... 2.2 to 5.5 V
  - XIN oscillation frequency at RC oscillation in high-speed mode or middle-speed mode
    - At 4 MHz ..... 4.0 to 5.5 V
    - At 2 MHz ..... 2.4 to 5.5 V
    - At 1 MHz ..... 2.2 to 5.5 V
  - XIN oscillation frequency at on-chip oscillation ..... 1.8 to 5.5 V
- Power dissipation ..... 29.5 mW (Typ.)
- Operating temperature range ..... –20 to 85 °C

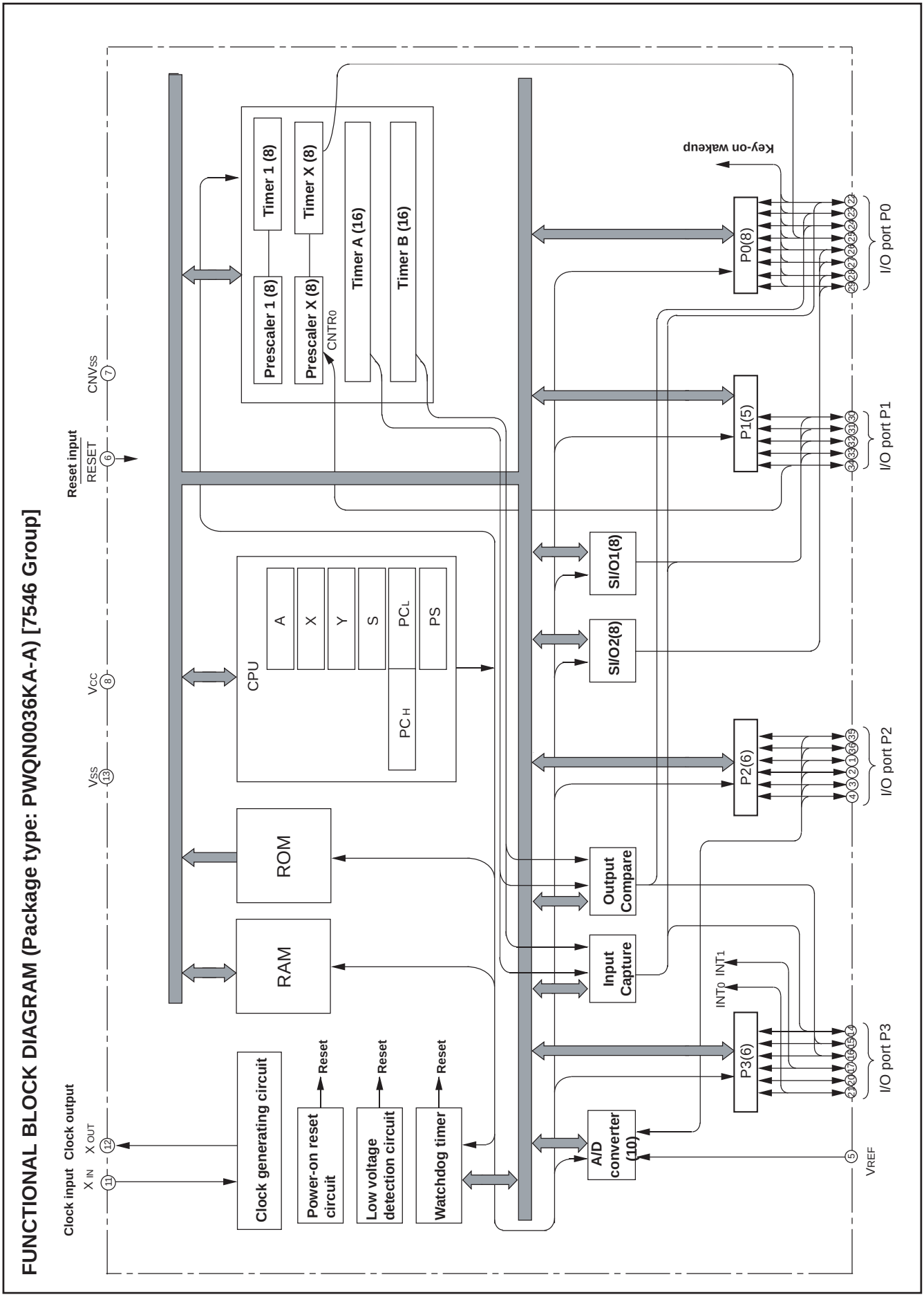


Fig. 6 Functional block diagram (Package type: PWQN0036KA-A)

Interrupts

The 7546 Group interrupts are vector interrupts with a fixed priority scheme, and generated by 16 sources among 18 sources: 6 external, 11 internal, and 1 software.
The interrupt sources, vector addresses(1) , and interrupt priority are shown in Table 8.

Each interrupt except the BRK instruction interrupt has the interrupt request bit and the interrupt enable bit. These bits and the interrupt disable flag (I flag) control the acceptance of interrupt requests. Fig. 20 shows an interrupt control diagram.

An interrupt request is accepted when all of the following conditions are satisfied:

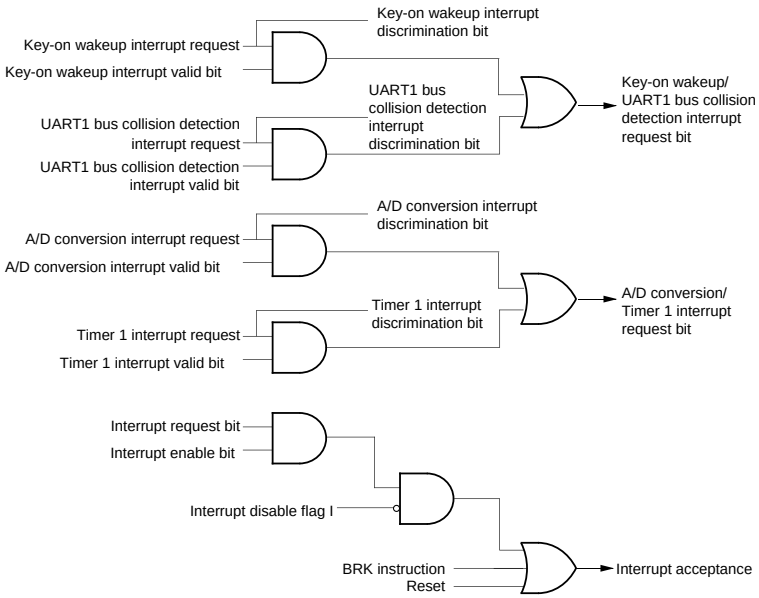
- Interrupt disable flag.....“0”
- Interrupt request bit.....“1”
- Interrupt enable bit.....“1”

Though the interrupt priority is determined by hardware, priority processing can be performed by software using the above bits and flag.

Table 8 Interrupt vector addresses and priority

Interrupt source	Priority	Vector addresses (Note 1)		Interrupt request generating conditions	Remarks
		High-order	Low-order		
Reset (Note 2)	1	FFFD16	FFFC16	At reset input	Non-maskable
Serial I/O1 receive	2	FFFB16	FFFA16	At completion of serial I/O1 data receive	Valid only when serial I/O1 is selected
Serial I/O1 transmit	3	FFF916	FFF816	At completion of serial I/O1 transmit shift or when transmit buffer is empty	Valid only when serial I/O1 is selected
Serial I/O2 receive	4	FFF716	FFF616	At completion of serial I/O2 data receive	Valid only when serial I/O2 is selected
Serial I/O2 transmit	5	FFF516	FFF416	At completion of serial I/O2 transmit shift or when transmit buffer is empty	Valid only when serial I/O2 is selected
INT_0	6	FFF316	FFF216	At detection of either rising or falling edge of INT_0 input	External interrupt (active edge selectable)
INT_1	7	FFF116	FFF016	At detection of either rising or falling edge of INT_1 input	External interrupt (active edge selectable)
Key-on wake-up/ UART1 bus collision detection (Note 3)	8	FFEF16	FFEE16	At falling of conjunction of input logical level for port P0 (at input) At detection of UART1 bus collision detection	External interrupt (valid at falling edge) When UART1 bus collision detection interrupt is enabled.
$CNTR_0$	9	FFED16	FFEC16	At detection of either rising or falling edge of $CNTR_0$ input	External interrupt (active edge selectable)
Capture 0	10	FFEB16	FFEA16	At detection of either rising or falling edge of Capture 0 input	External interrupt (active edge selectable)
Capture 1	11	FFE916	FFE816	At detection of either rising or falling edge of Capture 1 input	External interrupt (active edge selectable)
Compare	12	FFE716	FFE616	At compare matched	Compare interrupt source is selected.
Timer X	13	FFE516	FFE416	At timer X underflow	
Timer A	14	FFE316	FFE216	At timer A underflow	
Timer B	15	FFE116	FFE016	At timer B underflow	
A/D conversion/ Timer 1 (Note 4)	16	FFDF16	FFDE16	At completion of A/D conversion At timer 1 underflow	STP release timer underflow
BRK instruction	17	FFDD16	FFDC16	At BRK instruction execution	Non-maskable software interrupt

Notes1: Vector addresses contain internal jump destination addresses.
2: Reset function in the same way as an interrupt with the highest priority.
3: Key-on wakeup interrupt and UART1 bus collision detection interrupt can be enabled by setting of interrupt source set register. The occurrence of these interrupts are discriminated by interrupt source discrimination register.
4: A/D conversion interrupt and Timer 1 interrupt can be enabled by setting of interrupt source set register. The occurrence of these interrupts are discriminated by interrupt source discrimination register.



Note: For key-on wakeup, UART1 bus collision detection, A/D conversion and Timer 1 interrupt, even if interrupt valid bit ($000A_{16}$) is set "0: Invalid", interrupt discrimination bit ($000B_{16}$) is set to "1: interrupt occurs" when corresponding interrupt request occurs.
But corresponding interrupt request bit ($003C_{16}$, $003D_{16}$) is not set to "1".

Fig. 20 Interrupt control

• Interrupt Disable Flag

The interrupt disable flag is assigned to bit 2 of the processor status register. This flag controls the acceptance of all interrupt requests except for the BRK instruction. When this flag is set to "1", the acceptance of interrupt requests is disabled. When it is set to "0", the acceptance of interrupt requests is enabled. This flag is set to "1" with the SEI instruction and set to "0" with the CLI instruction.

When an interrupt request is accepted, the contents of the processor status register are pushed onto the stack while the interrupt disable flag remains set to "0". Subsequently, this flag is automatically set to "1" and multiple interrupts are disabled.

To use multiple interrupts, set this flag to "0" with the CLI instruction within the interrupt processing routine.

The contents of the processor status register are popped off the stack with the RTI instruction.

• Interrupt Request Bits

Once an interrupt request is generated, the corresponding interrupt request bit is set to "1" and remains "1" until the request is accepted. When the request is accepted, this bit is automatically set to "0".

Each interrupt request bit can be set to "0", but cannot be set to "1", by software.

• Interrupt Enable Bits

The interrupt enable bits control the acceptance of the corresponding interrupt requests. When an interrupt enable bit is set to "0", the acceptance of the corresponding interrupt request is disabled. If an interrupt request occurs in this condition, the corresponding interrupt request bit is set to "1", but the interrupt request is not accepted. When an interrupt enable bit is set to "1", the acceptance of the corresponding interrupt request is enabled.

Each interrupt enable bit can be set to "0" or "1" by software.

The interrupt enable bit for an unused interrupt should be set to "0".

• Interrupt Enable Setting

The following interrupt sources can be set to valid or invalid by the interrupt source set register ($000A_{16}$).

- Key-on wakeup
- UART1 bus collision detection interrupt
- A/D conversion
- Timer 1 interrupt

• Interrupt edge selection

The valid edge of external interrupt INT_0 and INT_1 can be selected by the interrupt edge selection bit of the interrupt edge selection register ($003A_{16}$), respectively.

Set bit 2 of interrupt edge selection register to "1".

• Key-on wakeup

Enable/disable of a key-on wakeup of pins $P0_0$, $P0_4$, and $P0_6$ can be selected by the key-on wakeup enable bit of the interrupt edge selection register ($003A_{16}$), respectively.

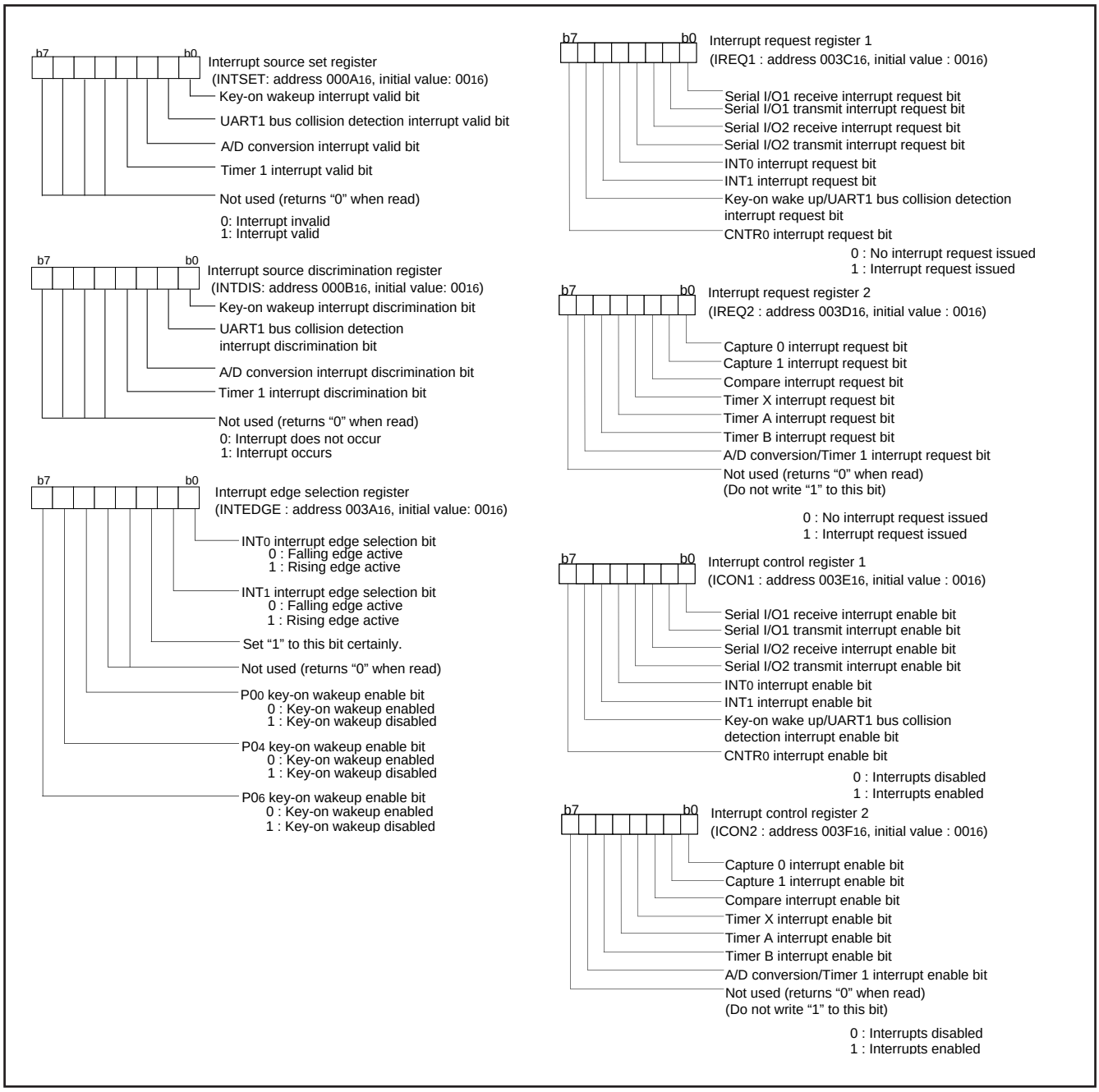


Fig. 21 Structure of Interrupt-related registers

(4) Pulse width measurement mode

In the pulse width measurement mode, the pulse width of the signal input to P14/CNTR0 pin is measured.

The operation of Timer X can be controlled by the level of the signal input from the CNTR0 pin.

When the CNTR0 active edge switch bit is “0”, the signal selected by the timer X count source selection bit is counted while the input signal level of CNTR0 pin is “H”. The count is stopped while the pin is “L”. Also, when the CNTR0 active edge switch bit is “1”, the signal selected by the timer X count source selection bit is counted while the input signal level of CNTR0 pin is “L”. The count is stopped while the pin is “H”.

Timer X can stop counting by setting “1” to the timer X count stop bit in any mode.

Also, when Timer X underflows, the timer X interrupt request bit is set to “1”.

Note on Timer X is described below;

■ Note on Timer X

(1) CNTR0 interrupt active edge selection-1

CNTR0 interrupt active edge depends on the CNTR0 active edge switch bit.

When this bit is “0”, the CNTR0 interrupt request bit is set to “1” at the falling edge of CNTR0 pin input signal. When this bit is “1”, the CNTR0 interrupt request bit is set to “1” at the rising edge of CNTR0 pin input signal.

(2) CNTR0 interrupt active edge selection-2

According to the setting value of CNTR0 active edge switch bit, the interrupt request bit may be set to “1”.

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 active edge switch 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).

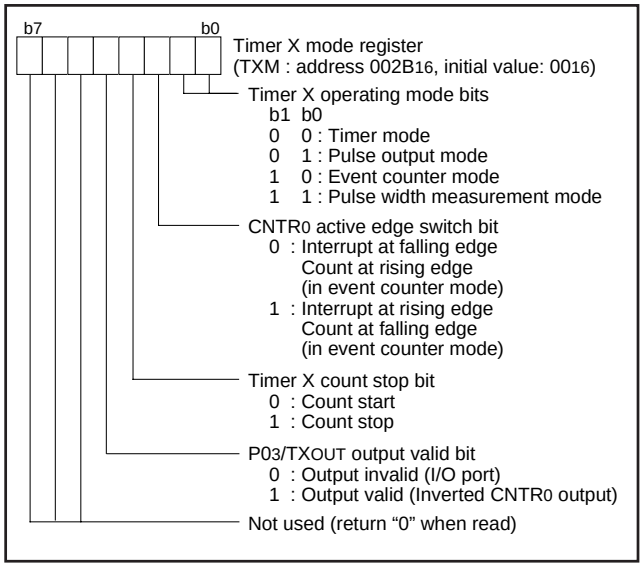
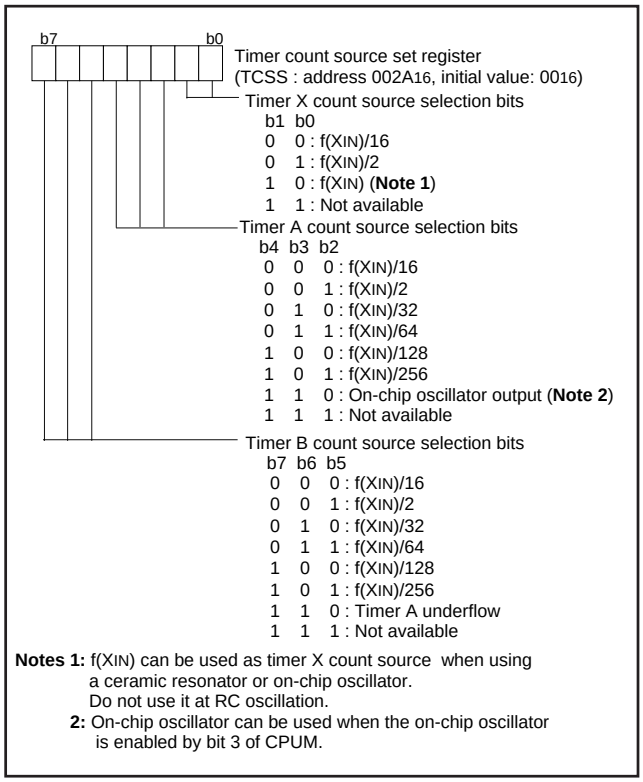


Fig. 26 Structure of timer X mode register



Notes 1: f(XIN) can be used as timer X count source when using a ceramic resonator or on-chip oscillator.
Do not use it at RC oscillation.
2: On-chip oscillator can be used when the on-chip oscillator is enabled by bit 3 of CPUM.

Fig. 27 Timer count source set register

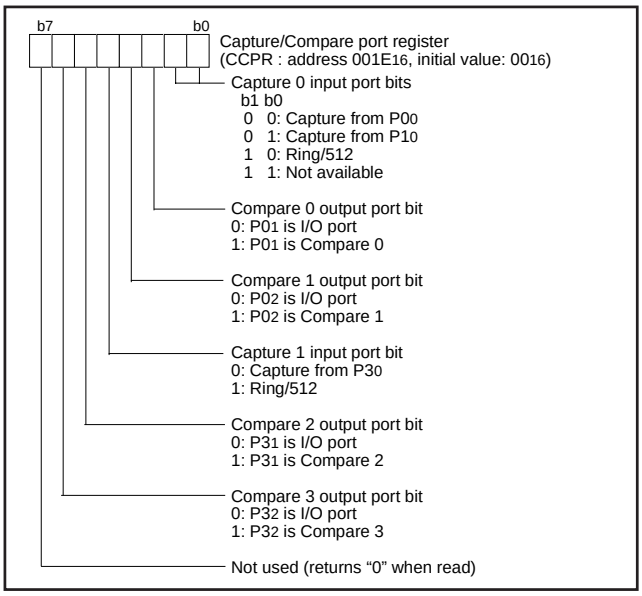


Fig. 34 Structure of capture/compare port register

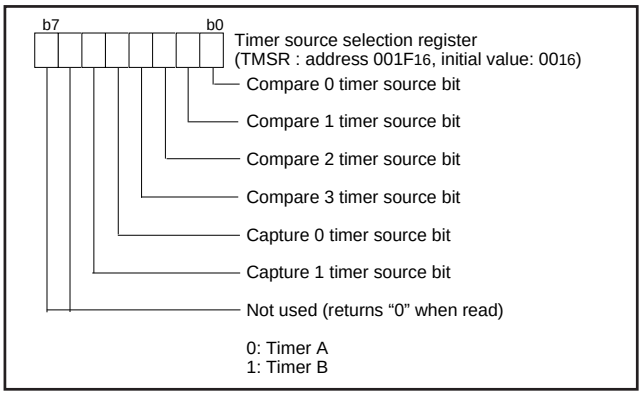


Fig. 35 Structure of timer source selection register

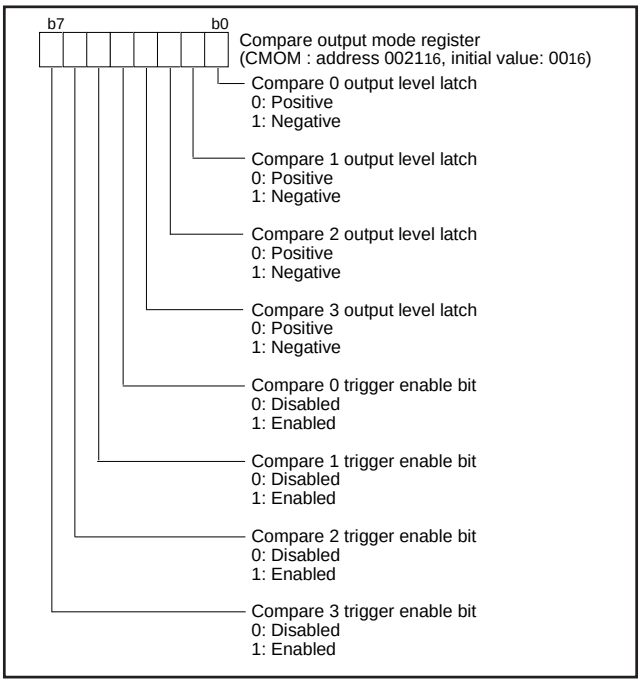


Fig. 36 Structure of compare output mode register

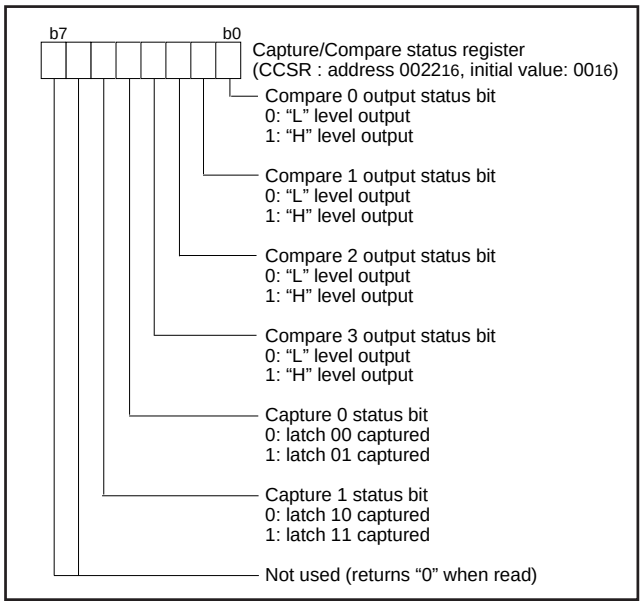


Fig. 37 Structure of capture/compare status register

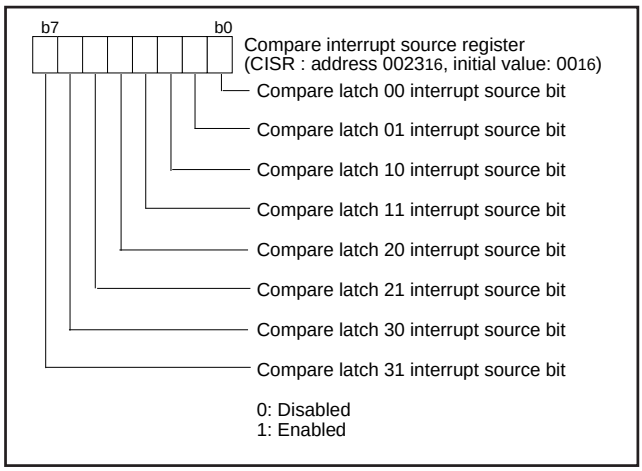


Fig. 38 Structure of compare interrupt source register

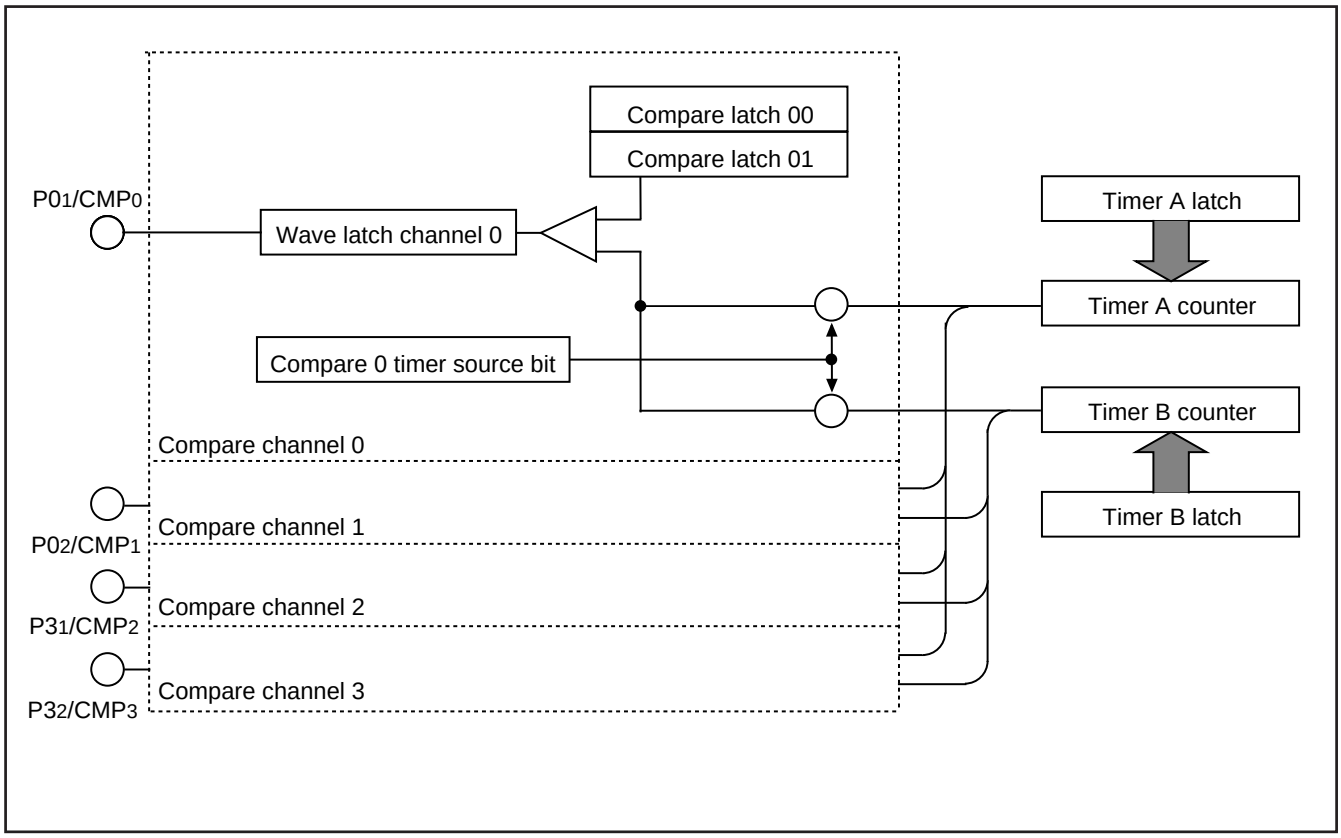


Fig. 39 Block diagram of output compare

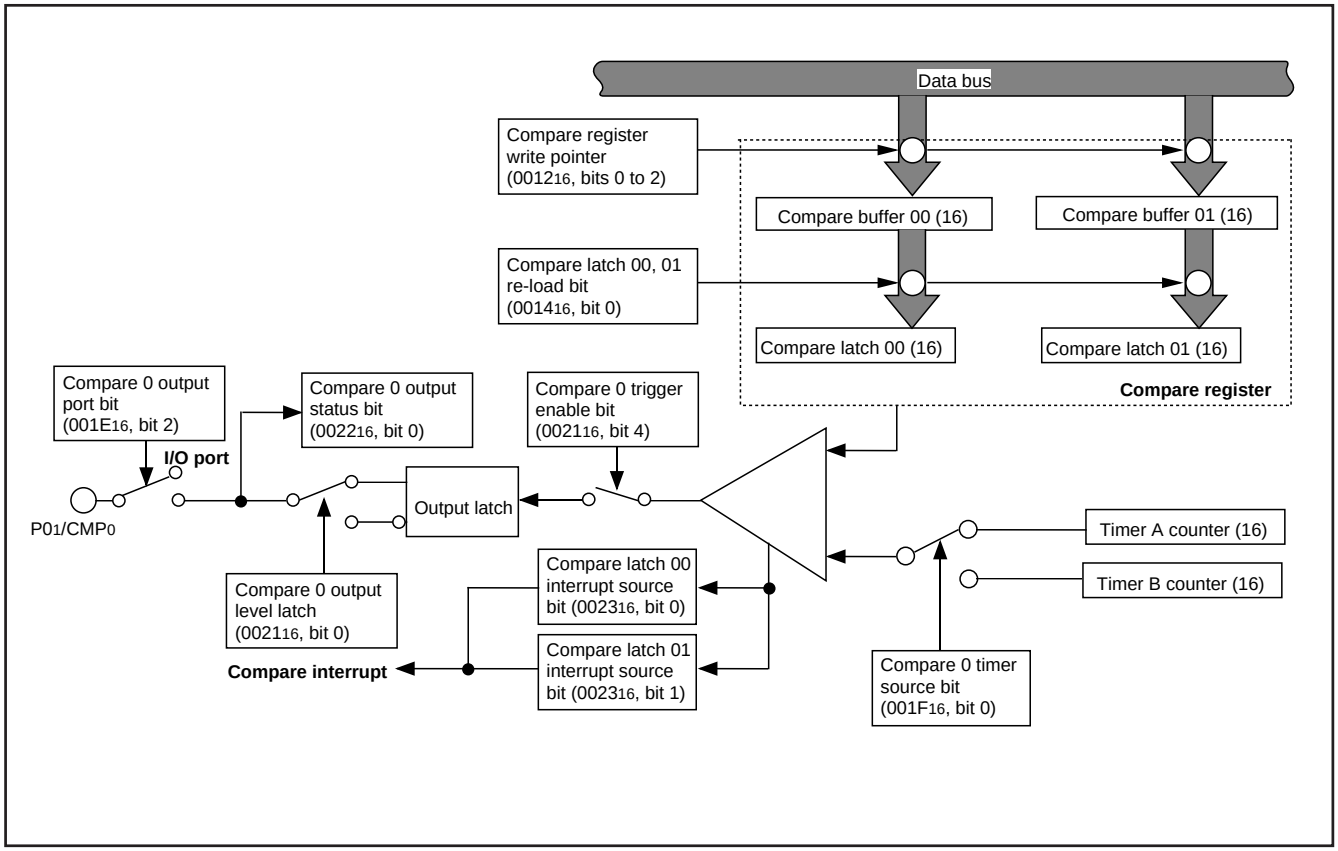


Fig. 40 Block diagram of compare channel 0

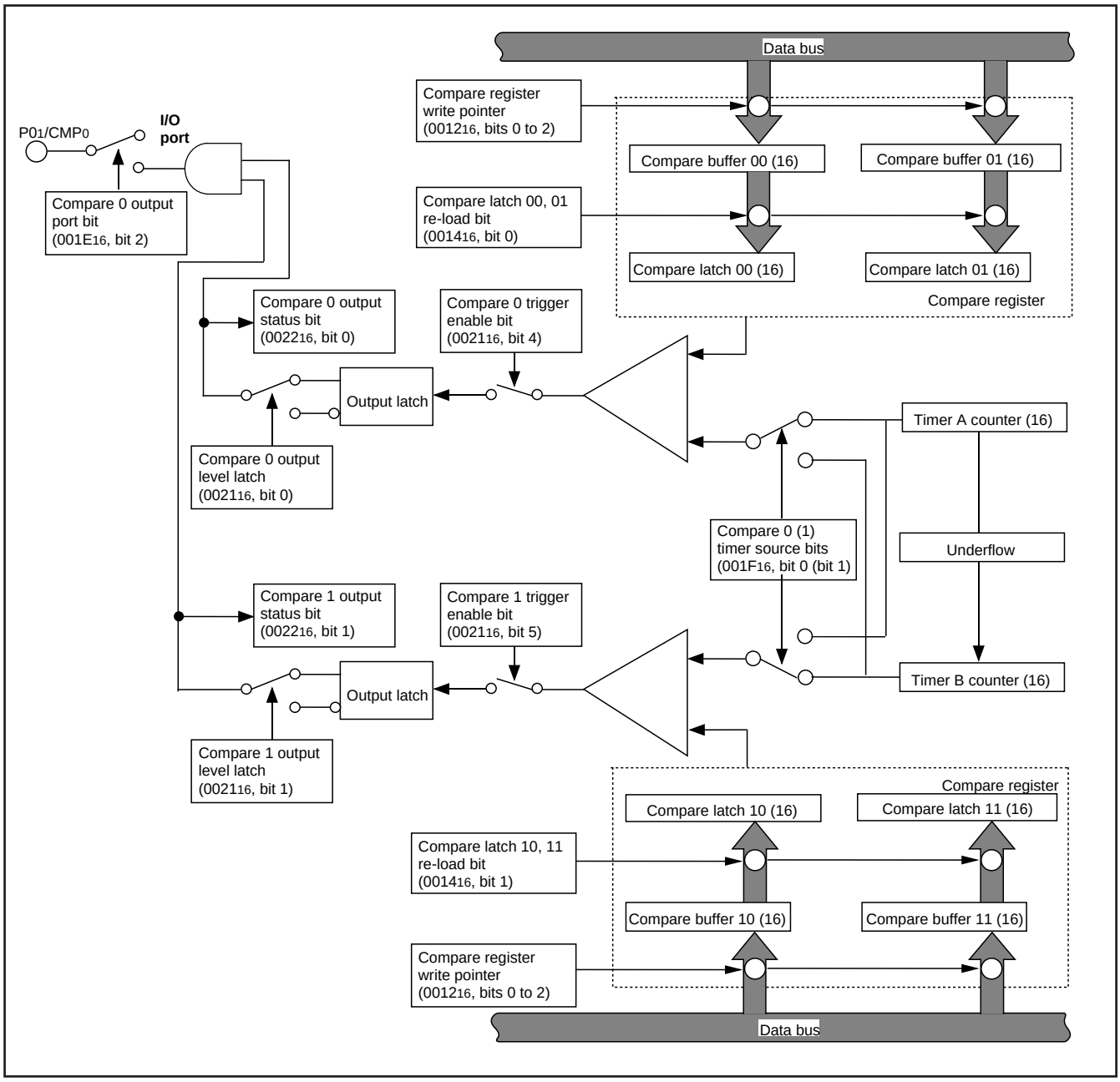
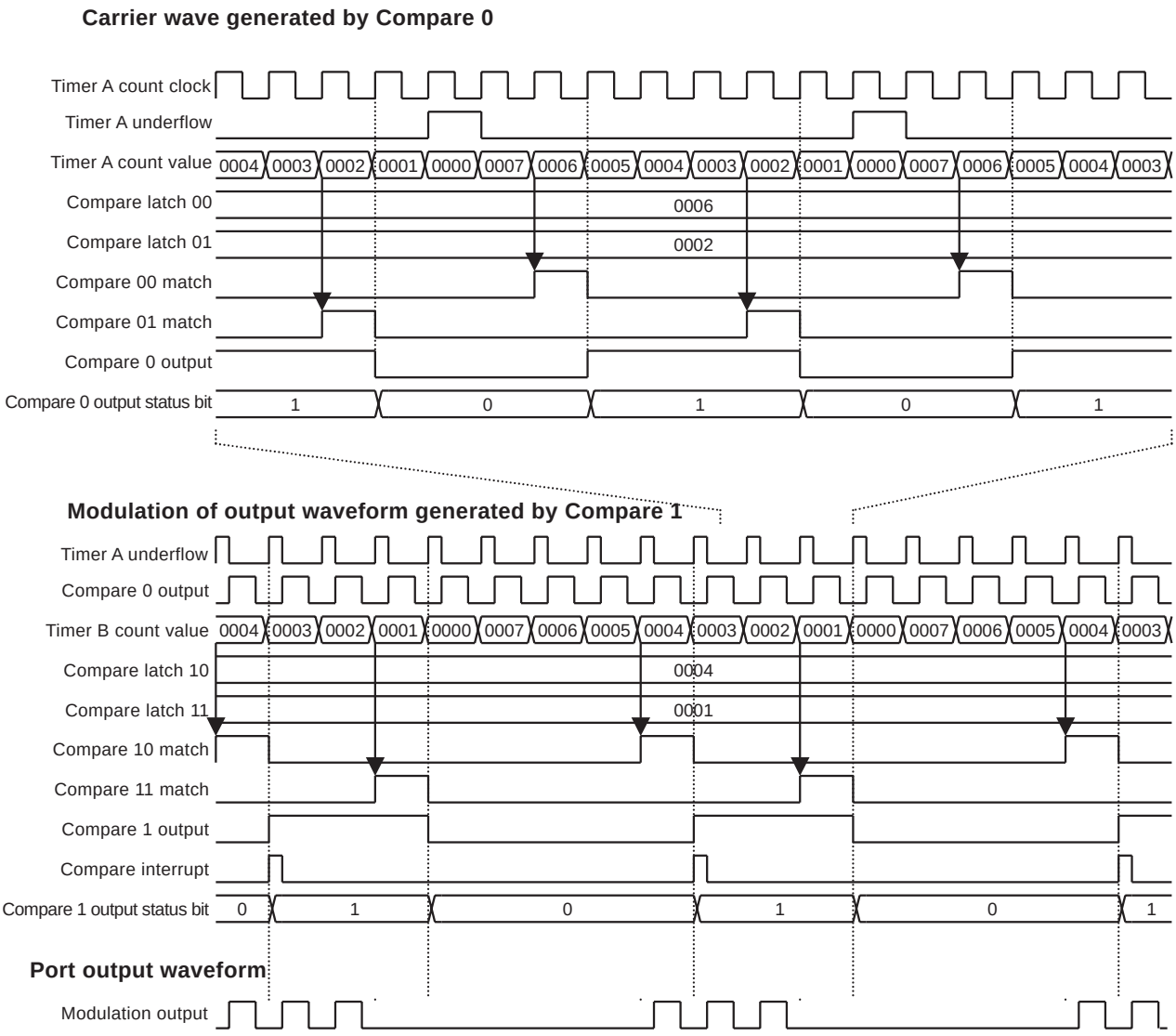


Fig. 41 Block diagram at modulation mode



Note: Compare interrupt occurs only for the interrupt source selected by Compare interrupt source register.

Fig. 44 Output compare mode (compare 0, 1 modulation mode)

●Serial I/O2

Serial I/O2 can be used as either clock synchronous or asynchronous (UART) serial I/O. A dedicated timer is also provided for baud rate generation.

(1) Clock Synchronous Serial I/O2 Mode

Clock synchronous serial I/O2 mode can be selected by setting the serial I/O2 mode selection bit of the serial I/O2 control register (bit 6) to "1".

For clock synchronous serial I/O2, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the TB/RB.

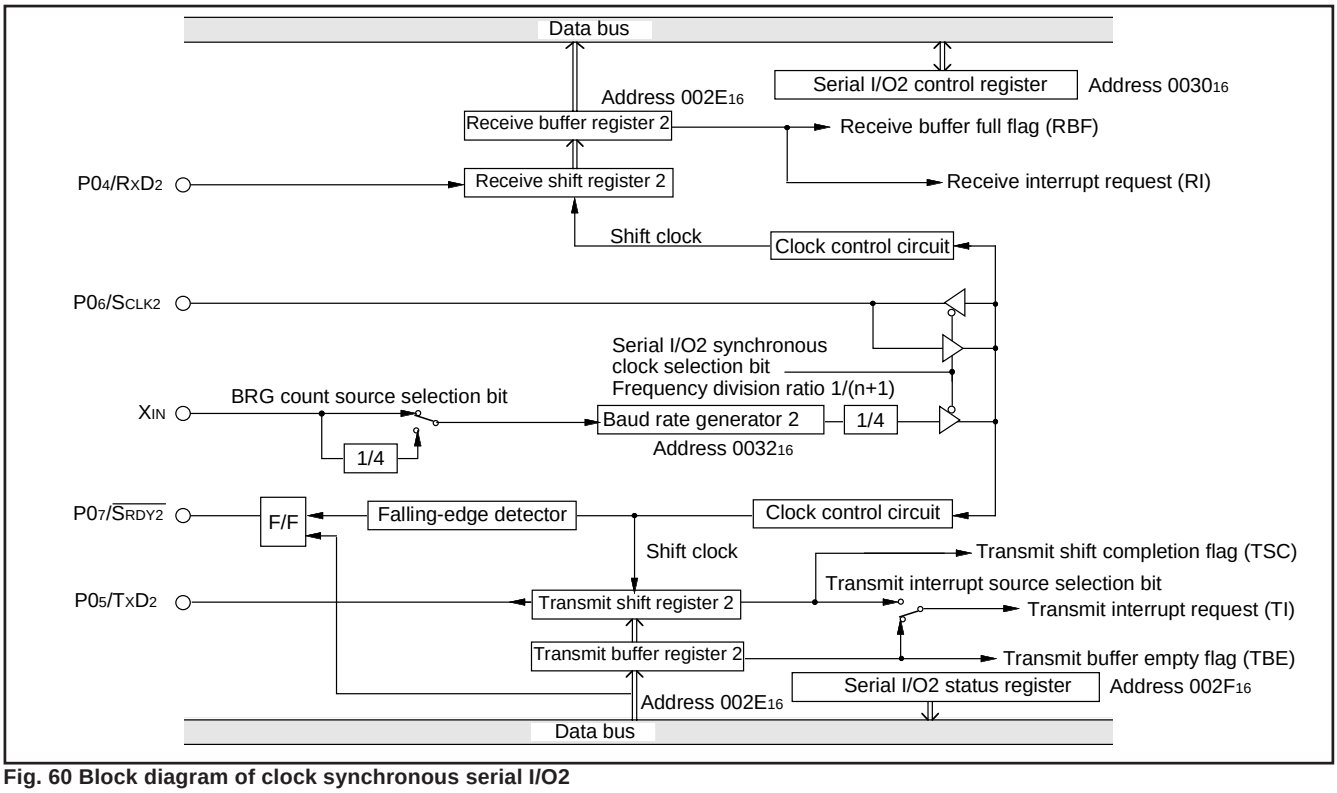
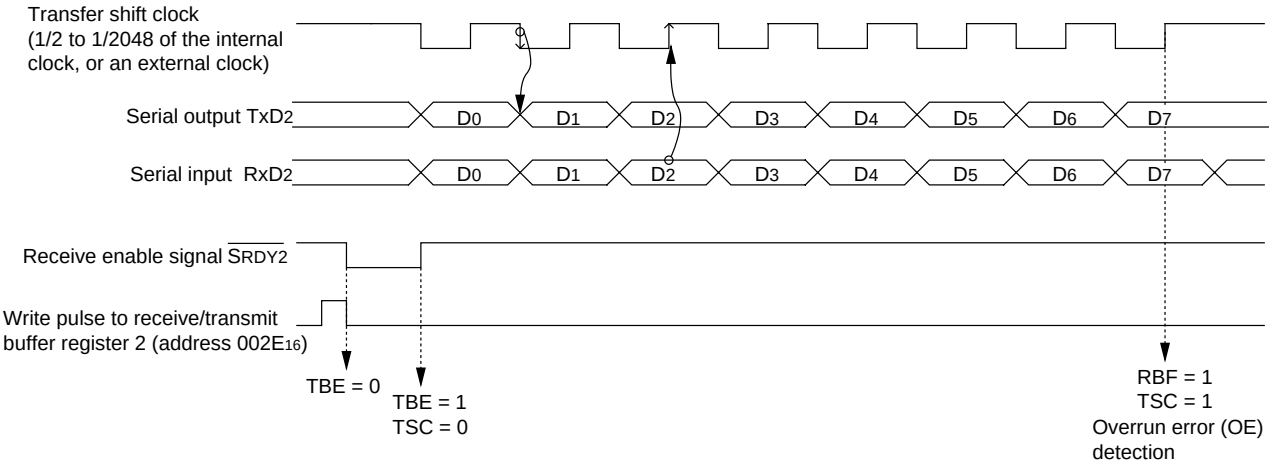


Fig. 60 Block diagram of clock synchronous serial I/O2



Notes 1: As the transmit interrupt (TI), which can be selected, either when the transmit buffer has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O2 control register.

2: If data is written to the transmit buffer register when TSC=0, the transmit clock is generated continuously and serial data is output continuously from the TxD_2 pin.

3: The receive interrupt (RI) is set when the receive buffer full flag (RBF) becomes "1" .

Fig. 61 Operation of clock synchronous serial I/O2 function

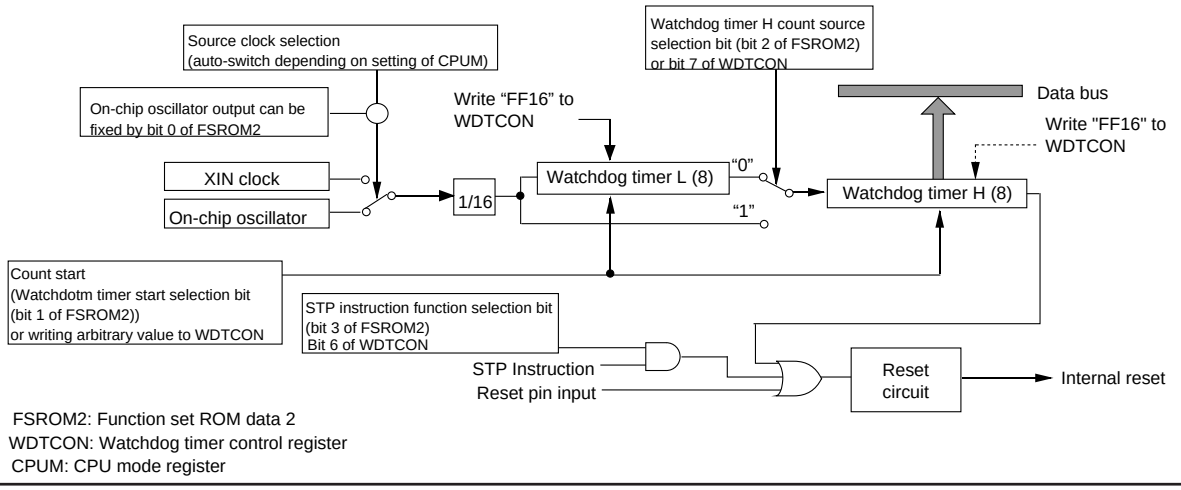
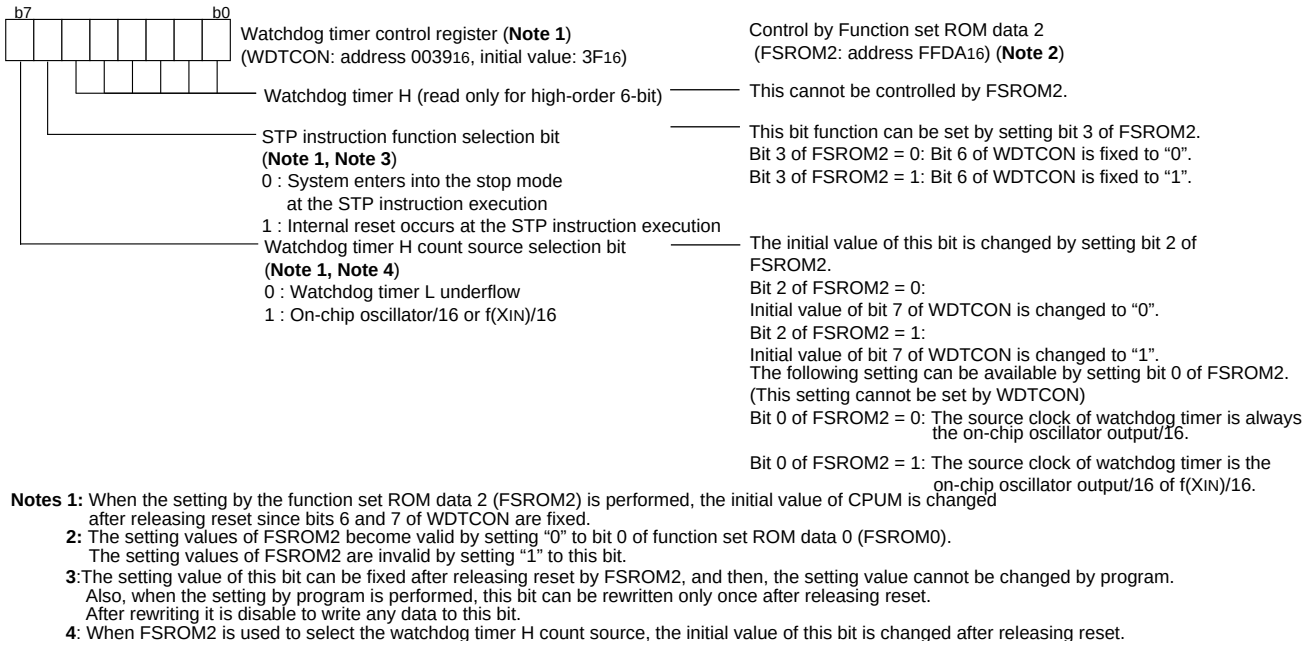


Fig. 68 Block diagram of watchdog timer



Notes 1: When the setting by the function set ROM data 2 (FSROM2) is performed, the initial value of CPUM is changed after releasing reset since bits 6 and 7 of WDTCON are fixed.

2: The setting values of FSROM2 become valid by setting “0” to bit 0 of function set ROM data 0 (FSROM0). The setting values of FSROM2 are invalid by setting “1” to this bit.

3:The setting value of this bit can be fixed after releasing reset by FSROM2, and then, the setting value cannot be changed by program. Also, when the setting by program is performed, this bit can be rewritten only once after releasing reset. After rewriting it is disable to write any data to this bit.

4: When FSROM2 is used to select the watchdog timer H count source, the initial value of this bit is changed after releasing reset.

Fig. 69 Structure of watchdog timer control register

Clock Generating Circuit

An oscillation circuit can be formed by connecting a resonator between XIN and XOUT, and an RC oscillation circuit can be formed by connecting a resistor and a capacitor.
Use the circuit constants in accordance with the resonator manufacturer's recommended values.
No external resistor is needed between XIN and XOUT since a feed-back resistor exists on-chip. (An external feed-back resistor may be needed depending on conditions.)

(1) On-chip oscillator operation

When the MCU operates by the on-chip oscillator for the main clock, connect XIN pin to VCC through a resistor and leave XOUT pin open.
The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.
Be careful that variable frequencies when designing application products.

(2) Ceramic resonator

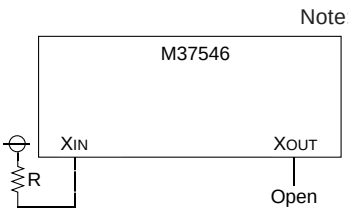
When the ceramic resonator is used for the main clock, connect the ceramic resonator and the external circuit to pins XIN and XOUT at the shortest distance. A feedback resistor is built in between pins XIN and XOUT.

(3) RC oscillation

When the RC oscillation is used for the main clock, connect the XIN pin and XOUT pin to the external circuit of resistor R and the capacitor C at the shortest distance.
The frequency is affected by a capacitor, a resistor and a microcomputer.
So, set the constants within the range of the frequency limits.

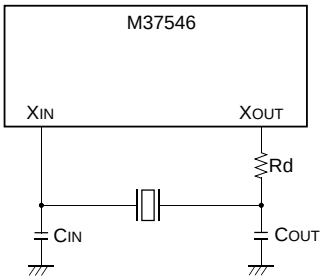
(4) External clock

When the external signal clock is used for the main clock, connect the XIN pin to the clock source and leave XOUT pin open.
Select “0” (ceramic oscillation) to oscillation mode selection bit of CPU mode register ($003B_{16}$).



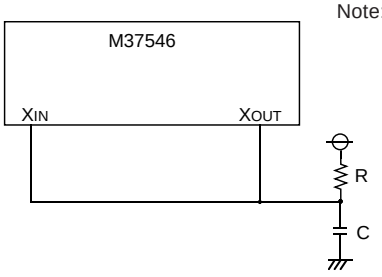
Note: The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.
Be careful that variable frequencies and obtain the sufficient margin.

Fig. 74 Processing of XIN and XOUT pins at on-chip oscillator operation



Note: Insert a damping resistor if required.
The resistance will vary depending on the oscillator and the oscillation drive capacity setting.
Use the value recommended by the maker of the oscillator.
Also, if the oscillator manufacturer’s data sheet specifies that a feedback resistor be added external to the chip though a feedback resistor exists on-chip, insert a feedback resistor between XIN and XOUT following the instruction.

Fig. 75 External circuit of ceramic resonator



Note: Connect the external circuit of resistor R and the capacitor C at the shortest distance.
The frequency is affected by a capacitor, a resistor and a microcomputer.
So, set the constants within the range of the frequency limits.

Fig. 76 External circuit of RC oscillation

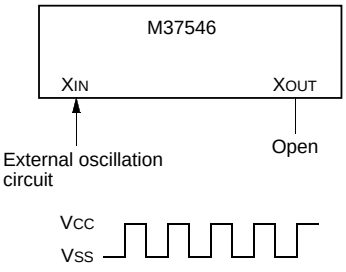


Fig. 77 External clock input circuit

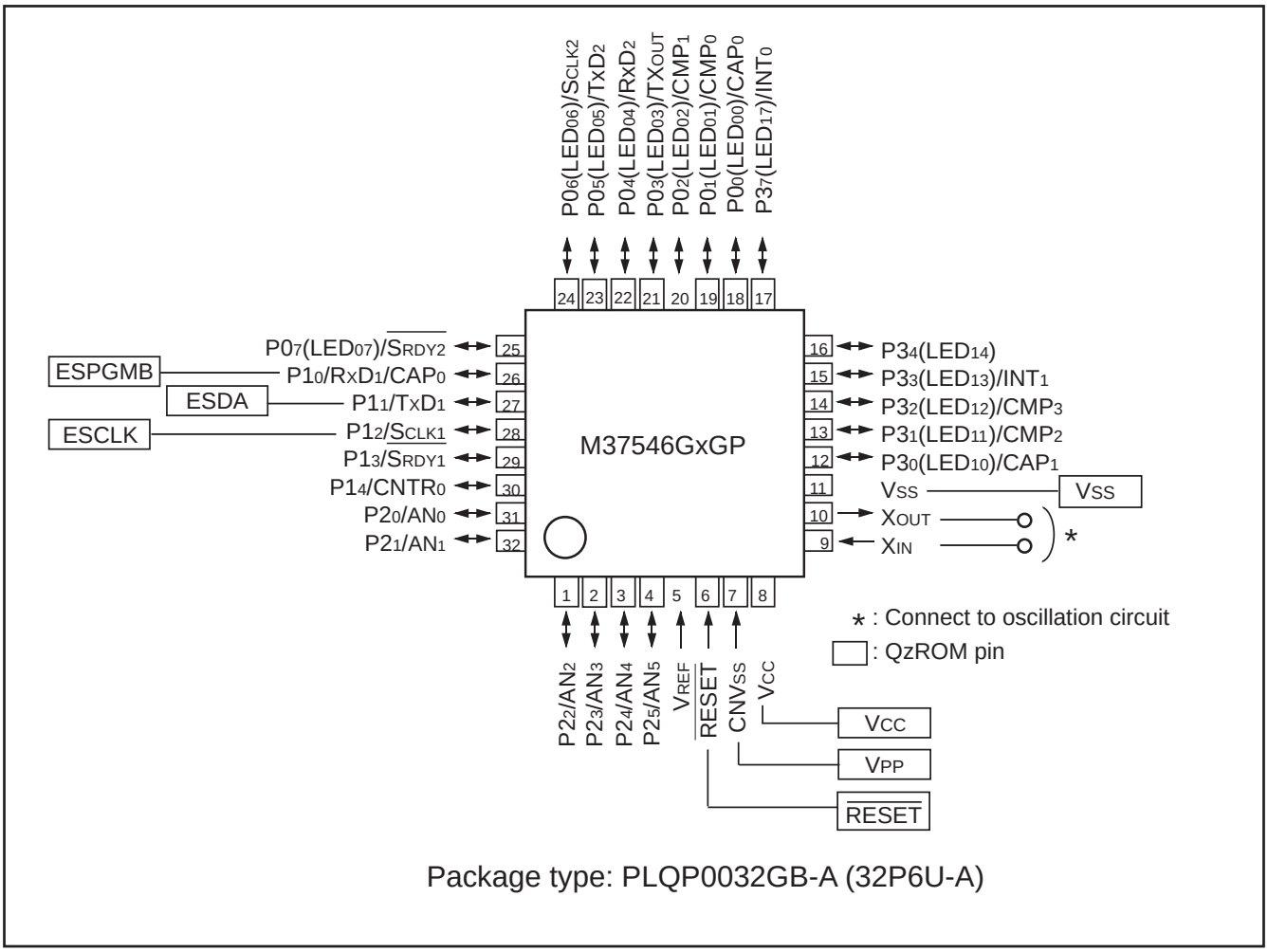


Fig. 89 Pin connection diagram (M37546GxGP)

NOTES ON USE

Countermeasures against noise

1. Shortest wiring length

(1) Package

Select the smallest possible package to make the total wiring length short.

<Reason>

The wiring length depends on a microcomputer package. Use of a small package, for example QFP and not DIP, makes the total wiring length short to reduce influence of noise.

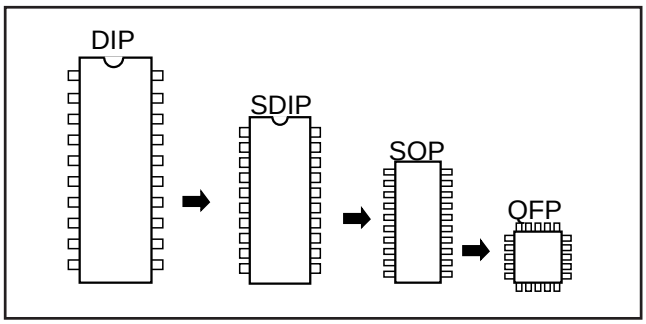


Fig. 94 Selection of packages

(2) Wiring for $\overline{\text{RESET}}$ pin

Make the length of wiring which is connected to the $\overline{\text{RESET}}$ pin as short as possible. Especially, connect a capacitor across the $\overline{\text{RESET}}$ pin and the Vss pin with the shortest possible wiring (within 20 mm).

<Reason>

The width of a pulse input into the $\overline{\text{RESET}}$ pin is determined by the timing necessary conditions. If noise having a shorter pulse width than the standard is input to the $\overline{\text{RESET}}$ pin, the reset is released before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

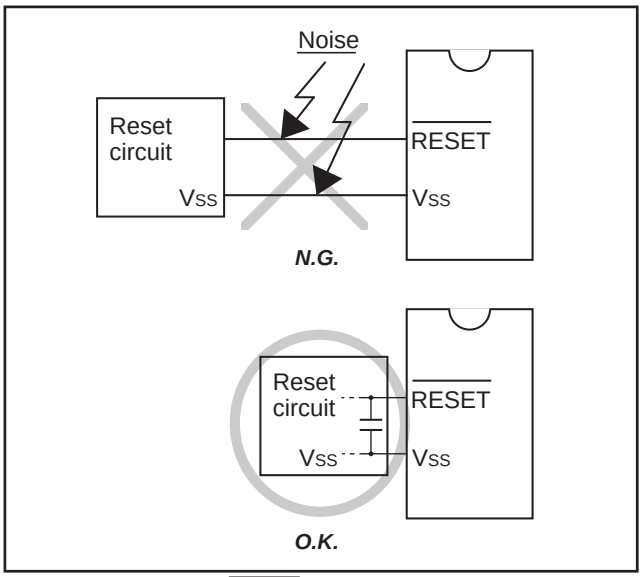


Fig. 95 Wiring for the $\overline{\text{RESET}}$ pin

(3) Wiring for clock input/output pins

- Make the length of wiring which is connected to clock I/O pins as short as possible.
- Make the length of wiring (within 20 mm) across the grounding lead of a capacitor which is connected to an oscillator and the Vss pin of a microcomputer as short as possible.
- Separate the Vss pattern only for oscillation from other Vss patterns.

<Reason>

If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a program failure or program runaway. Also, if a potential difference is caused by the noise between the Vss level of a microcomputer and the Vss level of an oscillator, the correct clock will not be input in the microcomputer.

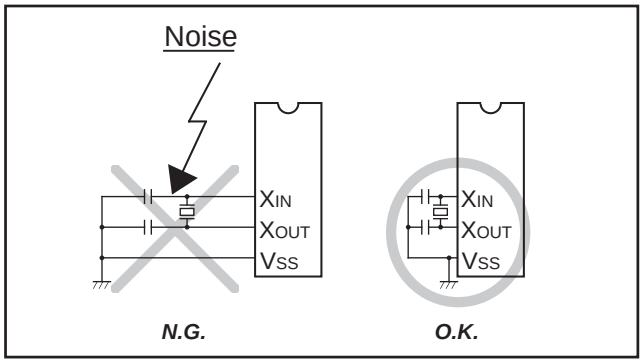


Fig. 96 Wiring for clock I/O pins

(4) Wiring to CNVss pin

Connect CNVss pin to a GND pattern at the shortest distance. The GND pattern is required to be as close as possible to the GND supplied to Vss.

In order to improve the noise reduction, to connect a 5 kΩ resistor serially to the CNVss pin - GND line may be valid.

As well as the above-mentioned, in this case, connect to a GND pattern at the shortest distance. The GND pattern is required to be as close as possible to the GND supplied to Vss.

<Reason>

The CNVss pin of the QzROM is the power source input pin for the built-in QzROM. When programming in the built-in QzROM, the impedance of the CNVss pin is low to allow the electric current for writing flow into the QzROM. Because of this, noise can enter easily. If noise enters the CNVss pin, abnormal instruction codes or data are read from the built-in QzROM, which may cause a program runaway.

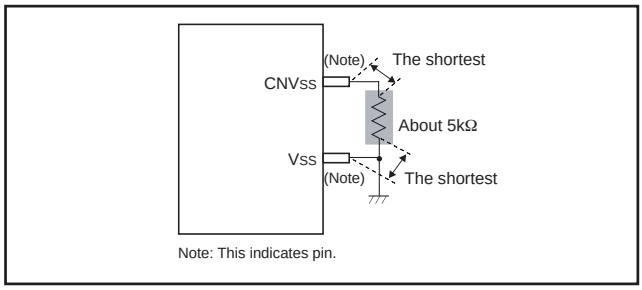


Fig. 97 Wiring for the CNVss pin of the QzPROM

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.

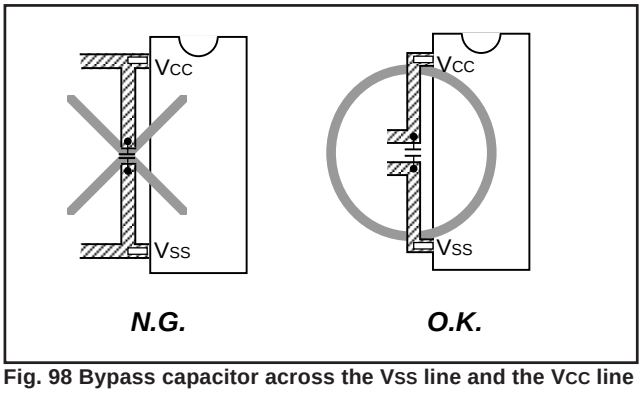


Fig. 98 Bypass capacitor across the Vss line and the Vcc line

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.

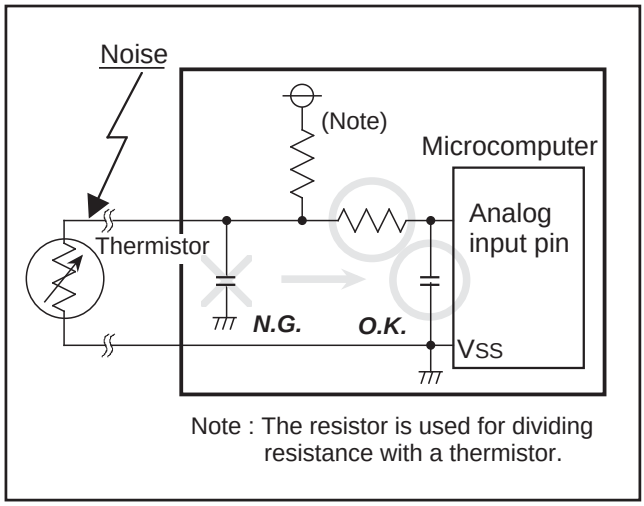


Fig. 99 Analog signal line and a resistor and a capacitor

- 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.

4. Oscillator concerns

Take care to prevent an oscillator that generates clocks for a microcomputer operation from being affected by other signals.

(1) Keeping oscillator away from large current signal lines

Install a microcomputer (and especially an oscillator) as far as possible from signal lines where a current larger than the tolerance of current value flows.

<Reason>

In the system using a microcomputer, there are signal lines for controlling motors, LEDs, and thermal heads or others. When a large current flows through those signal lines, strong noise occurs because of mutual inductance.

(2) Installing oscillator away from signal lines where potential levels change frequently

Install an oscillator and a connecting pattern of an oscillator away from signal lines where potential levels change frequently. Also, do not cross such signal lines over the clock lines or the signal lines which are sensitive to noise.

<Reason>

Signal lines where potential levels change frequently (such as the CNTR pin signal line) may affect other lines at signal rising edge or falling edge. If such lines cross over a clock line, clock waveforms may be deformed, which causes a microcomputer failure or a program runaway.

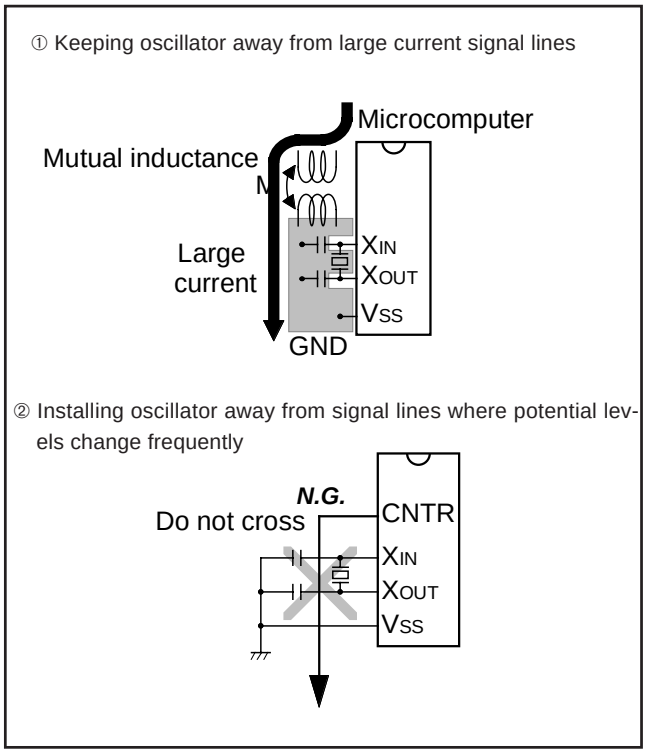


Fig. 100 Wiring for a large current signal line/Writing of signal lines where potential levels change frequently

(3) Oscillator protection using Vss pattern

As for a two-sided printed circuit board, print a Vss pattern on the underside (soldering side) of the position (on the component side) where an oscillator is mounted.

Connect the Vss pattern to the microcomputer Vss pin with the shortest possible wiring. Besides, separate this Vss pattern from other Vss patterns.

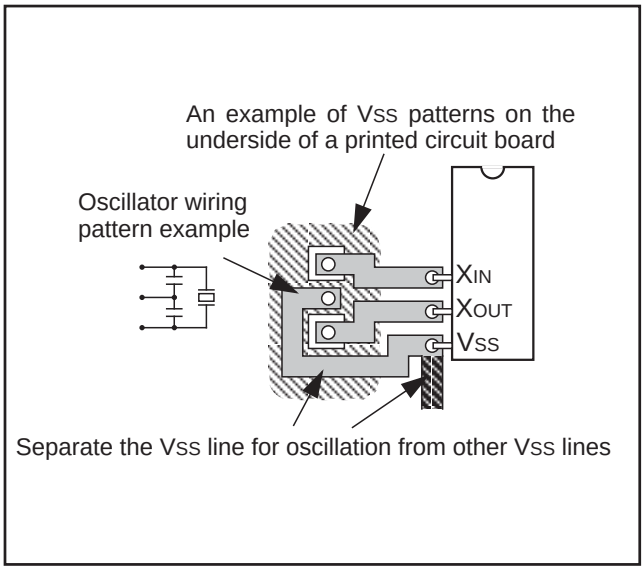


Fig. 101 Vss pattern on the underside of an oscillator

Electrical characteristics (2)

(V_{CC} = 1.8 to 5.5V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
I_{CC}	Power source current *LVD is valid (except at STP)	$f(X_{IN})$ = 8 MHz Output transistors “off”	Double-speed mode		5.9	9.5	mA
			High-speed mode		3.9	7.0	mA
			Middle-speed mode		2.4	5.5	mA
		$f(X_{IN})$ = 2 MHz, V_{CC} = 2.2 V Output transistors “off”	High-speed mode		0.45	1.25	mA
		On-chip oscillator operation mode, Output transistors “off”	Frequency/1		1.55	3.3	mA
			Frequency/2		0.95	2.3	mA
			Frequency/8		0.4	1.1	mA
			Frequency/128		0.25	0.7	mA
		$f(X_{IN})$ = 8 MHz (in WIT state), functions except timer 1 disabled, Output transistors “off”			2.0	3.5	mA
		$f(X_{IN})$ = 2 MHz, V_{CC} = 2.2 V (in WIT state), functions except timer 1 disabled, Output transistors “off”			0.25		mA
		On-chip oscillator operation mode, (in WIT state), functions except timer 1 disabled, Output transistors “off”			0.25	0.7	mA
		Increment when A/D conversion is executed $f(X_{IN})$ = 8 MHz, V_{CC} = 5 V			0.5		mA
		All oscillation stopped (in STP state) Output transistors “off”	Ta = 25 °C		0.1	1.0	μA
			Ta = 85 °C			10	μA
		Low voltage detection circuit self consumption current	Ta = 25 °C V_{CC} = 5 V		70		μA

Note: Increment when A/D conversion is executed includes the reference power source input current (I_{VREF}).

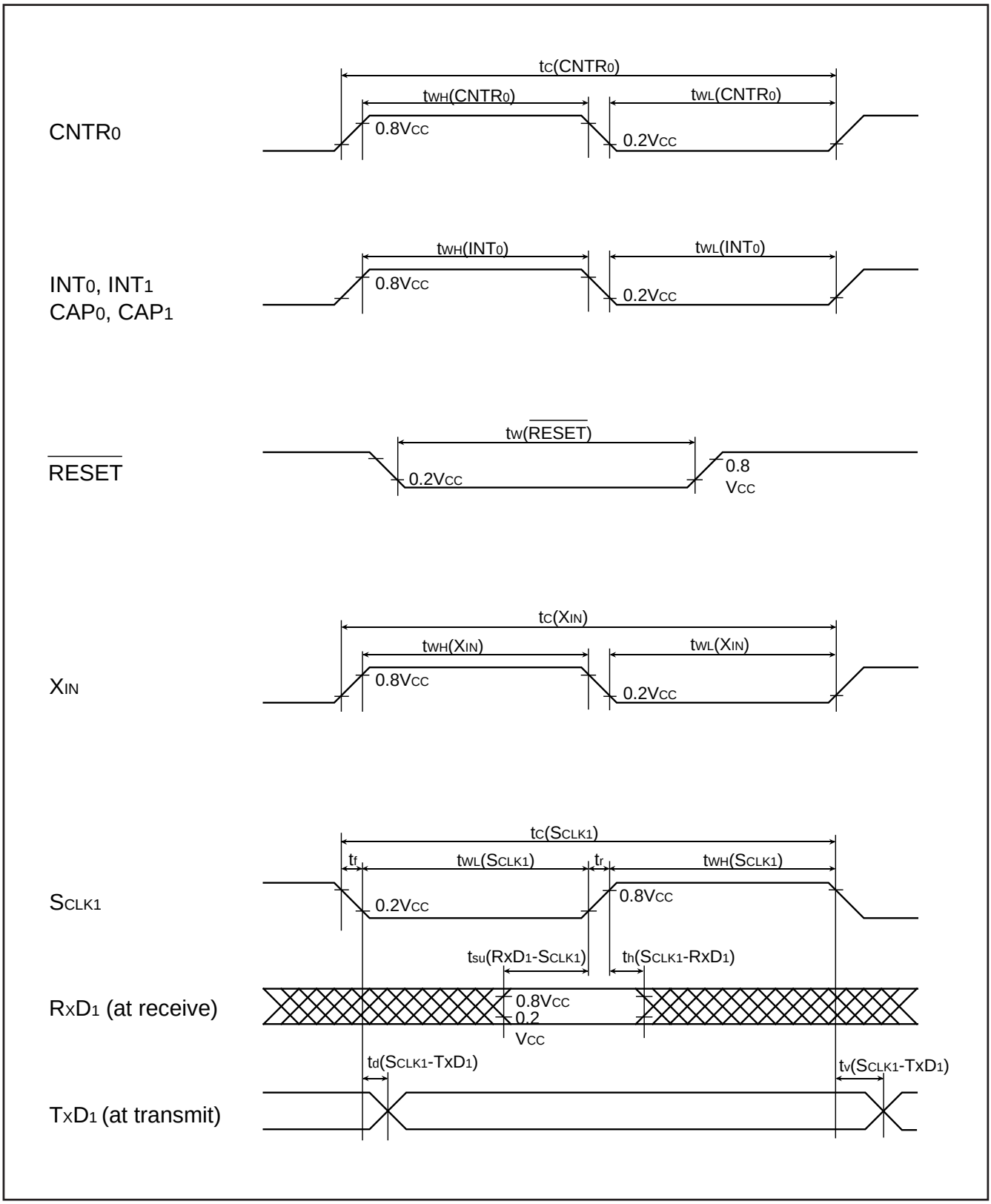


Fig. 107 Timing chart

Termination of Unused Pins

1. Terminate unused pins

Perform the following wiring at the shortest possible distance (20 mm or less) from microcomputer pins.

(1) I/O ports

Set the I/O ports for the input mode and connect each pin to V_{CC} or V_{SS} through each resistor of 1 kΩ to 10 kΩ. The port which can select a built-in pull-up resistor can also use the built-in pull-up resistor.

When using the I/O ports as the output mode, open them at "L" or "H".

- When opening them in the output mode, the input mode of the initial status remains until the mode of the ports is switched over to the output mode by the program after reset. Thus, the potential at these pins is undefined and the power source current may increase in the input mode. With regard to an effects on the system, thoroughly perform system evaluation on the user side.
- Since the direction register setup may be changed because of a program runaway or noise, set direction registers by program periodically to increase the reliability of program.

2. Termination remarks

(1) I/O ports setting as input mode

[1] Do not open in the input mode.

<Reason>

- The power source current may increase depending on the first-stage circuit.
- An effect due to noise may be easily produced as compared with proper termination (1) shown on the above "1. Terminate unused pins".

[2] Do not connect to V_{CC} or V_{SS} directly.

<Reason>

If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur.

[3] Do not connect multiple ports in a lump to V_{CC} or V_{SS} through a resistor.

<Reason>

If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.

Notes on Interrupts

1. Change of relevant register settings

When not requiring for the interrupt occurrence synchronous with the following case, take the sequence shown in Fig. 4.

- When switching external interrupt active edge
- When switching interrupt sources of an interrupt vector address where two or more interrupt sources are allocated

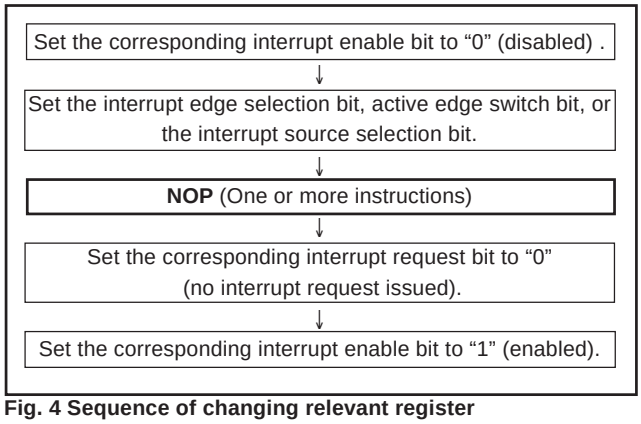


Fig. 4 Sequence of changing relevant register

<Reason>

When setting the followings, the interrupt request bit of the corresponding interrupt may be set to "1".

- When switching external interrupt active edge
  - INT_0 interrupt edge selection bit
    (bit 0 of Interrupt edge selection register (address $003A_{16}$))
  - INT_1 interrupt edge selection bit
    (bit 1 of Interrupt edge selection register)
  - $CNTR_0$ active edge switch bit
    (bit 2 of timer X mode register (address $002B_{16}$))
  - Capture 0 interrupt edge selection bit
    (bits 1 and 0 of capture mode register (address 0020_{16}))
  - Capture 1 interrupt edge selection bit
    (bits 3 and 2 of capture mode register)

2. Check of interrupt request bit

When executing the **BBC** or **BBS** instruction to determine an interrupt request bit immediately after this bit is set to "0", take the following sequence.

<Reason>

If the BBC or BBS instruction is executed immediately after an interrupt request bit is cleared to "0", the value of the interrupt request bit before being cleared to "0" is read.

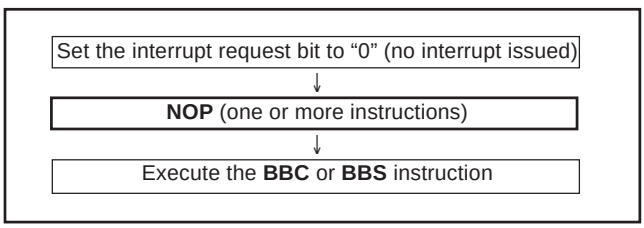


Fig. 5 Sequence of check of interrupt request bit