To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

3804 Group (Spec.L)

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

REJ03B0266-0100
Rev.1.00
Oct 27, 2008

DESCRIPTION

The 3804 group (Spec.L) is the 8-bit microcomputer based on the 740 family core technology.
The 3804 group (Spec.L) is designed for household products, office automation equipment, and controlling systems that require analog signal processing, including the A/D converter and D/A converters.

FEATURES

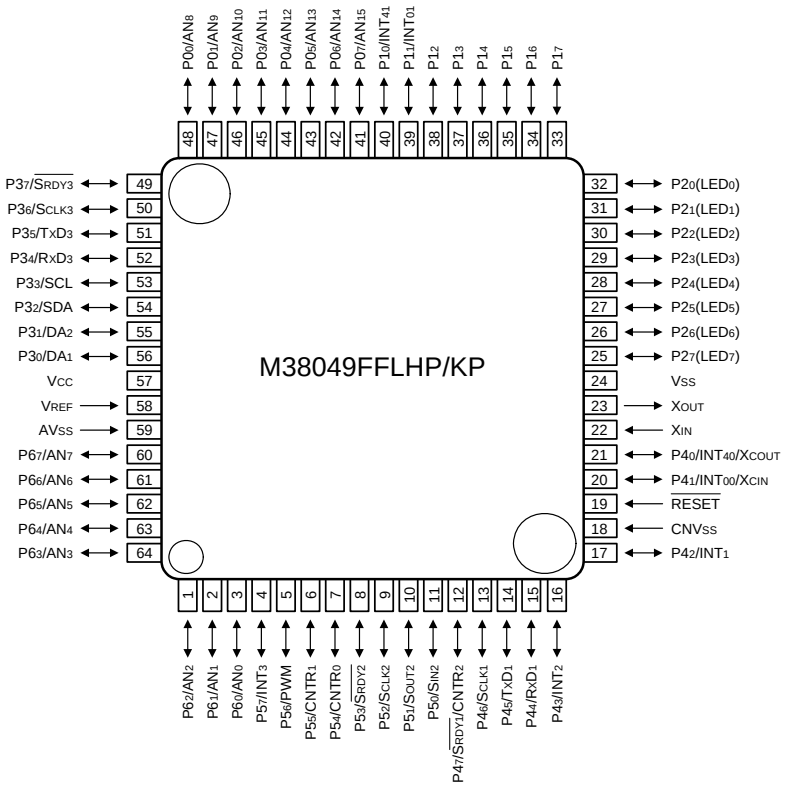
- Basic machine-language instructions 71
- Minimum instruction execution time ..... 0.24 µs
  (at 16.8 MHz oscillation frequency)
- Memory size
  ROM (Flash memory) ..... 60 K bytes
  RAM ..... 2048 bytes
- Programmable input/output ports 56
- Software pull-up resistors ..... Built-in
- Interrupts
  21 sources, 16 vectors .....
  (external 8, internal 12, software 1)
- Timers ..... 16-bit × 1
  8-bit × 4
  (with 8-bit prescaler)
- Serial interface ..... 8-bit × 2 (UART or Clock-synchronized)
  8-bit × 1 (Clock-synchronized)
- PWM ..... 8-bit × 1 (with 8-bit prescaler)
- A/D converter ..... 10-bit × 16 channels
  (8-bit reading enabled)
- D/A converter ..... 8-bit × 2 channels
- Watchdog timer ..... 16-bit × 1
- Multi-master I^2C-BUS interface ..... 1 channel
- LED direct drive port 8
- Clock generating circuit ..... Built-in 2 circuits
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage
  [In high-speed mode]
  At 16.8 MHz oscillation frequency ..... 4.5 to 5.5 V
  At 12.5 MHz oscillation frequency ..... 4.0 to 5.5 V
  At 8.4 MHz oscillation frequency ..... 2.7 to 5.5 V
  [In middle-speed mode]
  At 16.8 MHz oscillation frequency ..... 4.5 to 5.5 V
  At 12.5 MHz oscillation frequency ..... 2.7 to 5.5 V
  [In low-speed mode]
  At 32 kHz oscillation frequency ..... 2.7 to 5.5 V
- Power dissipation
  In high-speed mode ..... 27.5 mW (typ.)
  (at 16.8 MHz oscillation frequency, at 5 V power source voltage)
  In low-speed mode ..... 1200 µW (typ.)
  (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Operating temperature range ..... –20 to 85 °C
- Packages
  SP ..... PRDP0064BA-A (64P4B) (64-pin 750 mil SDIP)
  HP ..... PLQP0064KB-A (64P6Q-A) (64-pin 10 × 10 mm LQFP)
  KP ..... PLQP0064GA-A (64P6U-A) (64-pin 14 × 14 mm LQFP)
  WG ..... PTLG0064JA-A (64F0G) (64-pin 6 × 6 mm FLGA)

<Flash memory mode>

- Power source voltage ..... V_{CC} = 2.7 to 5.5 V
- Program/Erase voltage ..... V_{CC} = 2.7 to 5.5 V
- Programming method ..... Programming in unit of byte
- Erasing method ..... Block erasing
- Program/Erase control by software command
- Number of times for programming/erasing 100

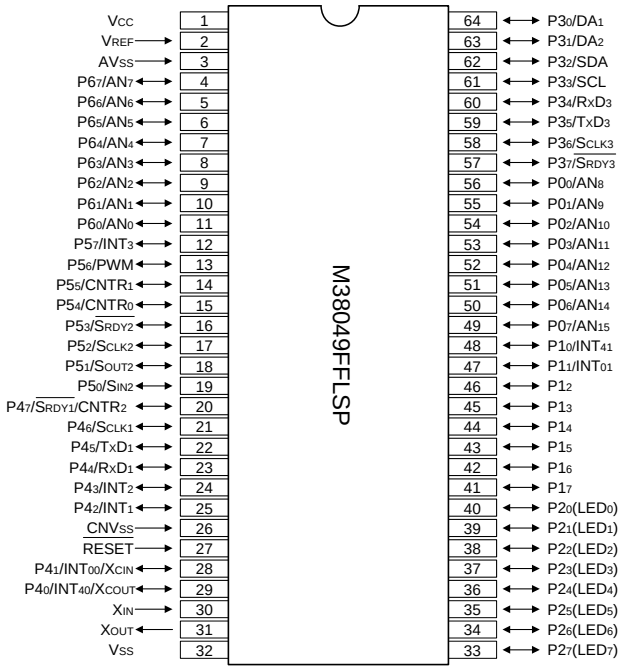
APPLICATION

Camera, Household appliance, Consumer electronics, etc.



Package code : PLQP0064KB-A (64P6Q-A)/PLQP0064GA-A (64P6U-A)

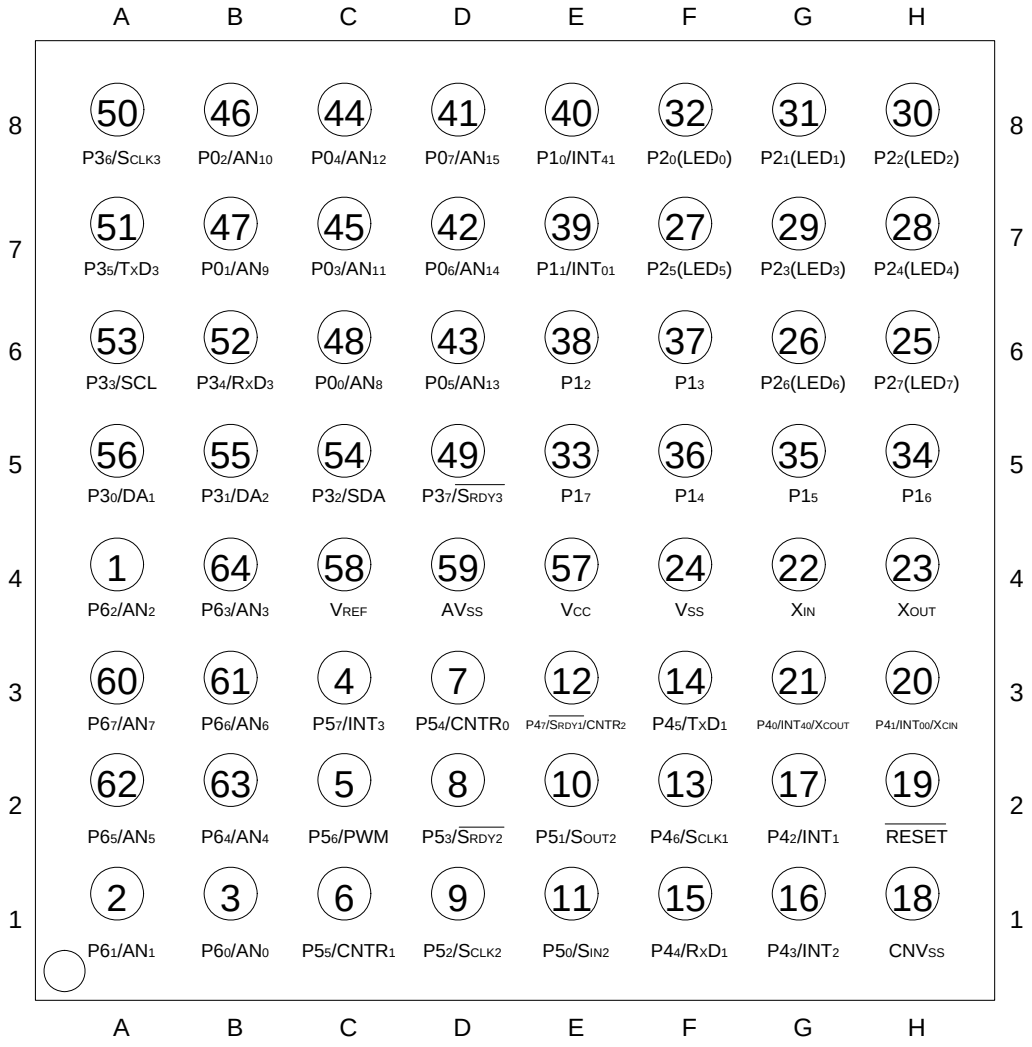
Fig. 1 Pin configuration (Top view) PLQP0064KB-A (64P6Q-A)/PLQP0064GA-A (64P6U-A)



Package code : PRDP0064BA-A (64P4B)

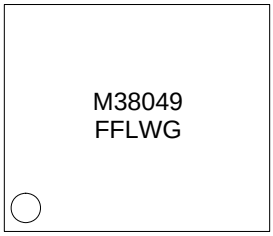
Fig. 2 Pin configuration (Top view) (PRDP0064BA-A (64P4B))

PIN CONFIGURATION (TOP VIEW)



Package code : PTLG0064JA-A (64F0G)

Note : The numbers in circles corresponds with the number on the packages HP/KP.



Package (TOP VIEW)

Fig. 3 Pin configuration (Top view) (PTLG0064JA-A (64F0G))

Table 1 Performance overview

Parameter			Function
Number of basic instructions			71
Minimum instruction execution time			0.24 μs (Oscillation frequency 16.8 MHz)
Oscillation frequency			Oscillation frequency 16.8 MHz (Maximum)
Memory sizes	ROM		60 Kbytes
	RAM		2048 bytes
I/O port	P0-P6		56 pins
Software pull-up resistors			Built-in
Interrupt			21 sources, 16 vectors (8 external, 12 internal, 1 software)
Timer			8-bit × 4 (with 8-bit prescaler), 16-bit × 1
Serial interface			8-bit × 2 (UART or Clock-synchronized) 8-bit × 1 (Clock-synchronized)
PWM			8-bit × 1 (with 8-bit prescaler)
A/D converter			10-bit × 16 channels (8-bit reading enabled)
D/A converter			8-bit × 2 channels
Watchdog timer			16-bit × 1
Multi-master I^2C-BUS interface			1 channel
LED direct drive port			8 (average current: 15 mA, peak current: 30 mA, total current: 90 mA)
Clock generating circuits			Built-in 2 circuits (connect to external ceramic resonator or quartz-crystal oscillator)
Power source voltage	In high-speed mode	At 16.8 MHz	4.5 to 5.5 V
		At 12.5 MHz	4.0 to 5.5 V
		At 8.4 MHz	2.7 to 5.5 V
	In middle-speed mode	At 16.8 MHz	4.5 to 5.5 V
		At 12.5 MHz	2.7 to 5.5 V
	In low-speed mode	At 32 MHz	2.7 to 5.5 V
Power dissipation	In high-speed mode		Typ. 27.5 mW (Vcc = 5.0 V, f(X_{IN}) = 16.8 MHz, Ta = 25 °C)
	In low-speed mode		Typ. 1200 μW (Vcc = 3.0 V, f(X_{IN}) = stop, f(X_{CIN}) = 32 kHz, Ta = 25 °C)
Operating temperature range			-20 to 85 °C
Device structure			CMOS silicon gate
Package			64-pin plastic molded SDIP/LQFP/FLGA

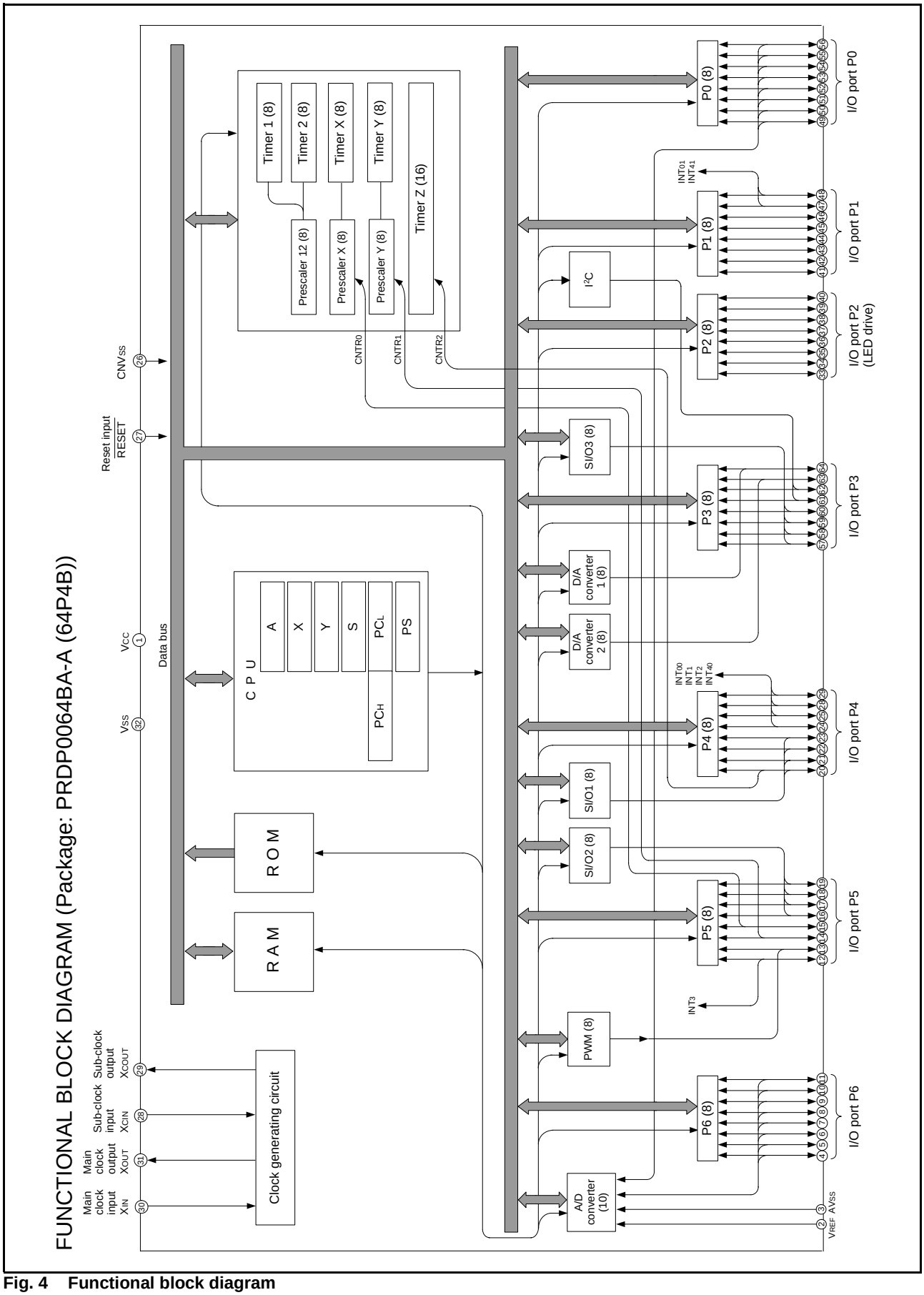


Fig. 4 Functional block diagram

PIN DESCRIPTION

Table 2 Pin description

Pin	Name	Functions	Function except a port function
V_{CC}, V_{SS}	Power source	• Apply voltage of 2.7 V – 5.5 V to V_{CC}, and 0 V to V_{SS}.	
CNV_{SS}	CNV_{SS} input	• This pin controls the operation mode of the chip. • Normally connected to V_{SS}.	
V_{REF}	Reference voltage	• Reference voltage input pin for A/D and D/A converters.	
AV_{SS}	Analog power source	• Analog power source input pin for A/D and D/A converters. • Connect to V_{SS}.	
RESET	Reset input	• Reset input pin for active “L”.	
X_{IN}	Main clock input	• Input and output pins for the clock generating circuit. • Connect a ceramic resonator or quartz-crystal oscillator between the X_{IN} and X_{OUT} pins to set the oscillation frequency. • When an external clock is used, connect the clock source to the X_{IN} pin and leave the X_{OUT} pin open.	
X_{OUT}	Main clock output		
$P0_0/AN_8$– $P0_7/AN_{15}$	I/O port P0	• 8-bit CMOS I/O port. • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level. • CMOS 3-state output structure. • Pull-up control is enabled in a bit unit. • $P2_0$ – $P2_7$ (8 bits) are enabled to output large current for LED drive.	• A/D converter input pin
$P1_0/INT_{41}$ $P1_1/INT_{01}$	I/O port P1		• Interrupt input pin
$P1_2$–$P1_7$			
$P2_0(LED_0)$- $P2_7(LED_7)$	I/O port P2		
$P3_0/DA_1$ $P3_1/DA_2$	I/O port P3	• 8-bit CMOS I/O port. • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level. • $P3_2$ to $P3_3$ can be switched between CMOS compatible input level or SMBUS input level in the I^2C-BUS interface function. • $P3_0$, $P3_1$, $P3_4$ – $P3_7$ are CMOS 3-state output structure. • $P3_2$, $P3_3$ are N-channel open-drain output structure. • Pull-up control of $P3_0$, $P3_1$, $P3_4$ – $P3_7$ is enabled in a bit unit.	• D/A converter input pin
$P3_2/SDA$ $P3_3/SCL$			• I^2C-BUS interface function pins
$P3_4/RxD_3$ $P3_5/TxD_3$ $P3_6/SCLK_3$ $P3_7/\overline{SRDY_3}$			• Serial I/O3 function pin
$P4_0/INT_{40}/XCOUT$ $P4_1/INT_{00}/XCIN$	I/O port P4	• 8-bit CMOS I/O port. • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level. • CMOS 3-state output structure. • Pull-up control is enabled in a bit unit.	• Interrupt input pin • Sub-clock generating I/O pin (resonator connected)
$P4_2/INT_1$ $P4_3/INT_2$			• Interrupt input pin
$P4_4/RxD_1$ $P4_5/TxD_1$ $P4_6/SCLK_1$			• Serial I/O1 function pin
$P4_7/\overline{SRDY_1}/CNTR_2$			• Serial I/O1, timer Z function pin
$P5_0/SIN_2$ $P5_1/SOUT_2$ $P5_2/SCLK_2$ $P5_3/\overline{SRDY_2}$	I/O port P5		• Serial I/O2 function pin
$P5_4/CNTR_0$			• Timer X function pin
$P5_5/CNTR_1$			• Timer Y function pin
$P5_6$/PWM			• PWM output pin
$P5_7/INT_3$			• Interrupt input pin
$P6_0/AN_0$– $P6_7/AN_7$	I/O port P6		• A/D converter input pin

PART NUMBERING

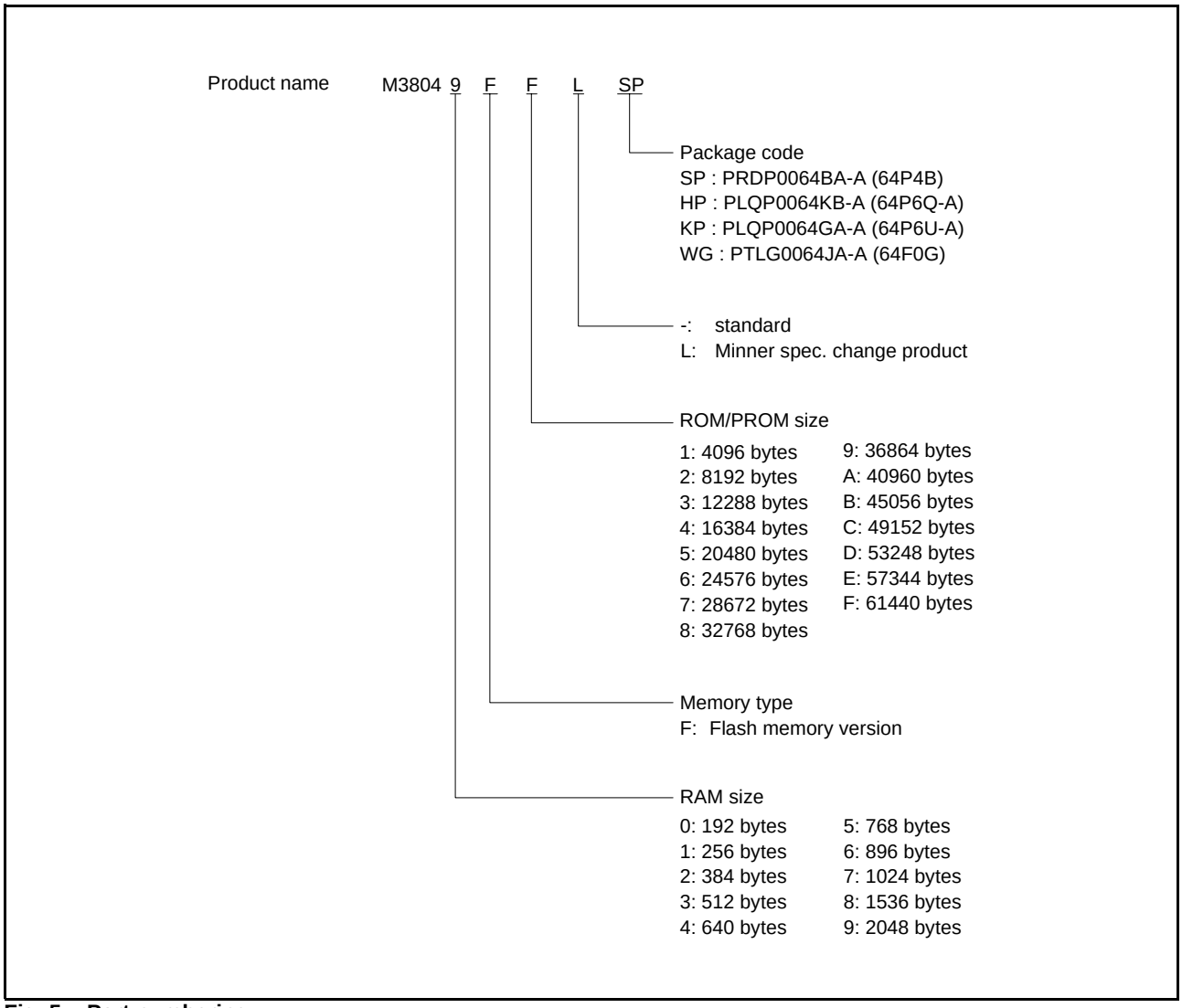


Fig. 5 Part numbering

GROUP EXPANSION

Renesas plans to expand the 3804 group (Spec.L) as follows.

Memory Size

- Flash memory size .....60 Kbytes
- RAM size ..... 2048 bytes

Packages

- PRDP0064BA-A (64P4B)
  ..... 64-pin shrink plastic-molded SDIP
- PLQP0064KB-A (64P6Q-A)
  .....0.5 mm-pitch plastic molded LQFP
- PLQP0064GA-A (64P6U-A)
  .....0.8 mm-pitch plastic molded LQFP
- PTLG0064JA-A (64F0G)
  .....0.65 mm-pitch plastic molded FLGA

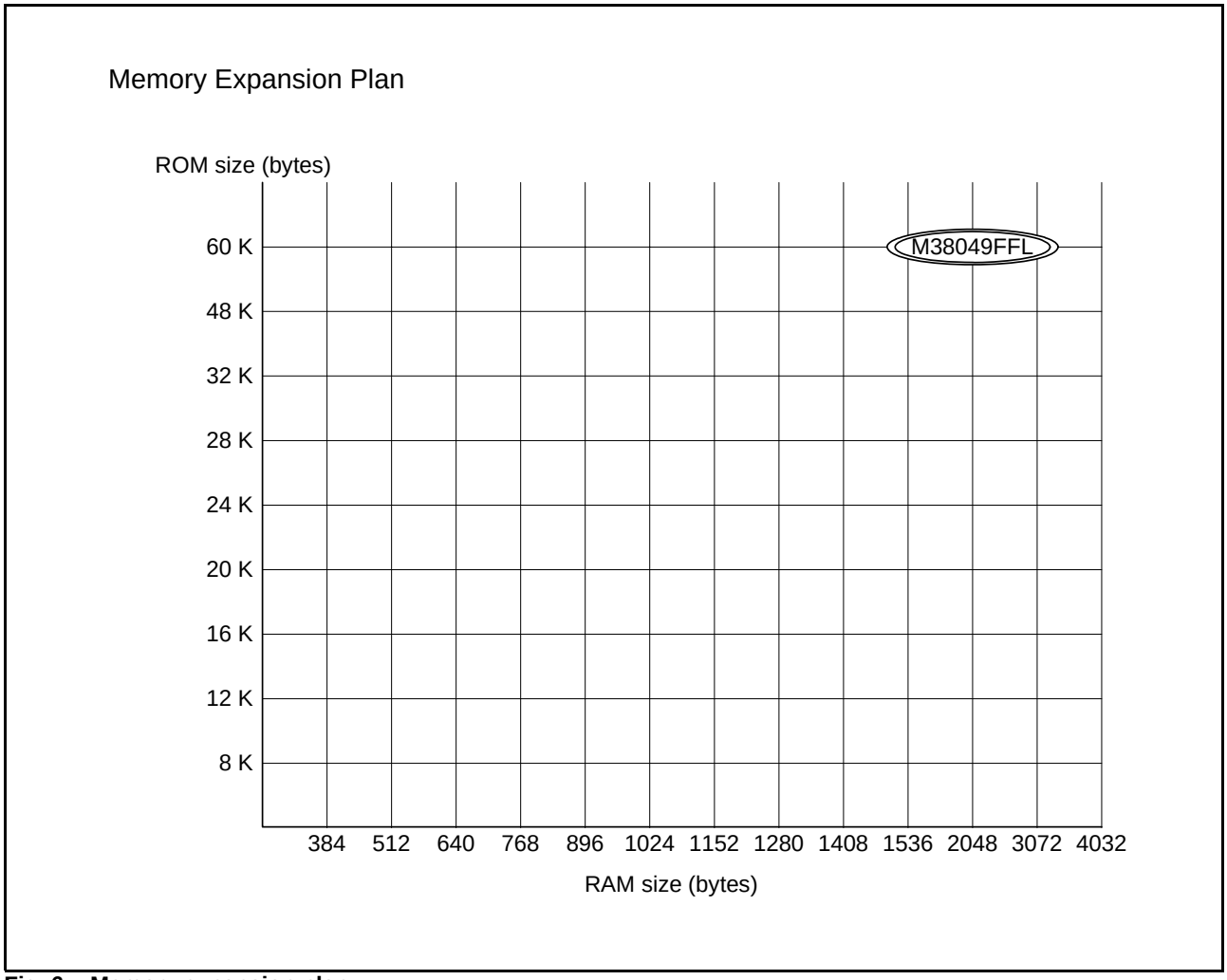


Fig. 6 Memory expansion plan

Table 3 Support products

Part No.	ROM size (bytes)	RAM size (bytes)	Package	Remarks
M38049FFLSP	57344+4096 (NOTE)	2048	PRDP0064BA-A (64P4B)	V_{CC} = 2.7 to 5.5 V
M38049FFLHP			PLQP0064KB-A (64P6Q-A)	
M38049FFLKP			PLQP0064GA-A (64P6U-A)	
M38049FFLWG			PTLG0064JA-A (64F0G)	

NOTE:
1. ROM size includes the ID code area.

FUNCTIONAL DESCRIPTION

CENTRAL PROCESSING UNIT (CPU)

The 3804 group (Spec.L) uses the standard 740 Family instruction set. Refer to the 740 Family Software Manual for details on the instruction set.
Machine-resident 740 Family instructions are as follows:
The FST and SLW instructions cannot be used.
The STP, WIT, MUL, and DIV instructions can be used.

[Accumulator (A)]

The accumulator is an 8-bit register. Data operations such as data transfer, etc. are executed mainly through the accumulator.

[Index Register X (X)]

The index register X is an 8-bit register. In the index addressing modes, the value of the OPERAND is added to the contents of register X and specifies the real address.

[Index Register Y (Y)]

The index register Y is an 8-bit register. In partial instruction, the value of the OPERAND is added to the contents of register Y and specifies the real address.

[Stack Pointer (S)]

The stack pointer is an 8-bit register used during subroutine calls and interrupts. This register indicates start address of stored area (stack) for storing registers during subroutine calls and interrupts.
The low-order 8 bits of the stack address are determined by the contents of the stack pointer. The high-order 8 bits of the stack address are determined by the stack page selection bit. If the stack page selection bit is “0”, the high-order 8 bits becomes “00_{16}”. If the stack page selection bit is “1”, the high-order 8 bits becomes “01_{16}”.
The operations of pushing register contents onto the stack and popping them from the stack are shown in Figure 8.
Store registers other than those described in Figure 7 with program when the user needs them during interrupts or subroutine calls (see Table 4).

[Program Counter (PC)]

The program counter is a 16-bit counter consisting of two 8-bit registers PC_H and PC_L. It is used to indicate the address of the next instruction to be executed.

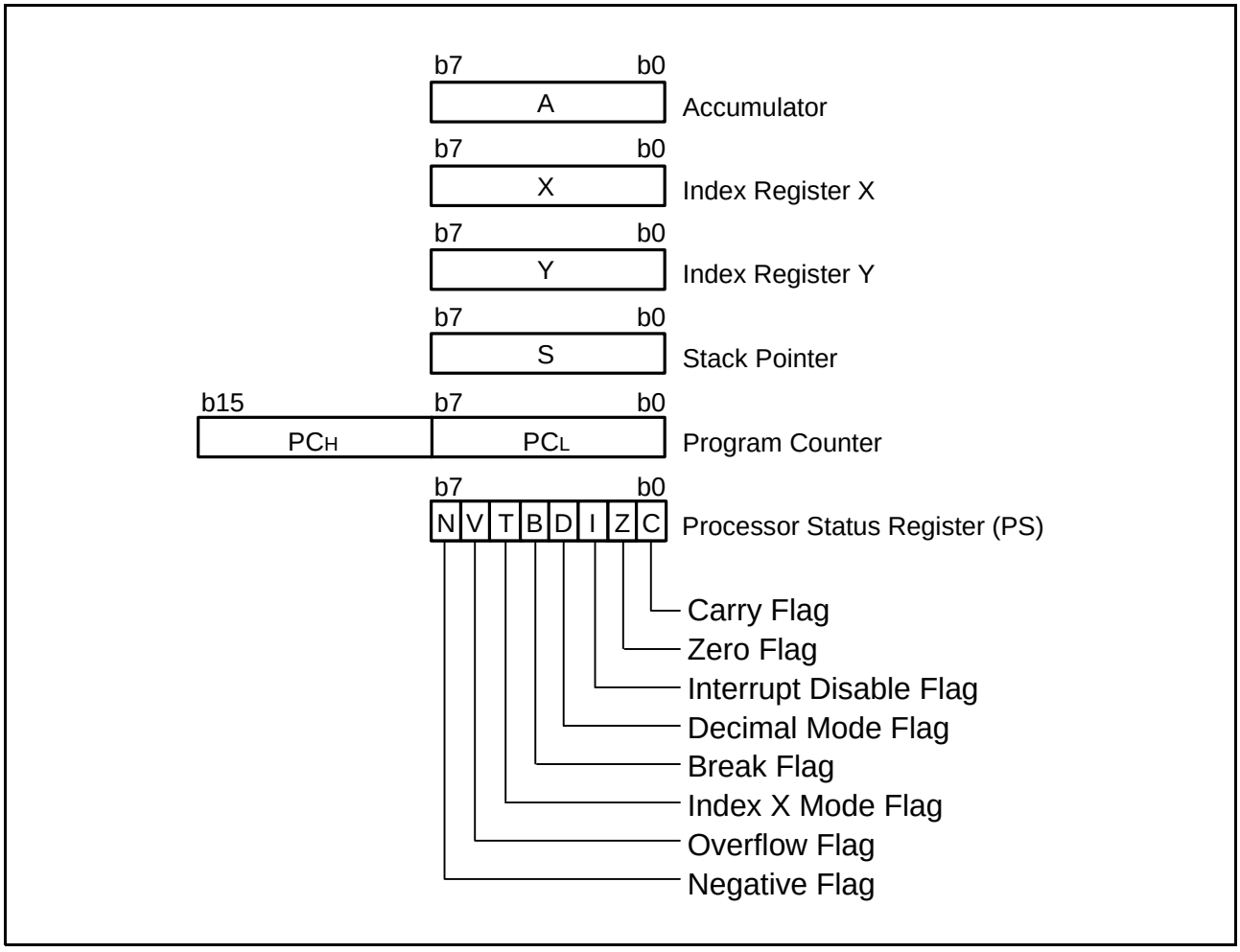


Fig. 7 740 Family CPU register structure

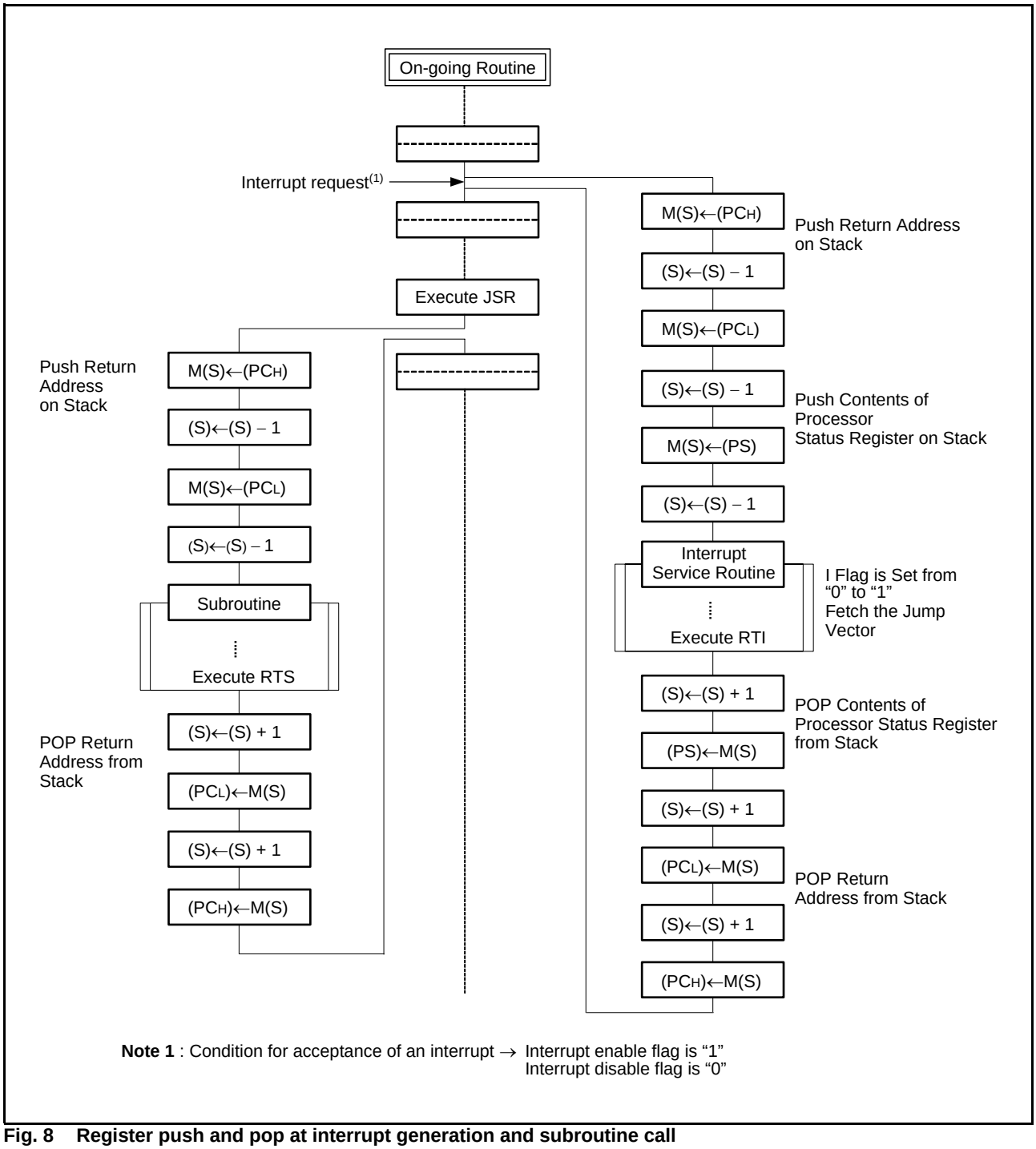


Fig. 8 Register push and pop at interrupt generation and subroutine call

Table 4 Push and pop instructions of accumulator or processor status register

	Push instruction to stack	Pop instruction from stack
Accumulator	PHA	PLA
Processor status register	PHP	PLP

[Processor status register (PS)]

The processor status register is an 8-bit register consisting of 5 flags which indicate the status of the processor after an arithmetic operation and 3 flags which decide MCU operation. Branch operations can be performed by testing the Carry (C) flag, Zero (Z) flag, Overflow (V) flag, or the Negative (N) flag. In decimal mode, the Z, V, N flags are not valid.

Bit 0: Carry flag (C)

The C flag contains a carry or borrow generated by the arithmetic logic unit (ALU) immediately after an arithmetic operation. It can also be changed by a shift or rotate instruction.

Bit 1: Zero flag (Z)

The Z flag is set if the result of an immediate arithmetic operation or a data transfer is “0”, and cleared if the result is anything other than “0”.

Bit 2: Interrupt disable flag (I)

The I flag disables all interrupts except for the interrupt generated by the BRK instruction.
Interrupts are disabled when the I flag is “1”.

Bit 3: Decimal mode flag (D)

The D flag determines whether additions and subtractions are executed in binary or decimal. Binary arithmetic is executed when this flag is “0”; decimal arithmetic is executed when it is “1”.
Decimal correction is automatic in decimal mode. Only the ADC and SBC instructions can execute decimal arithmetic.

Bit 4: Break flag (B)

The B flag is used to indicate that the current interrupt was generated by the BRK instruction. The BRK flag in the processor status register is always “0”. When the BRK instruction is used to generate an interrupt, the processor status register is pushed onto the stack with the break flag set to “1”.

Bit 5: Index X mode flag (T)

When the T flag is “0”, arithmetic operations are performed between accumulator and memory. When the T flag is “1”, direct arithmetic operations and direct data transfers are enabled between memory locations.

Bit 6: Overflow flag (V)

The V flag is used during the addition or subtraction of one byte of signed data. It is set if the result exceeds +127 to –128. When the BIT instruction is executed, bit 6 of the memory location operated on by the BIT instruction is stored in the overflow flag.

Bit 7: Negative flag (N)

The N flag is set if the result of an arithmetic operation or data transfer is negative. When the BIT instruction is executed, bit 7 of the memory location operated on by the BIT instruction is stored in the negative flag.

Table 5 Set and clear instructions of each bit of processor status register

	C flag	Z flag	I flag	D flag	B flag	T flag	V flag	N flag
Set instruction	SEC	–	SEI	SED	–	SET	–	–
Clear instruction	CLC	–	CLI	CLD	–	CLT	CLV	–

[CPU Mode Register (CPUM)] 003B16

The CPU mode register contains the stack page selection bit, the internal system clock control bits, etc.
The CPU mode register is allocated at address 003B16.

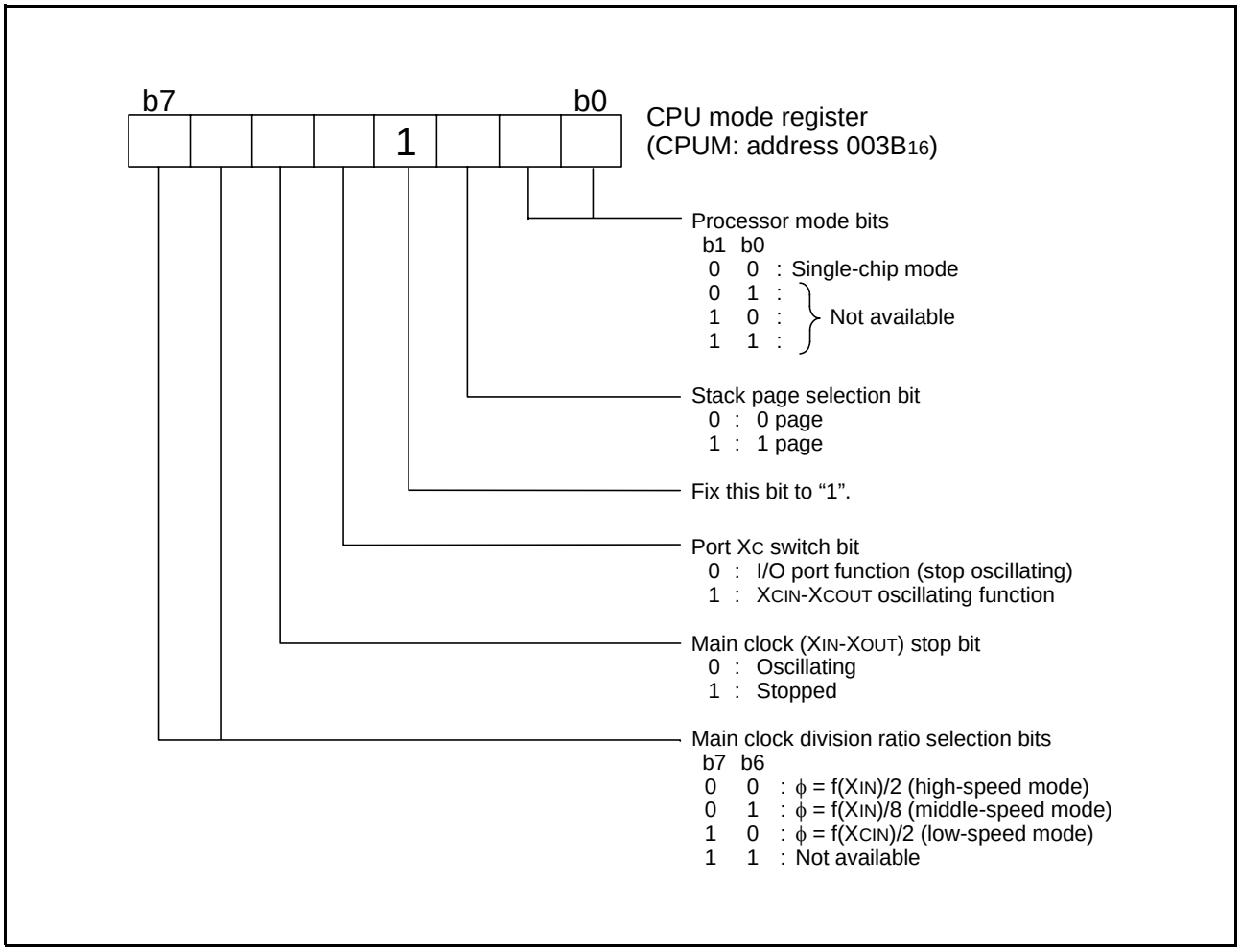


Fig. 9 Structure of CPU mode register

MISRG

(1) Bit 0 of address 0010$_{16}$: Oscillation stabilizing time set after STP instruction released bit

When the MCU stops the clock oscillation by the STP instruction and the STP instruction has been released by an external interrupt source, usually, the fixed values of Timer 1 and Prescaler 12 (Timer 1 = 01$_{16}$, Prescaler 12 = FF$_{16}$) are automatically reloaded in order for the oscillation to stabilize. The user can inhibit the automatic setting by setting "1" to bit 0 of MISRG (address 0010$_{16}$).
However, by setting this bit to "1", the previous values, set just before the STP instruction was executed, will remain in Timer 1 and Prescaler 12. Therefore, you will need to set an appropriate value to each register, in accordance with the oscillation stabilizing time, before executing the STP instruction.
Figure 10 shows the structure of MISRG.

(2) Bits 1, 2, 3 of address 0010$_{16}$: Middle-speed Mode Automatic Switch Function

In order to switch the clock mode of an MCU which has a sub-clock, the following procedure is necessary:
set CPU mode register (003B$_{16}$) --> start main clock oscillation --> wait for oscillation stabilization --> switch to middle-speed mode (or high-speed mode).
However, the 3804 group (Spec.L) has the built-in function which automatically switches from low to middle-speed mode by program.

• Middle-speed mode automatic switch by program

The middle-speed mode can also be automatically switched by program while operating in low-speed mode. By setting the middle-speed automatic switch start bit (bit 3) of MISRG (address 0010$_{16}$) to "1" in the condition that the middle-speed mode automatic switch set bit is "1" while operating in low-speed mode, the MCU will automatically switch to middle-speed mode. In this case, the oscillation stabilizing time of the main clock can be selected by the middle-speed automatic switch wait time set bit (bit 2) of MISRG (address 0010$_{16}$).

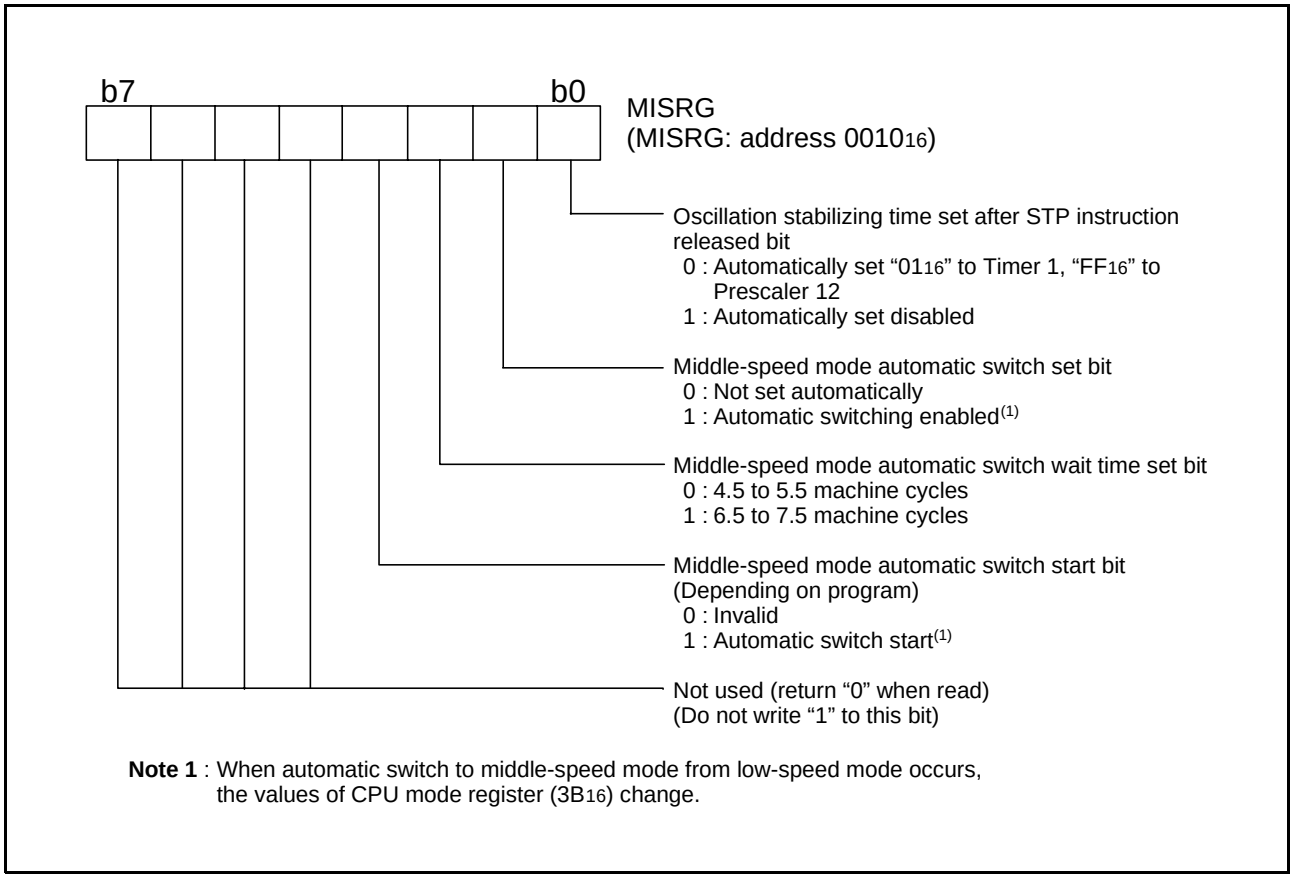


Fig. 10 Structure of MISRG

MEMORY

• Special Function Register (SFR) Area
The Special Function Register area in the zero page contains control registers such as I/O ports and timers.

• RAM
The RAM is used for data storage and for stack area of subroutine calls and interrupts.

• ROM
The ROM area can program/erase.

• Interrupt Vector Area
The interrupt vector area contains reset and interrupt vectors.

• Zero Page
Access to this area with only 2 bytes is possible in the zero page addressing mode.

• Special Page
Access to this area with only 2 bytes is possible in the special page addressing mode.

<Note>
Since the contents of RAM are undefined at reset, be sure to set an initial value before use.

RAM area

RAM size (bytes)	Address $XXXX_{16}$
192	$00FF_{16}$
256	$013F_{16}$
384	$01BF_{16}$
512	$023F_{16}$
640	$02BF_{16}$
768	$033F_{16}$
896	$03BF_{16}$
1024	$043F_{16}$
1536	$063F_{16}$
2048	$083F_{16}$

ROM area

ROM size (bytes)	Address $YYYY_{16}$
4096	$F000_{16}$
8192	$E000_{16}$
12288	$D000_{16}$
16384	$C000_{16}$
20480	$B000_{16}$
24576	$A000_{16}$
28672	9000_{16}
32768	8000_{16}
36864	7000_{16}
40960	6000_{16}
45056	5000_{16}
49152	4000_{16}
53248	3000_{16}
57344	2000_{16}
61440	1000_{16}

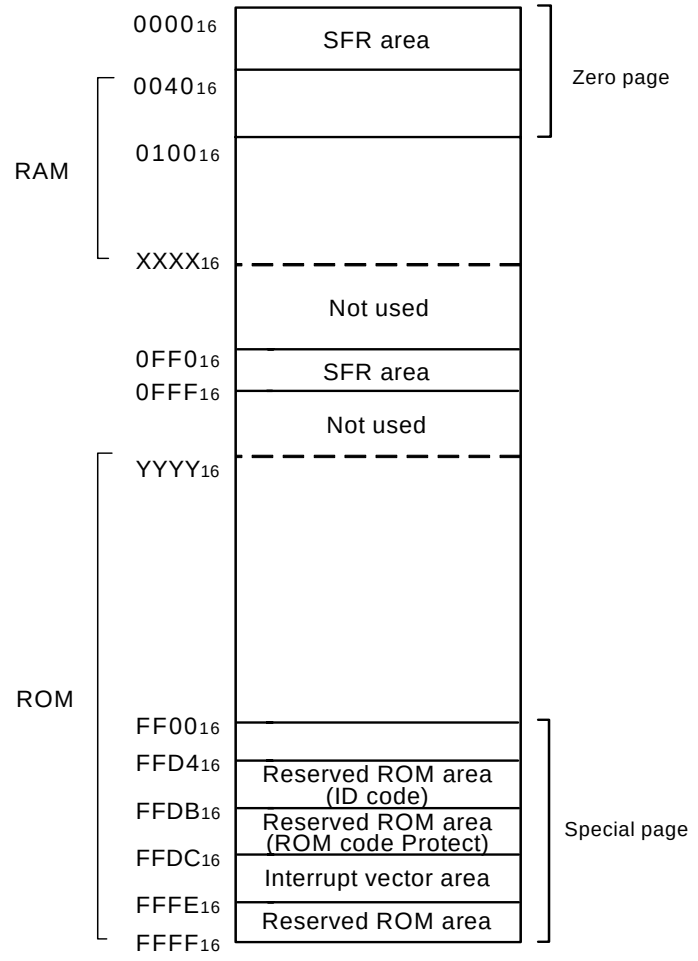


Fig. 11 Memory map diagram

0000₁₆	Port P0 (P0)
0001₁₆	Port P0 direction register (P0D)
0002₁₆	Port P1 (P1)
0003₁₆	Port P1 direction register (P1D)
0004₁₆	Port P2 (P2)
0005₁₆	Port P2 direction register (P2D)
0006₁₆	Port P3 (P3)
0007₁₆	Port P3 direction register (P3D)
0008₁₆	Port P4 (P4)
0009₁₆	Port P4 direction register (P4D)
000A₁₆	Port P5 (P5)
000B₁₆	Port P5 direction register (P5D)
000C₁₆	Port P6 (P6)
000D₁₆	Port P6 direction register (P6D)
000E₁₆	Timer 12, X count source selection register (T12XCSS)
000F₁₆	Timer Y, Z count source selection register (TYZCSS)
0010₁₆	MISRG
0011₁₆	I^2C data shift register (S0)
0012₁₆	I^2C special mode status register (S3)
0013₁₆	I^2C status register (S1)
0014₁₆	I^2C control register (S1D)
0015₁₆	I^2C clock control register (S2)
0016₁₆	I^2C START/STOP condition control register (S2D)
0017₁₆	I^2C special mode control register (S3D)
0018₁₆	Transmit/Receive buffer register 1 (TB1/RB1)
0019₁₆	Serial I/O1 status register (SIO1STS)
001A₁₆	Serial I/O1 control register (SIO1CON)
001B₁₆	UART1 control register (UART1CON)
001C₁₆	Baud rate generator (BRG1)
001D₁₆	Serial I/O2 control register (SIO2CON)
001E₁₆	Watchdog timer control register (WDTCON)
001F₁₆	Serial I/O2 register (SIO2)

0020₁₆	Prescaler 12 (PRE12)
0021₁₆	Timer 1 (T1)
0022₁₆	Timer 2 (T2)
0023₁₆	Timer XY mode register (TM)
0024₁₆	Prescaler X (PREX)
0025₁₆	Timer X (TX)
0026₁₆	Prescaler Y (PREY)
0027₁₆	Timer Y (TY)
0028₁₆	Timer Z low-order (TZL)
0029₁₆	Timer Z high-order (TZH)
002A₁₆	Timer Z mode register (TZM)
002B₁₆	PWM control register (PWMCON)
002C₁₆	PWM prescaler (PREPWM)
002D₁₆	PWM register (PWM)
002E₁₆	
002F₁₆	Baud rate generator 3 (BRG3)
0030₁₆	Transmit/Receive buffer register 3 (TB3/RB3)
0031₁₆	Serial I/O3 status register (SIO3STS)
0032₁₆	Serial I/O3 control register (SIO3CON)
0033₁₆	UART3 control register (UART3CON)
0034₁₆	AD/DA control register (ADCON)
0035₁₆	AD conversion register 1 (AD1)
0036₁₆	DA1 conversion register (DA1)
0037₁₆	DA2 conversion register (DA2)
0038₁₆	AD conversion register 2 (AD2)
0039₁₆	Interrupt source selection register (INTSEL)
003A₁₆	Interrupt edge selection register (INTEDGE)
003B₁₆	CPU mode register (CPUM)
003C₁₆	Interrupt request register 1 (IREQ1)
003D₁₆	Interrupt request register 2 (IREQ2)
003E₁₆	Interrupt control register 1 (ICON1)
003F₁₆	Interrupt control register 2 (ICON2)

0FE0₁₆	Flash memory control register 0 (FMCR0)
0FE1₁₆	Flash memory control register 1 (FMCR1)
0FE2₁₆	Flash memory control register 2 (FMCR2)
0FE3₁₆	Reserved (1)
0FE4₁₆	Reserved (1)
0FE5₁₆	Reserved (1)
0FE6₁₆	Reserved (1)
0FE7₁₆	Reserved (1)
0FE8₁₆	Reserved (1)
0FE9₁₆	Reserved (1)
0FEA₁₆	Reserved (1)
0FEB₁₆	Reserved (1)
0FEC₁₆	Reserved (1)
0FED₁₆	Reserved (1)
0FEE₁₆	Reserved (1)
0FEF₁₆	Reserved (1)

0FF0₁₆	Port P0 pull-up control register (PULL0)
0FF1₁₆	Port P1 pull-up control register (PULL1)
0FF2₁₆	Port P2 pull-up control register (PULL2)
0FF3₁₆	Port P3 pull-up control register (PULL3)
0FF4₁₆	Port P4 pull-up control register (PULL4)
0FF5₁₆	Port P5 pull-up control register (PULL5)
0FF6₁₆	Port P6 pull-up control register (PULL6)
0FF7₁₆	I^2C slave address register 0 (S0D0)
0FF8₁₆	I^2C slave address register 1 (S0D1)
0FF9₁₆	I^2C slave address register 2 (S0D2)

Note 1: Do not write any data to these addresses, because these are reserved area.

Fig. 12 Memory map of special function register (SFR)

I/O PORTS

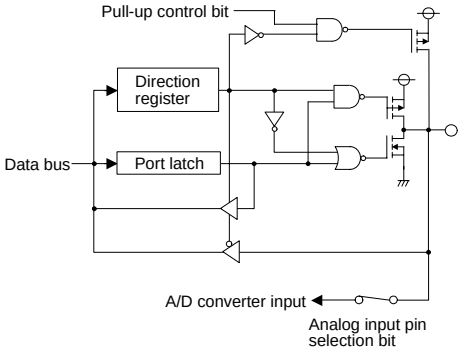
The I/O ports have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, and each pin can be set to be input port or output port.
When "0" is written to the bit corresponding to a pin, that pin becomes an input pin. When "1" is written to that bit, that pin becomes an output pin.
If data is read from a pin which is set to output, the value of the port output latch is read, not the value of the pin itself. Pins set to input are floating. If a pin set to input is written to, only the port output latch is written to and the pin remains floating.
By setting the port P0 pull-up control register (address 0FF0$_{16}$) to the port P6 pull-up control register (address 0FF6$_{16}$) ports can control pull-up with a program. However, the contents of these registers do not affect ports programmed as the output ports.

Table 6 I/O port function

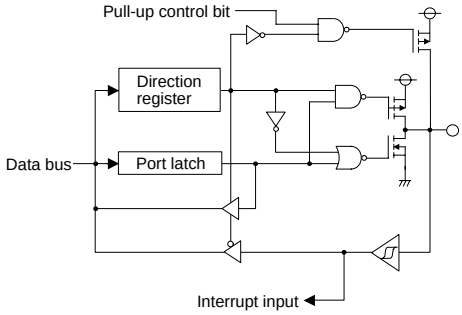
Pin	Name	Input/ Output	I/O Structure	Non-Port Function	Related SFRs	Ref. No.
$P0_0/AN_8$–$P0_7/AN_{15}$	Port P0	Input/output, individual bits	CMOS compatible input level CMOS 3-state output	A/D converter input	AD/DA control register	(1)
$P1_0/INT_{41}$ $P1_1/INT_{01}$	Port P1			External interrupt input	Interrupt edge selection register	(2)
$P1_2$–$P1_7$						(3)
$P2_0(LED_0)$– $P2_7(LED_7)$	Port P2					
$P3_0/DA_1$ $P3_1/DA_2$	Port P3			D/A converter output	AD/DA control register	(4)
$P3_2$/SDA $P3_3$/SCL			CMOS compatible input level N-channel open-drain output CMOS/SMBUS input level (when selecting I^2C-BUS interface function)	I^2C-BUS interface function I/O	I^2C control register	(5)
$P3_4/RxD_3$ $P3_5/TxD_3$ $P3_6/SCLK_3$ $P3_7/\overline{SRDY_3}$			CMOS compatible input level CMOS 3-state output	Serial I/O3 function I/O	Serial I/O3 control register UART3 control register	(6) (7) (8) (9)
$P4_0/INT_{40}/XCOUT$ $P4_1/INT_{00}/XCIN$	Port P4			External interrupt input Sub-clock generating circuit	Interrupt edge selection register CPU mode register	(10) (11)
$P4_2/INT_1$ $P4_3/INT_2$				External interrupt input	Interrupt edge selection register	(2)
$P4_4/RxD_1$ $P4_5/TxD_1$ $P4_6/SCLK_1$				Serial I/O1 function I/O	Serial I/O1 control register UART1 control register	(6) (7) (8)
$P4_7/\overline{SRDY_1}/CNTR_2$				Serial I/O1 function I/O Timer Z function I/O	Serial I/O1 control register Timer Z mode register	(12)
$P5_0/SIN_2$ $P5_1/SOUT_2$ $P5_2/SCLK_2$ $P5_3/\overline{SRDY_2}$	Port P5			Serial I/O2 function I/O	Serial I/O2 control register	(13) (14) (15) (16)
$P5_4/CNTR_0$ $P5_5/CNTR_1$				Timer X, Y function I/O	Timer XY mode register	(17)
$P5_6$/PWM				PWM output	PWM control register	(18)
$P5_7/INT_3$				External interrupt input	Interrupt edge selection register	(2)
$P6_0/AN_0$–$P6_7/AN_7$	Port P6			A/D converter input	AD/DA control register	(1)

NOTES:
1. Refer to the applicable sections how to use double-function ports as function I/O ports.
2. Make sure that the input level at each pin is either 0 V or V_{CC} during execution of the STP instruction.
   When an input level is at an intermediate potential, a current will flow from V_{CC} to V_{SS} through the input-stage gate.

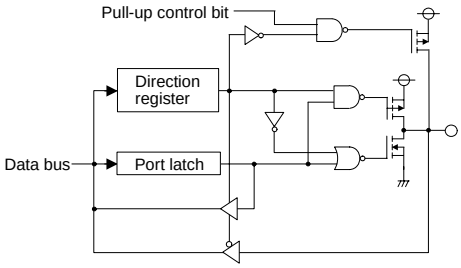
(1) Ports P0, P6



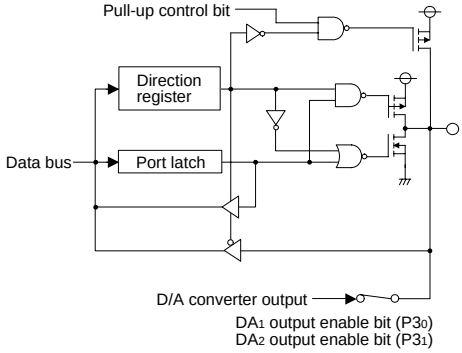
(2) Ports P10, P11, P42, P43, P57



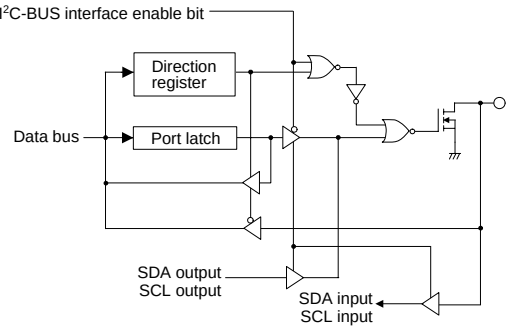
(3) Ports P12 to P17, P2



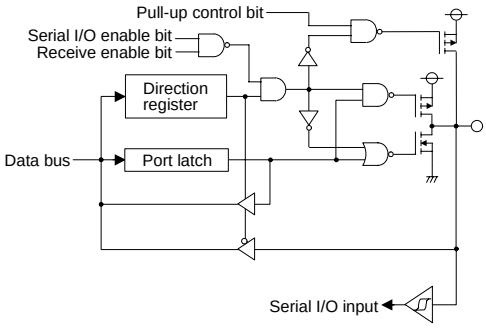
(4) Ports P30, P31



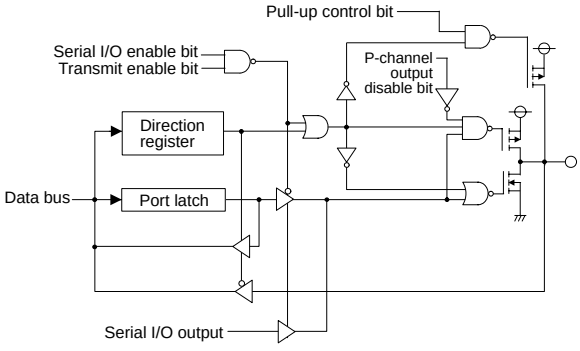
(5) Ports P32, P33



(6) Ports P34, P44



(7) Ports P35, P45



(8) Ports P36, P46

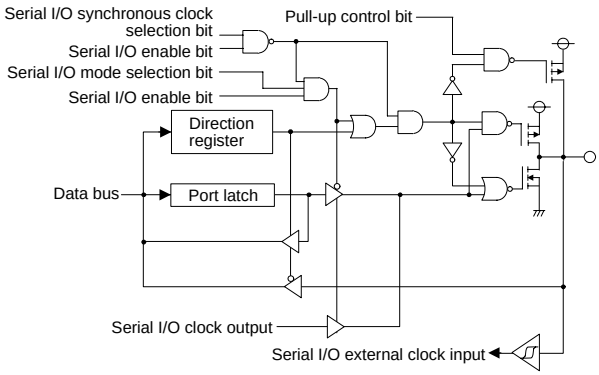


Fig. 13 Port block diagram (1)

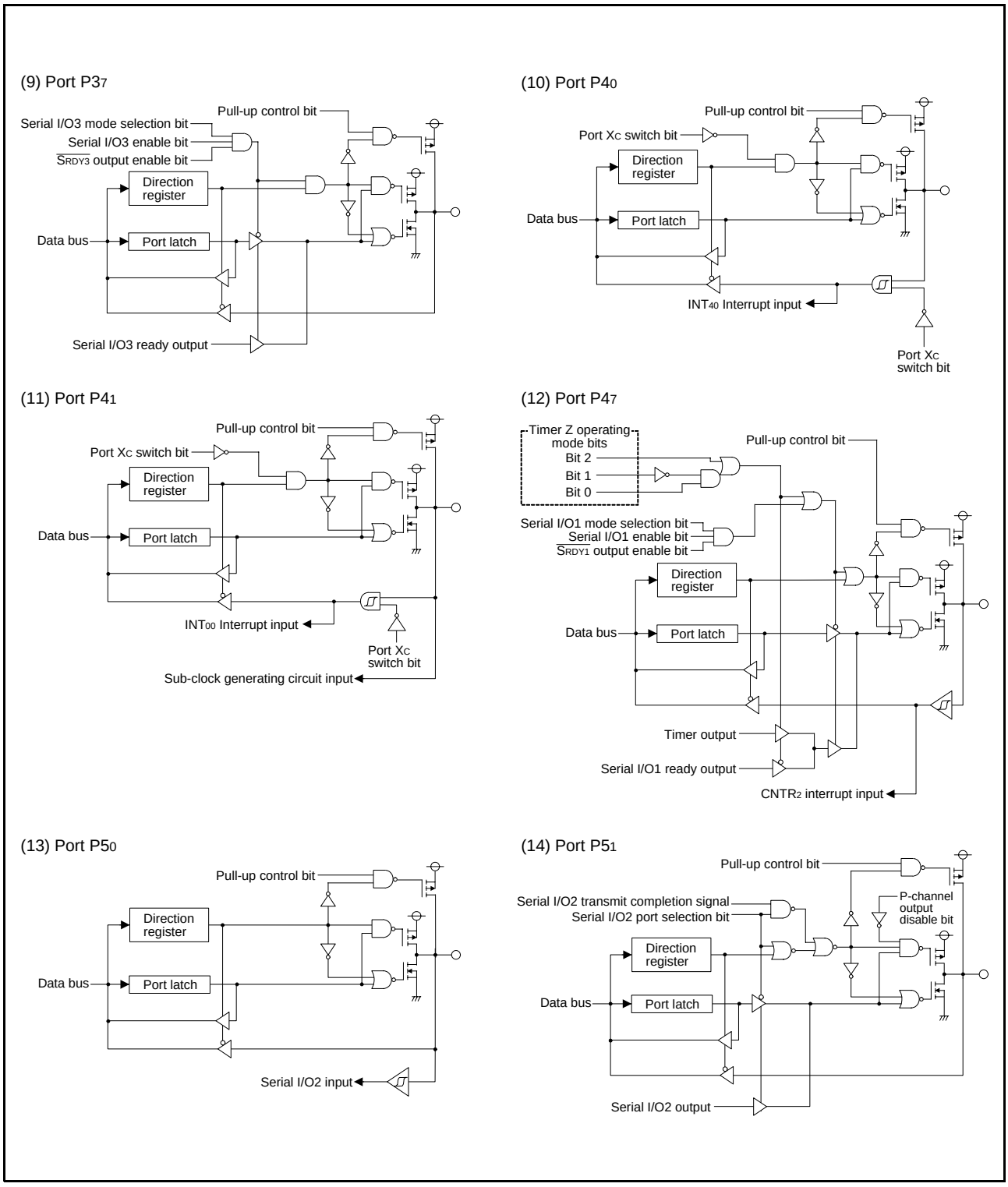


Fig. 14 Port block diagram (2)

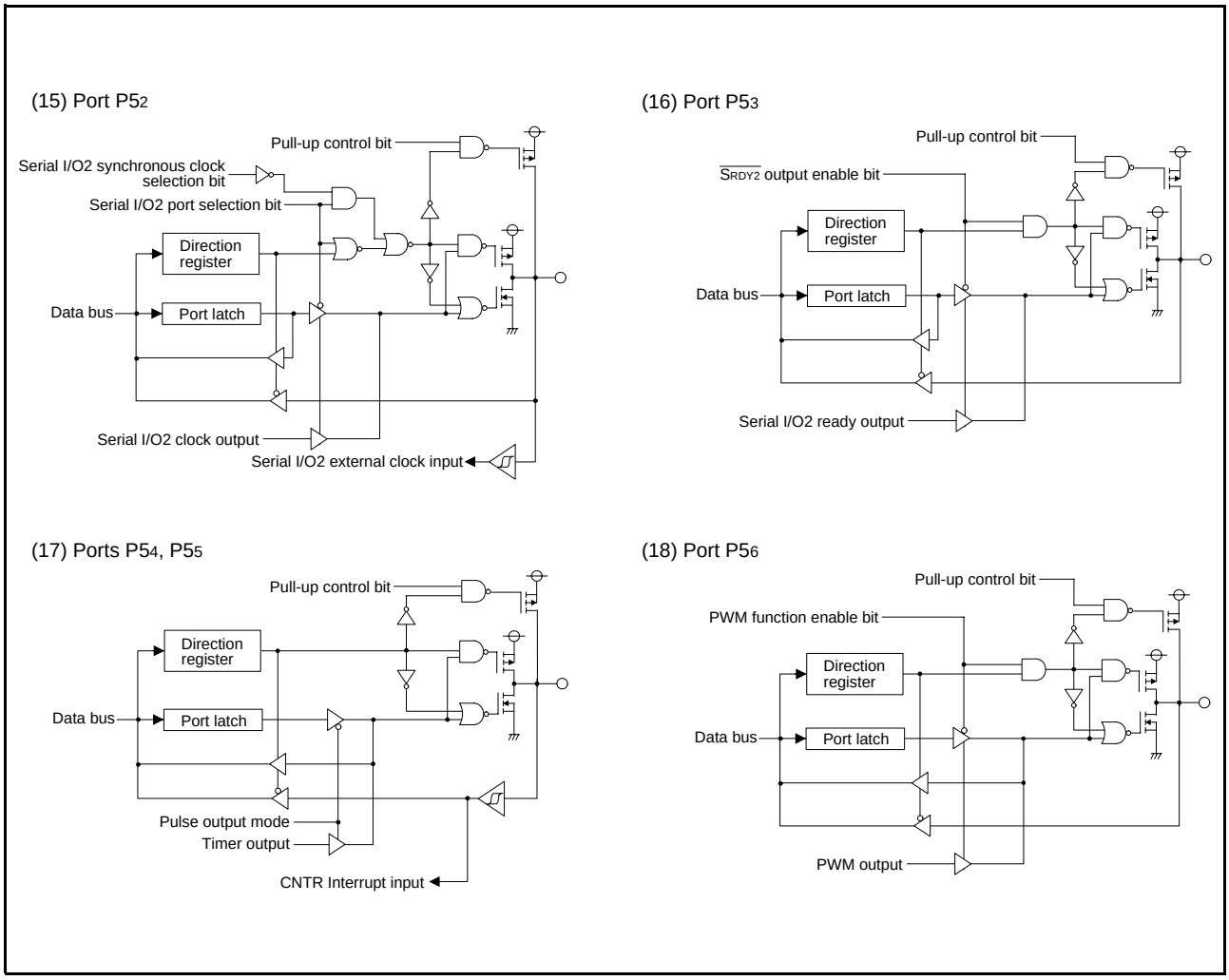
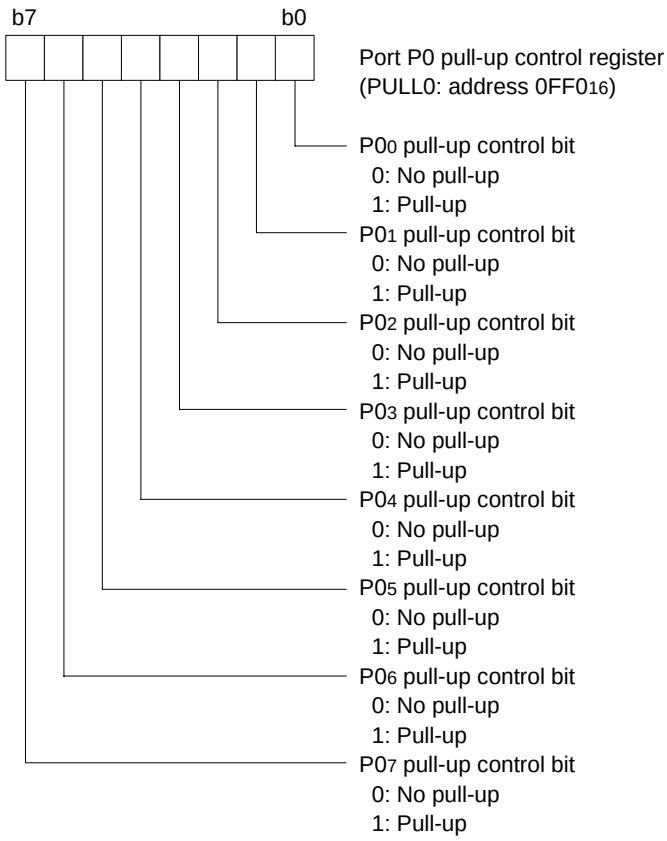
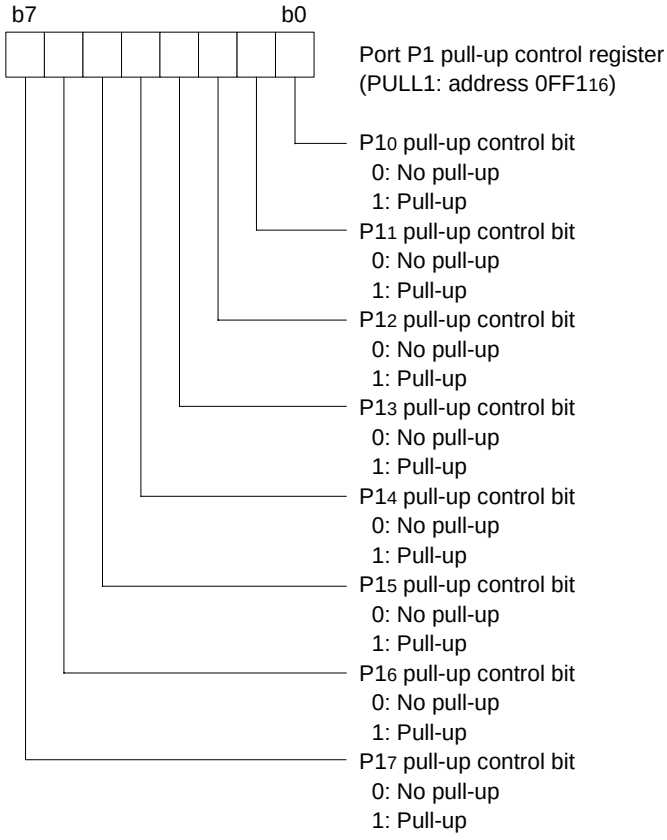


Fig. 15 Port block diagram (3)

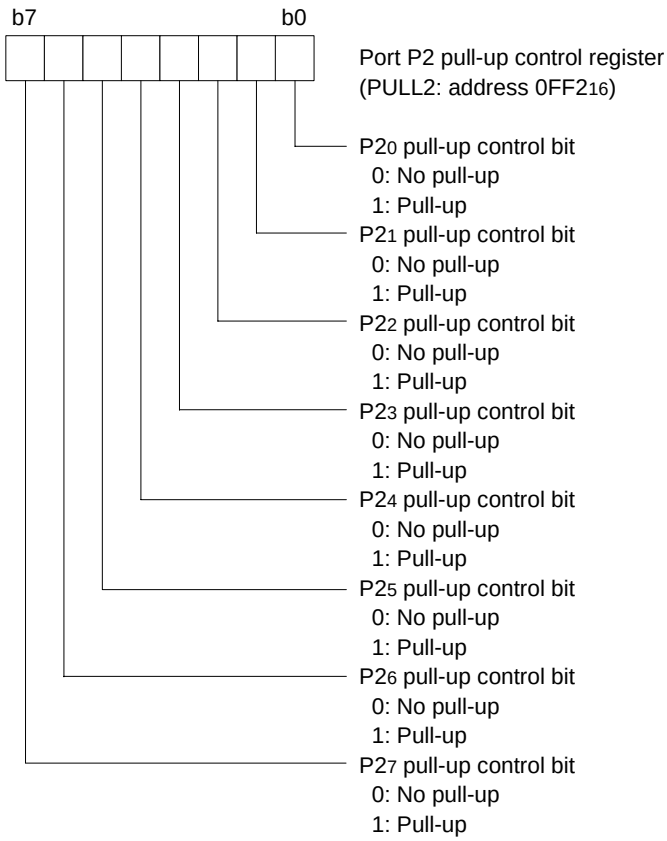


Note: Pull-up control is valid when the corresponding bit of the port direction register is "0" (input). When that bit is "1" (output), pull-up cannot be set to the port of which pull-up is selected.

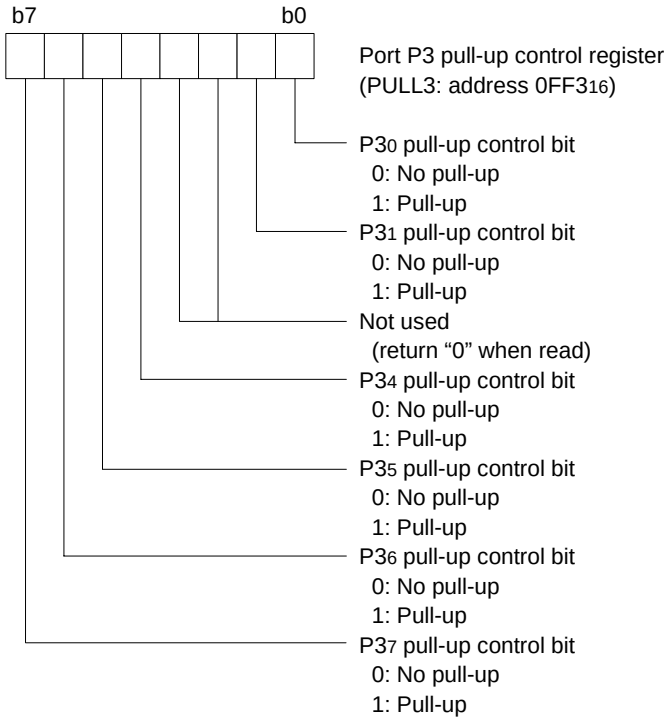


Note: Pull-up control is valid when the corresponding bit of the port direction register is "0" (input). When that bit is "1" (output), pull-up cannot be set to the port of which pull-up is selected.

Fig. 16 Structure of port pull-up control register (1)

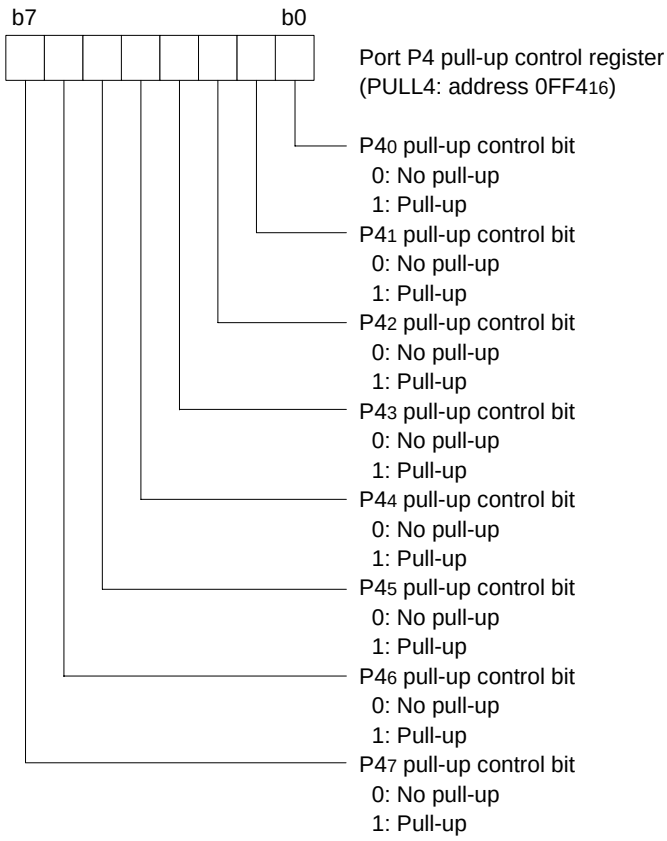


Note: Pull-up control is valid when the corresponding bit of the port direction register is "0" (input). When that bit is "1" (output), pull-up cannot be set to the port of which pull-up is selected.

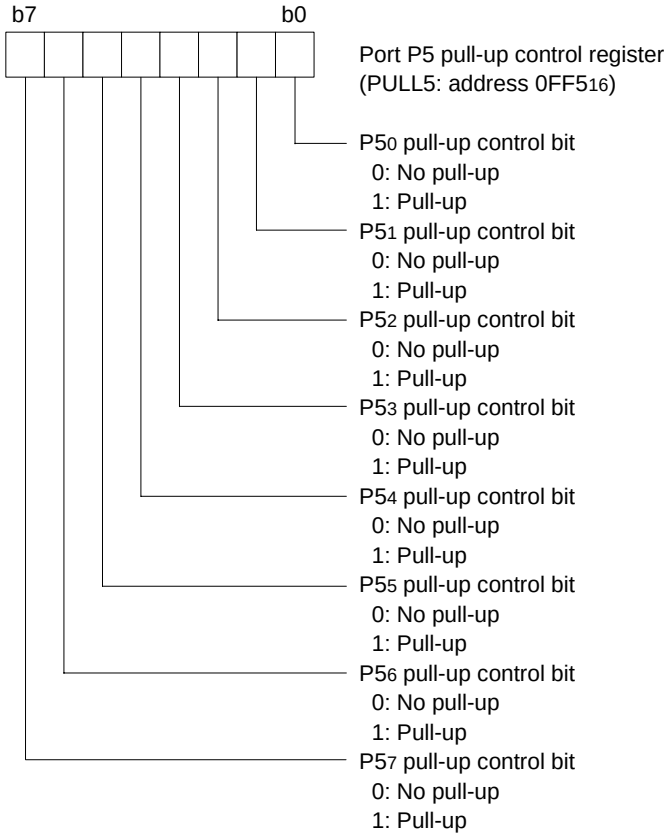


Note: Pull-up control is valid when the corresponding bit of the port direction register is "0" (input). When that bit is "1" (output), pull-up cannot be set to the port of which pull-up is selected.

Fig. 17 Structure of port pull-up control register (2)



Note: Pull-up control is valid when the corresponding bit of the port direction register is "0" (input). When that bit is "1" (output), pull-up cannot be set to the port of which pull-up is selected.



Note: Pull-up control is valid when the corresponding bit of the port direction register is "0" (input). When that bit is "1" (output), pull-up cannot be set to the port of which pull-up is selected.

Fig. 18 Structure of port pull-up control register (3)

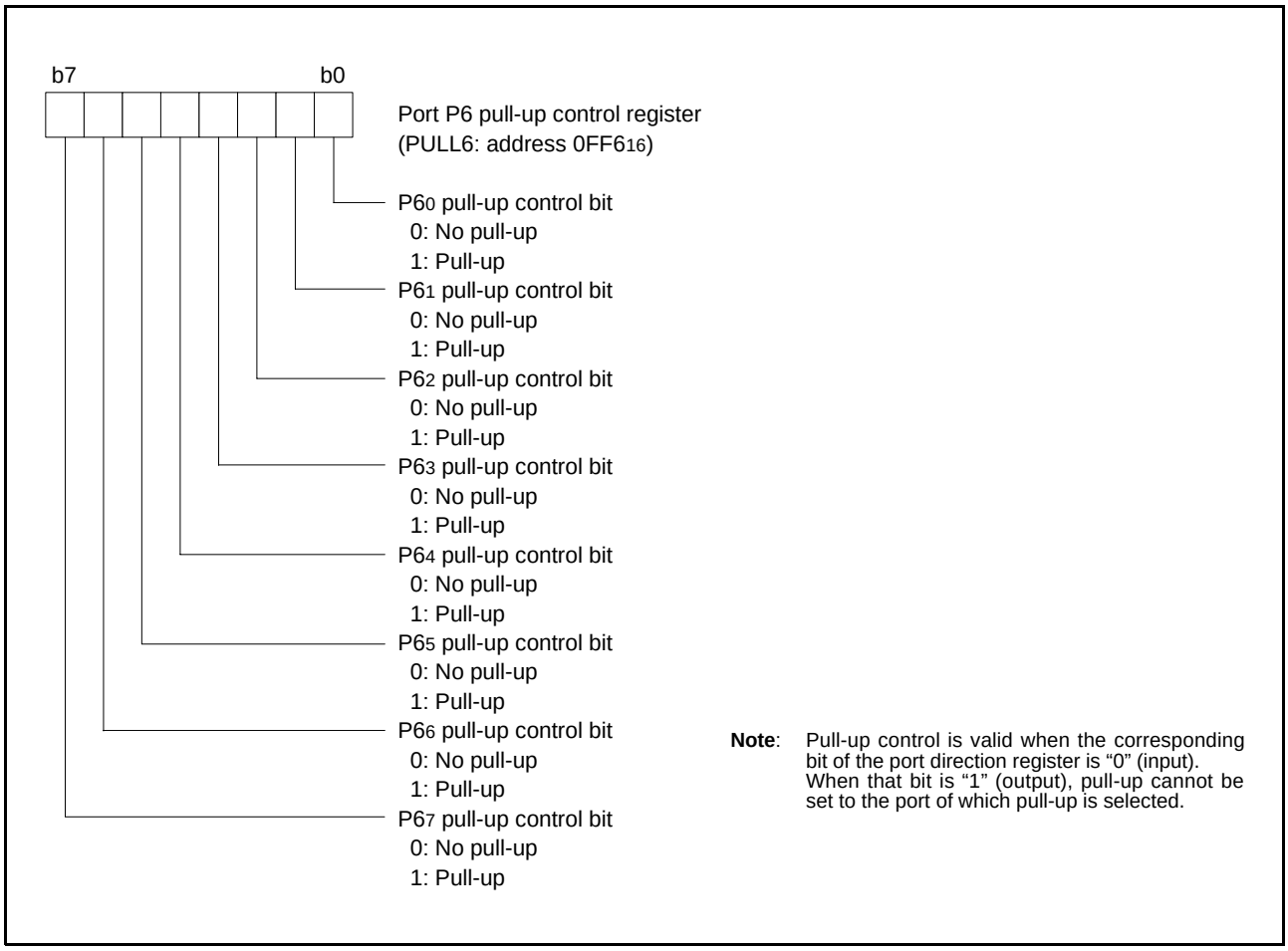


Fig. 19 Structure of port pull-up control register (4)

Termination of unused pins

• Termination of common pins

I/O ports: Select an input port or an output port and follow each processing method.
In addition, it is recommended that related registers be overwritten periodically to prevent malfunctions, etc.

Output ports: Open.

Input ports: If the input level become unstable, through current flow to an input circuit, and the power supply current may increase.
Especially, when expecting low consumption current (at STP or WIT instruction execution etc.), pull-up or pull-down input ports to prevent through current (built-in resistor can be used).
We recommend processing unused pins through a resistor which can secure $I_{OH(avg)}$ or $I_{OL(avg)}$.
Because, when an I/O port or a pin which have an output function is selected as an input port, it may operate as an output port by incorrect operation etc.

Table 7 Termination of unused pins

Pins	Termination
P0, P1, P2, P3, P4, P5, P6	• Set to the input mode and connect each to V_{CC} or V_{SS} through a resistor of 1 kΩ to 10 kΩ. • Set to the output mode and open at “L” or “H” output state.
V_{REF}	Connect to V_{CC} or V_{SS} (GND).
AV_{SS}	Connect to V_{CC} or V_{SS} (GND).
X_{OUT}	Open (only when using external clock)

INTERRUPTS

The 3804 group (Spec.L) interrupts are vector interrupts with a fixed priority scheme, and generated by 16 sources among 24 sources: 10 external, 13 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. Figure 20 shows an interrupt control diagram.

An interrupt requests 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(1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset(2)	1	FFFD16	FFFC16	At reset	Non-maskable
INT0	2	FFFB16	FFFA16	At detection of either rising or falling edge of INT0 input	External interrupt (active edge selectable)
Timer Z				At timer Z underflow	
INT1	3	FFF916	FFF816	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
Serial I/O1 reception	4	FFF716	FFF616	At completion of serial I/O1 data reception	Valid when serial I/O1 is selected
Serial I/O1 transmission	5	FFF516	FFF416	At completion of serial I/O1 transmission shift or when transmission buffer is empty	Valid when serial I/O1 is selected
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
Timer X	6	FFF316	FFF216	At timer X underflow	
Timer Y	7	FFF116	FFF016	At timer Y underflow	
Timer 1	8	FFEF16	FFEE16	At timer 1 underflow	STP release timer underflow
Timer 2	9	FFED16	FFEC16	At timer 2 underflow	
CNTR0	10	FFEB16	FFEA16	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
CNTR1	11	FFE916	FFE816	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
Serial I/O3 reception				At completion of serial I/O3 data reception	Valid when serial I/O3 is selected
Serial I/O2	12	FFE716	FFE616	At completion of serial I/O2 data transmission or reception	Valid when serial I/O2 is selected
Timer Z				At timer Z underflow	
INT2	13	FFE516	FFE416	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
I^2C				At completion of data transfer	
INT3	14	FFE316	FFE216	At detection of either rising or falling edge of INT3 input	External interrupt (active edge selectable)
INT4	15	FFE116	FFE016	At detection of either rising or falling edge of INT4 input	External interrupt (active edge selectable)
CNTR2				At detection of either rising or falling edge of CNTR2 input	External interrupt (active edge selectable)
A/D conversion	16	FFDF16	FFDE16	At completion of A/D conversion	
Serial I/O3 transmission				At completion of serial I/O3 transmission shift or when transmission buffer is empty	Valid when serial I/O3 is selected
BRK instruction	17	FFDD16	FFDC16	At BRK instruction execution	Non-maskable software interrupt

NOTES:
1. Vector addresses contain interrupt jump destination addresses.
2. Reset function in the same way as an interrupt with the highest priority.

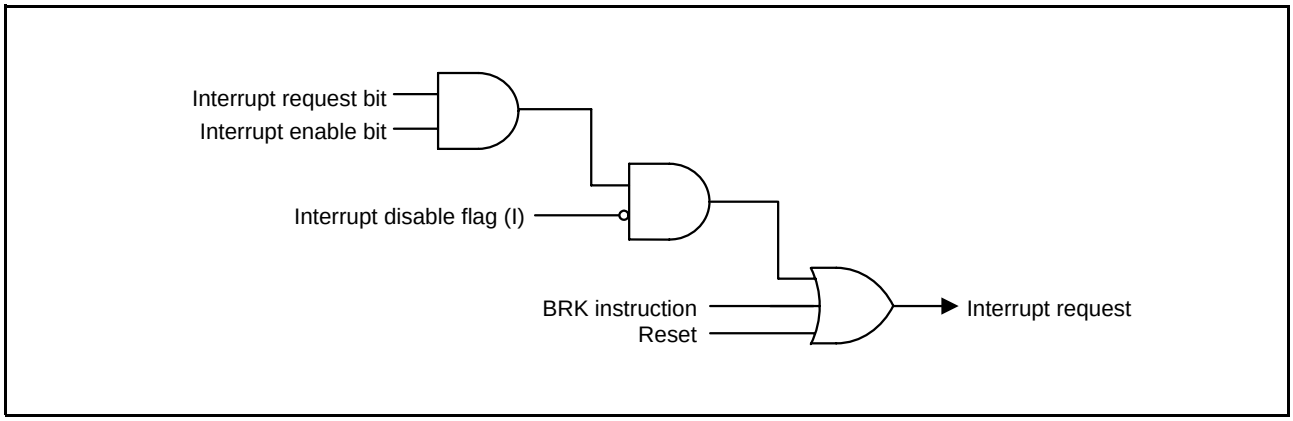


Fig. 20 Interrupt control diagram

• 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", acceptance of interrupt requests is enabled. This flag is set to "1" with the SET 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", 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 Source Selection

Any of the following combinations can be selected by the interrupt source selection register (0039_{16}).

1. INT0 or timer Z
2. Serial I/O1 transmission or SCL, SDA
3. $CNTR_0$ or SCL, SDA
4. $CNTR_1$ or Serial I/O3 reception
5. Serial I/O2 or timer Z
6. INT_2 or I^2C
7. INT_4 or $CNTR_2$
8. A/D conversion or serial I/O3 transmission

• External Interrupt Pin Selection

For external interrupts INT_0 and INT_4, the INT_0, INT_4 interrupt switch bit in the interrupt edge selection register (bit 6 of address $003A_{16}$) can be used to select INT_{00} and INT_{40} pin input or INT_{01} and INT_{41} pin input.

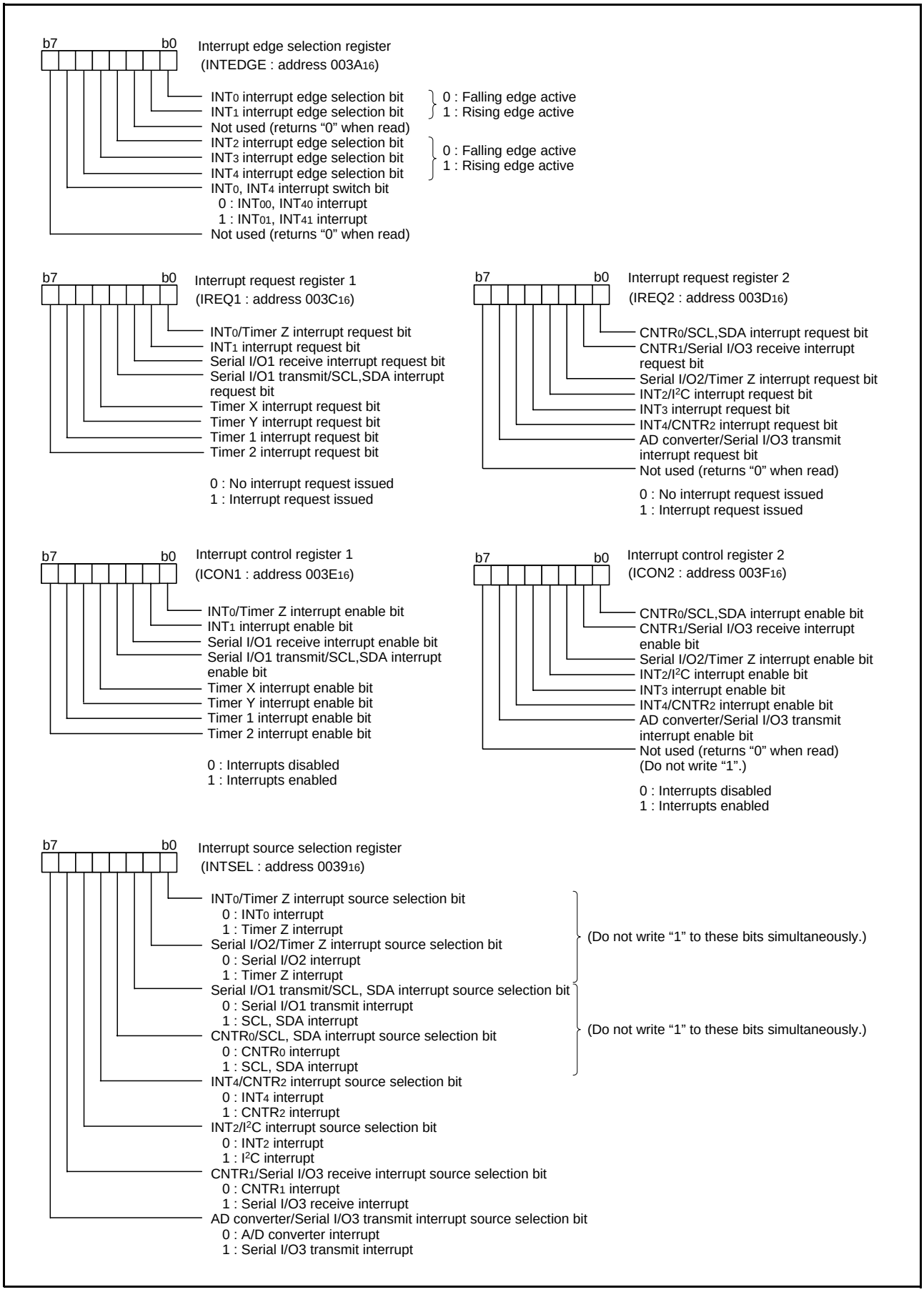


Fig. 21 Structure of interrupt-related registers

• 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 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 22 shows the time up to execution in the interrupt processing routine, and Figure 23 shows the interrupt sequence. Figure 24 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>

The interrupt request bit may be set to "1" in the following cases.

• When setting the external interrupt active edge
Related registers: Interrupt edge selection register (address $003A_{16}$)
Timer XY mode register (address 0023_{16})
Timer Z mode register (address $002A_{16}$)
I^2C START/STOP condition control register (address 0016_{16})

• When switching the interrupt sources of an interrupt vector address where two or more interrupt sources are assigned
Related registers: Interrupt source selection register (address 0039_{16})

If it is not necessary to generate an interrupt synchronized with these settings, take the following sequence.

(1) Set the corresponding enable bit to "0" (disabled).
(2) Set the interrupt edge selection bit (the active edge switch bit) or the interrupt source bit.
(3) Set the corresponding interrupt request bit to "0" after one or more instructions have been executed.
(4) Set the corresponding interrupt enable bit to "1" (enabled).

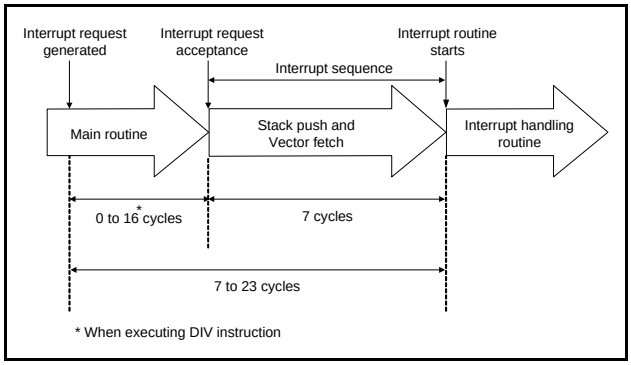


Fig. 22 Time up to execution in interrupt routine

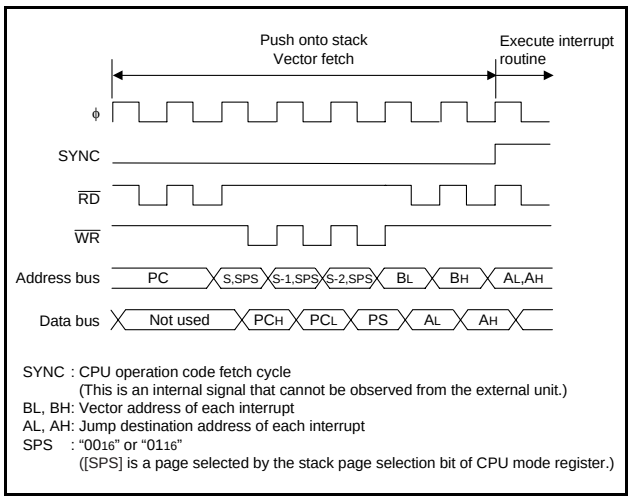
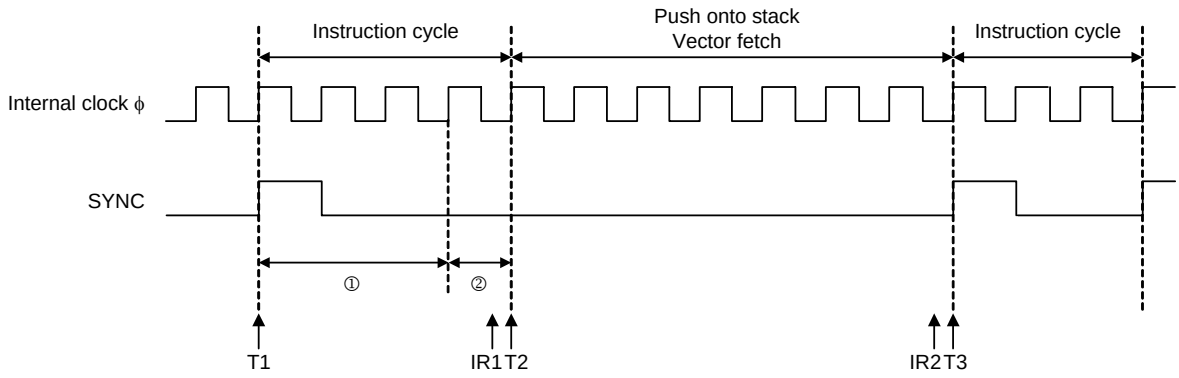


Fig. 23 Interrupt sequence



T1 T2 T3 : Interrupt acceptance timing points
IR1 IR2 : Timings points at which the interrupt request bit is set to “1”.

Note : Period 2 indicates the last ϕ cycle during one instruction cycle.

(1) The interrupt request bit for an interrupt request generated during period 1 is set to “1” at timing point IR1.
(2) The interrupt request bit for an interrupt request generated during period 2 is set to “1” at timing point IR1 or IR2. The timing point at which the bit is set to “1” varies depending on conditions. When two or more interrupt requests are generated during the period 2, each request bit may be set to “1” at timing point IR1 or IR2 separately.

Fig. 24 Timing of interrupt request generation, interrupt request bit, and interrupt acceptance

TIMERS

• 8-bit Timers

The 3804 group (Spec.L) has four 8-bit timers: timer 1, timer 2, timer X, and timer Y.
The timer 1 and timer 2 use one prescaler in common, and the timer X and timer Y use each prescaler. Those are 8-bit prescalers. Each of the timers and prescalers has a timer latch or a prescaler latch.
The division ratio of each timer or prescaler is given by 1/(n + 1), where n is the value in the corresponding timer or prescaler latch. All timers are down-counters. When the timer reaches "00$_{16}$", an underflow occurs at the next count pulse and the contents of the corresponding timer latch are reloaded into the timer and the count is continued. When the timer underflows, the interrupt request bit corresponding to that timer is set to "1".

• Timer divider

The divider count source is switched by the main clock division ratio selection bits of CPU mode register (bits 7 and 6 at address 003B$_{16}$). When these bits are "00" (high-speed mode) or "01" (middle-speed mode), X_{IN} is selected. When these bits are "10" (low-speed mode), X_{CIN} is selected.

• Prescaler 12

The prescaler 12 counts the output of the timer divider. The count source is selected by the timer 12, X count source selection register among 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 of f(X_{IN}) or f(X_{CIN}).

• Timer 1 and Timer 2

The timer 1 and timer 2 counts the output of prescaler 12 and periodically set the interrupt request bit.

• Prescaler X and prescaler Y

The prescaler X and prescaler Y count the output of the timer divider or f(X_{CIN}). The count source is selected by the timer 12, X count source selection register (address 000E$_{16}$) and the timer Y, Z count source selection register (address 000F$_{16}$) among 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, and 1/1024 of f(X_{IN}) or f(X_{CIN}); and f(X_{CIN}).

• Timer X and Timer Y

The timer X and timer Y can each select one of four operating modes by setting the timer XY mode register (address 0023$_{16}$).

(1) Timer mode

• Mode selection

This mode can be selected by setting "00" to the timer X operating mode bits (bits 1 and 0) and the timer Y operating mode bits (bits 5 and 4) of the timer XY mode register (address 0023$_{16}$).

• Explanation of operation

The timer count operation is started by setting "0" to the timer X count stop bit (bit 3) and the timer Y count stop bit (bit 7) of the timer XY mode register (address 0023$_{16}$).
When the timer reaches "00$_{16}$", an underflow occurs at the next count pulse and the contents of timer latch are reloaded into the timer and the count is continued.

(2) Pulse Output Mode

• Mode selection

This mode can be selected by setting "01" to the timer X operating mode bits (bits 1 and 0) and the timer Y operating mode bits (bits 5 and 4) of the timer XY mode register (address 0023$_{16}$).

• Explanation of operation

The operation is the same as the timer mode's. Moreover the pulse which is inverted each time the timer underflows is output from $CNTR_0$/$CNTR_1$ pin. Regardless of the timer counting or not the output of $CNTR_0$/$CNTR_1$ pin is initialized to the level of specified by their active edge switch bits when writing to the timer. When the $CNTR_0$ active edge switch bit (bit 2) and the $CNTR_1$ active edge switch bit (bit 6) of the timer XY mode register (address 0023$_{16}$) is "0", the output starts with "H" level. When it is "1", the output starts with "L" level.
Switching the $CNTR_0$ or $CNTR_1$ active edge switch bit will reverse the output level of the corresponding $CNTR_0$ or $CNTR_1$ pin.

• Precautions

Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to output in this mode.

(3) Event Counter Mode

• Mode selection

This mode can be selected by setting "10" to the timer X operating mode bits (bits 1 and 0) and the timer Y operating mode bits (bits 5 and 4) of the timer XY mode register (address 0023$_{16}$).

• Explanation of operation

The operation is the same as the timer mode's except that the timer counts signals input from the $CNTR_0$ or $CNTR_1$ pin. The valid edge for the count operation depends on the $CNTR_0$ active edge switch bit (bit 2) or the $CNTR_1$ active edge switch bit (bit 6) of the timer XY mode register (address 0023$_{16}$). When it is "0", the rising edge is valid. When it is "1", the falling edge is valid.

• Precautions

Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to input in this mode.

(4) Pulse Width Measurement Mode

• Mode selection

This mode can be selected by setting “11” to the timer X operating mode bits (bits 1 and 0) and the timer Y operating mode bits (bits 5 and 4) of the timer XY mode register (address 0023_{16}).

• Explanation of operation

When the $CNTR_0$ active edge switch bit (bit 2) or the $CNTR_1$ active edge switch bit (bit 6) of the timer XY mode register (address 0023_{16}) is “1”, the timer counts during the term of one falling edge of $CNTR_0$/$CNTR_1$ pin input until the next rising edge of input (“L” term). When it is “0”, the timer counts during the term of one rising edge input until the next falling edge input (“H” term).

• Precautions

Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to input in this mode.

The count operation can be stopped by setting “1” to the timer X count stop bit (bit 3) and the timer Y count stop bit (bit 7) of the timer XY mode register (address 0023_{16}). The interrupt request bit is set to “1” each time the timer underflows.

• Precautions when switching count source

When switching the count source by the timer 12, X and Y count source selection bits, the value of timer count is altered in inconsiderable amount owing to generating of thin pulses on the count input signals.

Therefore, select the timer count source before setting the value to the prescaler and the timer.

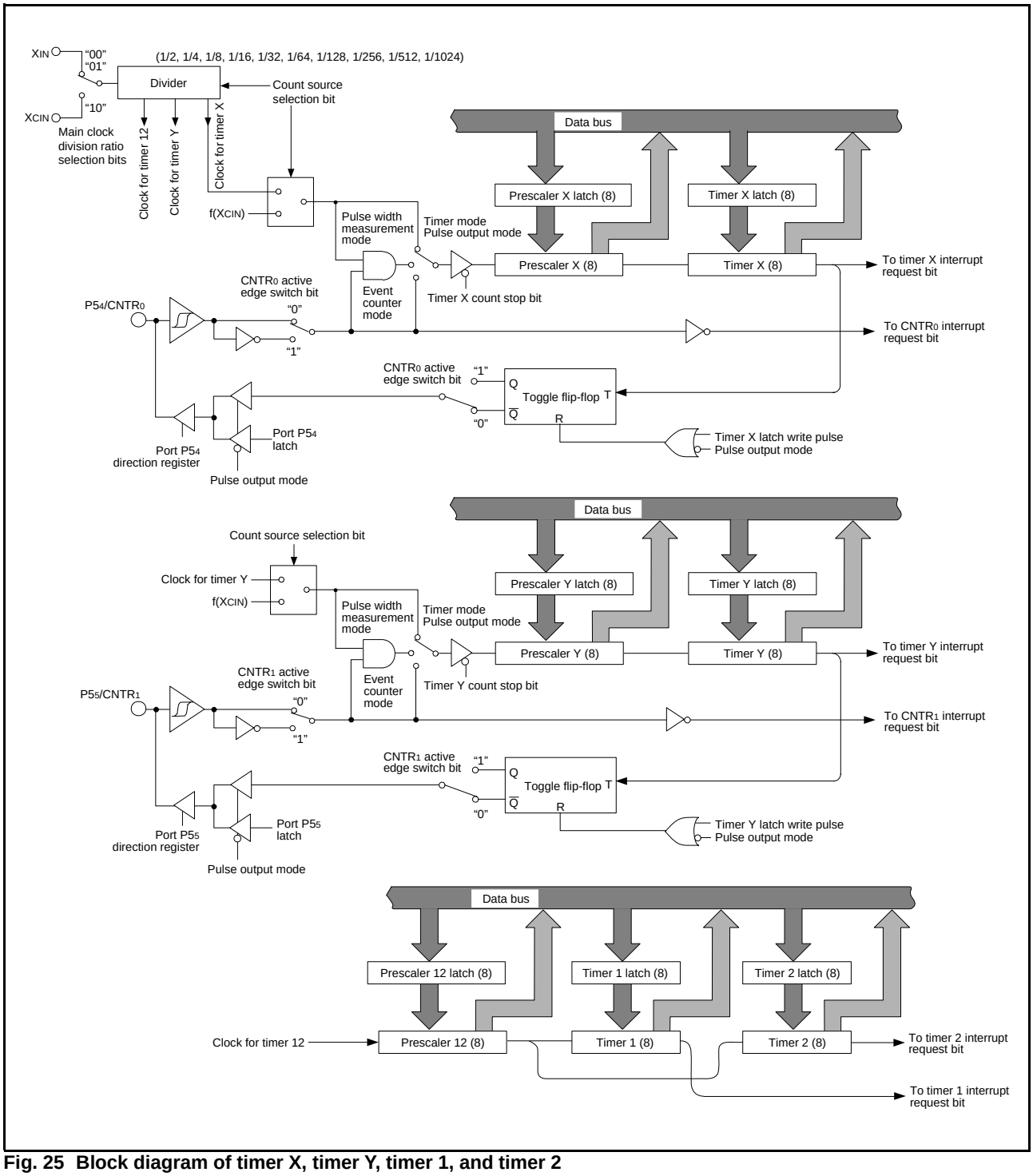


Fig. 25 Block diagram of timer X, timer Y, timer 1, and timer 2

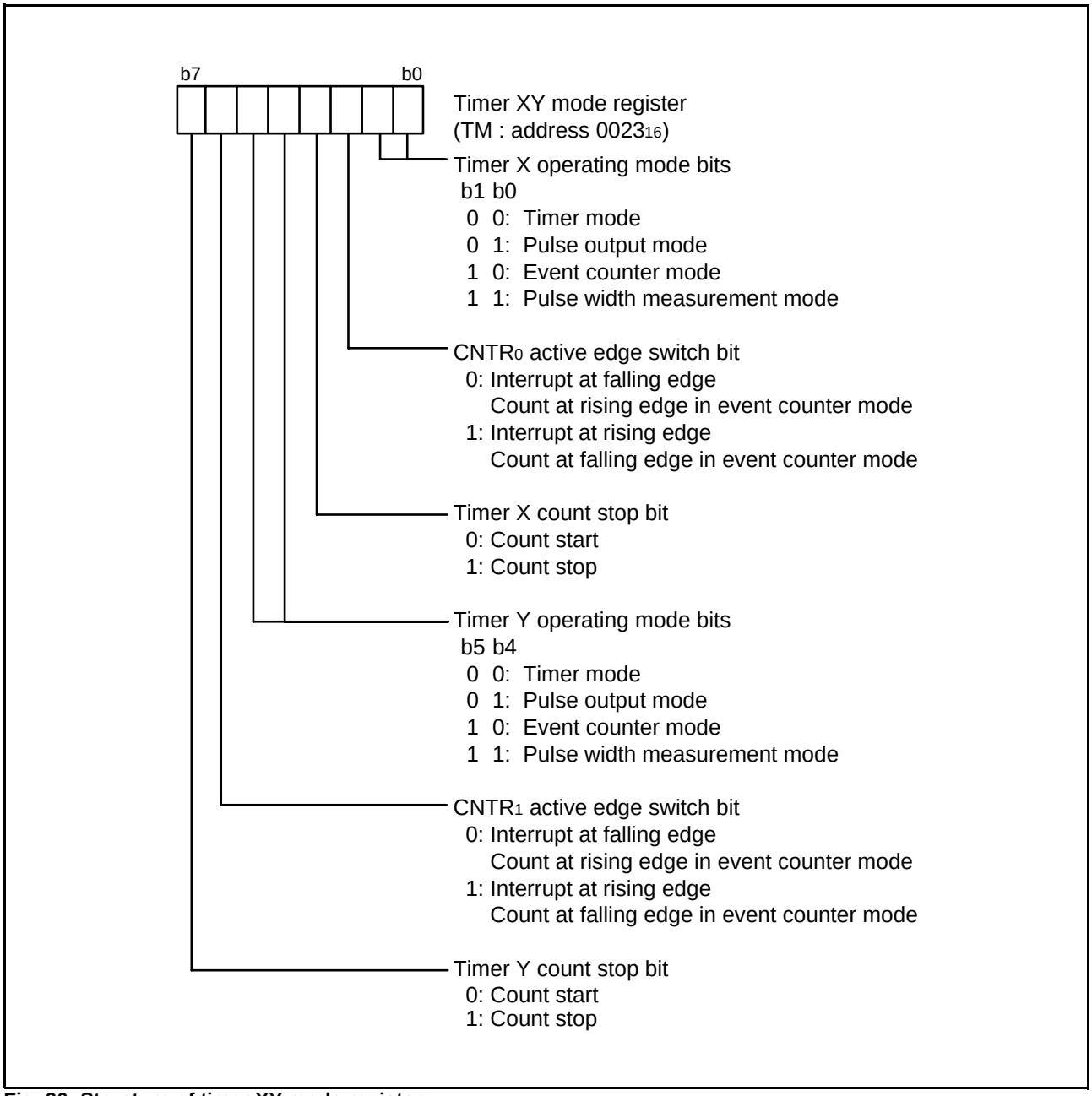


Fig. 26 Structure of timer XY mode register

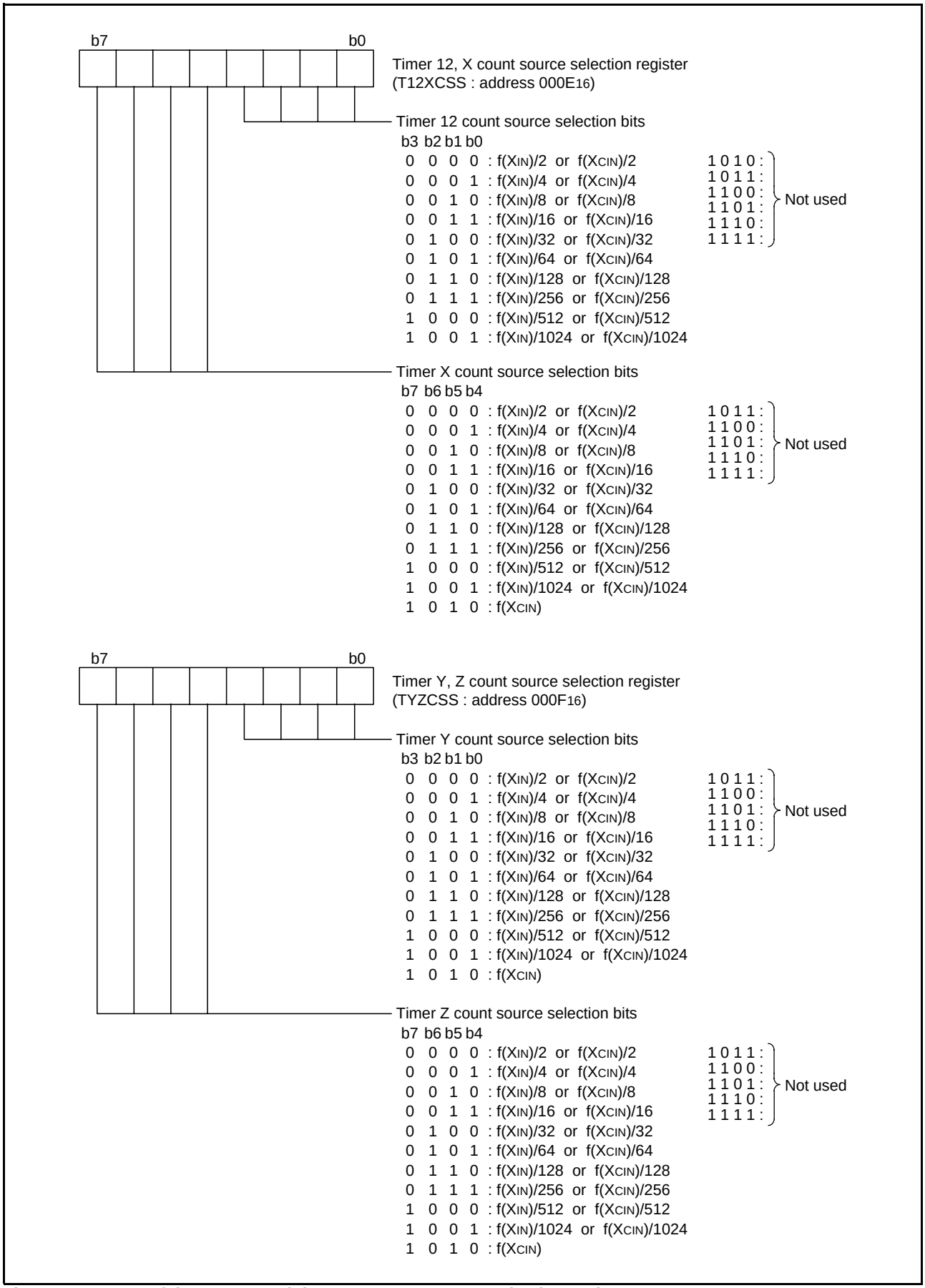


Fig. 27 Structure of timer 12, X and timer Y, Z count source selection registers

• 16-bit Timer

The timer Z is a 16-bit timer. When the timer reaches "0000$_{16}$", an underflow occurs at the next count pulse and the corresponding timer latch is reloaded into the timer and the count is continued. When the timer underflows, the interrupt request bit corresponding to the timer Z is set to "1".

When reading/writing to the timer Z, perform reading/writing to both the high-order byte and the low-order byte. When reading the timer Z, read from the high-order byte first, followed by the low-order byte. Do not perform the writing to the timer Z between read operation of the high-order byte and read operation of the low-order byte. When writing to the timer Z, write to the low-order byte first, followed by the high-order byte. Do not perform the reading to the timer Z between write operation of the low-order byte and write operation of the high-order byte.

The timer Z can select the count source by the timer Z count source selection bits of timer Y, Z count source selection register (bits 7 to 4 at address 000F$_{16}$).

Timer Z can select one of seven operating modes by setting the timer Z mode register (address 002A$_{16}$).

(1) Timer mode

• Mode selection

This mode can be selected by setting "000" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A$_{16}$).

• Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(X$_{IN}$); or f(X$_{CIN}$) can be selected as the count source.

In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(X$_{CIN}$); or f(X$_{CIN}$) can be selected as the count source.

• Interrupt

When an underflow occurs, the INT$_0$/timer Z interrupt request bit (bit 0) of the interrupt request register 1 (address 003C$_{16}$) is set to "1".

• Explanation of operation

During timer stop, usually write data to a latch and a timer at the same time to set the timer value.

The timer count operation is started by setting "0" to the timer Z count stop bit (bit 6) of the timer Z mode register (address 002A$_{16}$).

When the timer reaches "0000$_{16}$", an underflow occurs at the next count pulse and the contents of timer latch are reloaded into the timer and the count is continued.

When writing data to the timer during operation, the data is written only into the latch. Then the new latch value is reloaded into the timer at the next underflow.

(2) Event counter mode

• Mode selection

This mode can be selected by setting "000" to the timer Z operating mode bits (bits 2 to 0) and setting "1" to the timer/event counter mode switch bit (bit 7) of the timer Z mode register (address 002A$_{16}$).

The valid edge for the count operation depends on the CNTR$_2$ active edge switch bit (bit 5) of the timer Z mode register (address 002A$_{16}$). When it is "0", the rising edge is valid. When it is "1", the falling edge is valid.

• Interrupt

The interrupt at an underflow is the same as the timer mode's.

• Explanation of operation

The operation is the same as the timer mode's.

Set the double-function port of CNTR$_2$ pin and port P4$_7$ to input in this mode.

Figure 30 shows the timing chart of the timer/event counter mode.

(3) Pulse output mode

• Mode selection

This mode can be selected by setting "001" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A$_{16}$).

• Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(X$_{IN}$); or f(X$_{CIN}$) can be selected as the count source.

In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(X$_{CIN}$); or f(X$_{CIN}$) can be selected as the count source.

• Interrupt

The interrupt at an underflow is the same as the timer mode's.

• Explanation of operation

The operation is the same as the timer mode's. Moreover the pulse which is inverted each time the timer underflows is output from CNTR$_2$ pin. When the CNTR$_2$ active edge switch bit (bit 5) of the timer Z mode register (address 002A$_{16}$) is "0", the output starts with "H" level. When it is "1", the output starts with "L" level.

• Precautions

The double-function port of CNTR$_2$ pin and port P4$_7$ is automatically set to the timer pulse output port in this mode.

The output from CNTR$_2$ pin is initialized to the level depending on CNTR$_2$ active edge switch bit by writing to the timer.

When the value of the CNTR$_2$ active edge switch bit is changed, the output level of CNTR$_2$ pin is inverted.

Figure 31 shows the timing chart of the pulse output mode.

(4) Pulse period measurement mode

• Mode selection

This mode can be selected by setting "010" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A16).

• Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XIN); or f(XCIN) can be selected as the count source.
In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XCIN); or f(XCIN) can be selected as the count source.

• Interrupt

The interrupt at an underflow is the same as the timer mode's. When the pulse period measurement is completed, the INT4/CNTR2 interrupt request bit (bit 5) of the interrupt request register 2 (address 003D16) is set to "1".

• Explanation of operation

The cycle of the pulse which is input from the CNTR2 pin is measured. When the CNTR2 active edge switch bit (bit 5) of the timer Z mode register (address 002A16) is "0", the timer counts during the term from one falling edge of CNTR2 pin input to the next falling edge. When it is "1", the timer counts during the term from one rising edge input to the next rising edge input.
When the valid edge of measurement completion/start is detected, the 1's complement of the timer value is written to the timer latch and "FFFF16" is set to the timer.
Furthermore when the timer underflows, the timer Z interrupt request occurs and "FFFF16" is set to the timer. When reading the timer Z, the value of the timer latch (measured value) is read. The measured value is retained until the next measurement completion.

• Precautions

Set the double-function port of CNTR2 pin and port P47 to input in this mode.
A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse period).
Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.
"FFFF16" is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected. Consequently, the timer value at start of pulse period measurement depends on the timer value just before measurement start.
Figure 32 shows the timing chart of the pulse period measurement mode.

(5) Pulse width measurement mode

• Mode selection

This mode can be selected by setting "011" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A16).

• Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XIN); or f(XCIN) can be selected as the count source.
In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XCIN); or f(XCIN) can be selected as the count source.

• Interrupt

The interrupt at an underflow is the same as the timer mode's. When the pulse widths measurement is completed, the INT4/CNTR2 interrupt request bit (bit 5) of the interrupt request register 2 (address 003D16) is set to "1".

• Explanation of operation

The pulse width which is input from the CNTR2 pin is measured. When the CNTR2 active edge switch bit (bit 5) of the timer Z mode register (address 002A16) is "0", the timer counts during the term from one rising edge input to the next falling edge input ("H" term). When it is "1", the timer counts during the term from one falling edge of CNTR2 pin input to the next rising edge of input ("L" term).
When the valid edge of measurement completion is detected, the 1's complement of the timer value is written to the timer latch.
When the valid edge of measurement completion/start is detected, "FFFF16" is set to the timer.
When the timer Z underflows, the timer Z interrupt occurs and "FFFF16" is set to the timer Z. When reading the timer Z, the value of the timer latch (measured value) is read. The measured value is retained until the next measurement completion.

• Precautions

Set the double-function port of CNTR2 pin and port P47 to input in this mode.
A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse widths).
Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.
"FFFF16" is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected. Consequently, the timer value at start of pulse width measurement depends on the timer value just before measurement start.
Figure 33 shows the timing chart of the pulse width measurement mode.

(6) Programmable waveform generating mode

• Mode selection

This mode can be selected by setting "100" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A16).

• Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XIN); or f(XCIN) can be selected as the count source.
In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XCIN); or f(XCIN) can be selected as the count source.

• Interrupt

The interrupt at an underflow is the same as the timer mode's.

• Explanation of operation

The operation is the same as the timer mode's. Moreover the timer outputs the data set in the output level latch (bit 4) of the timer Z mode register (address 002A16) from the CNTR2 pin each time the timer underflows.
Changing the value of the output level latch and the timer latch after an underflow makes it possible to output an optional waveform from the CNTR2 pin.

• Precautions

The double-function port of CNTR2 pin and port P47 is automatically set to the programmable waveform generating port in this mode.
Figure 34 shows the timing chart of the programmable waveform generating mode.

(7) Programmable one-shot generating mode

• Mode selection

This mode can be selected by setting "101" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A16).

• Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XIN); or f(XCIN) can be selected as the count source.

• Interrupt

The interrupt at an underflow is the same as the timer mode's.
The trigger to generate one-shot pulse can be selected by the INT1 active edge selection bit (bit 1) of the interrupt edge selection register (address 003A16). When it is "0", the falling edge active is selected; when it is "1", the rising edge active is selected.
When the valid edge of the INT1 pin is detected, the INT1 interrupt request bit (bit 1) of the interrupt request register 1 (address 003C16) is set to "1".

• Explanation of operation

1. "H" one-shot pulse; Bit 5 of timer Z mode register = "0"
   The output level of the CNTR2 pin is initialized to "L" at mode selection. When trigger generation (input signal to INT1 pin) is detected, "H" is output from the CNTR2 pin. When an underflow occurs, "L" is output. The "H" one-shot pulse width is set by the setting value to the timer Z register low-order and high-order. When trigger generating is detected during timer count stop, although "H" is output from the CNTR2 pin, "H" output state continues because an underflow does not occur.
2. "L" one-shot pulse; Bit 5 of timer Z mode register = "1"
   The output level of the CNTR2 pin is initialized to "H" at mode selection. When trigger generation (input signal to INT1 pin) is detected, "L" is output from the CNTR2 pin. When an underflow occurs, "H" is output. The "L" one-shot pulse width is set by the setting value to the timer Z low-order and high-order. When trigger generating is detected during timer count stop, although "L" is output from the CNTR2 pin, "L" output state continues because an underflow does not occur.

• Precautions

Set the double-function port of INT1 pin and port P42 to input in this mode.
The double-function port of CNTR2 pin and port P47 is automatically set to the programmable one-shot generating port in this mode.
This mode cannot be used in low-speed mode.
If the value of the CNTR2 active edge switch bit is changed during one-shot generating enabled or generating one-shot pulse, then the output level from CNTR2 pin changes.
Figure 35 shows the timing chart of the programmable one-shot generating mode.

<Notes regarding all modes>

• Timer Z write control

Which write control can be selected by the timer Z write control bit (bit 3) of the timer Z mode register (address 002A16), writing data to both the latch and the timer at the same time or writing data only to the latch.

When the operation "writing data only to the latch" is selected, the value is set to the timer latch by writing data to the address of timer Z and the timer is updated at next underflow. After reset release, the operation "writing data to both the latch and the timer at the same time" is selected, and the value is set to both the latch and the timer at the same time by writing data to the address of timer Z.

In the case of writing data only to the latch, if writing data to the latch and an underflow are performed almost at the same time, the timer value may become undefined.

• Timer Z read control

A read-out of timer value is impossible in pulse period measurement mode and pulse width measurement mode. In the other modes, a read-out of timer value is possible regardless of count operating or stopped.

However, a read-out of timer latch value is impossible.

• Switch of interrupt active edge of CNTR2 and INT1

Each interrupt active edge depends on setting of the CNTR2 active edge switch bit and the INT1 active edge selection bit.

• Switch of count source

When switching the count source by the timer Z count source selection bits, the value of timer count is altered in inconsiderable amount owing to generating of thin pulses on the count input signals.

Therefore, select the timer count source before setting the value to the prescaler and the timer.

• Usage of CNTR2 pin as normal I/O port P47

To use the CNTR2 pin as normal I/O port P47, set timer Z operating mode bits (b2, b1, b0) of timer Z mode register (address 002A16) to "000".

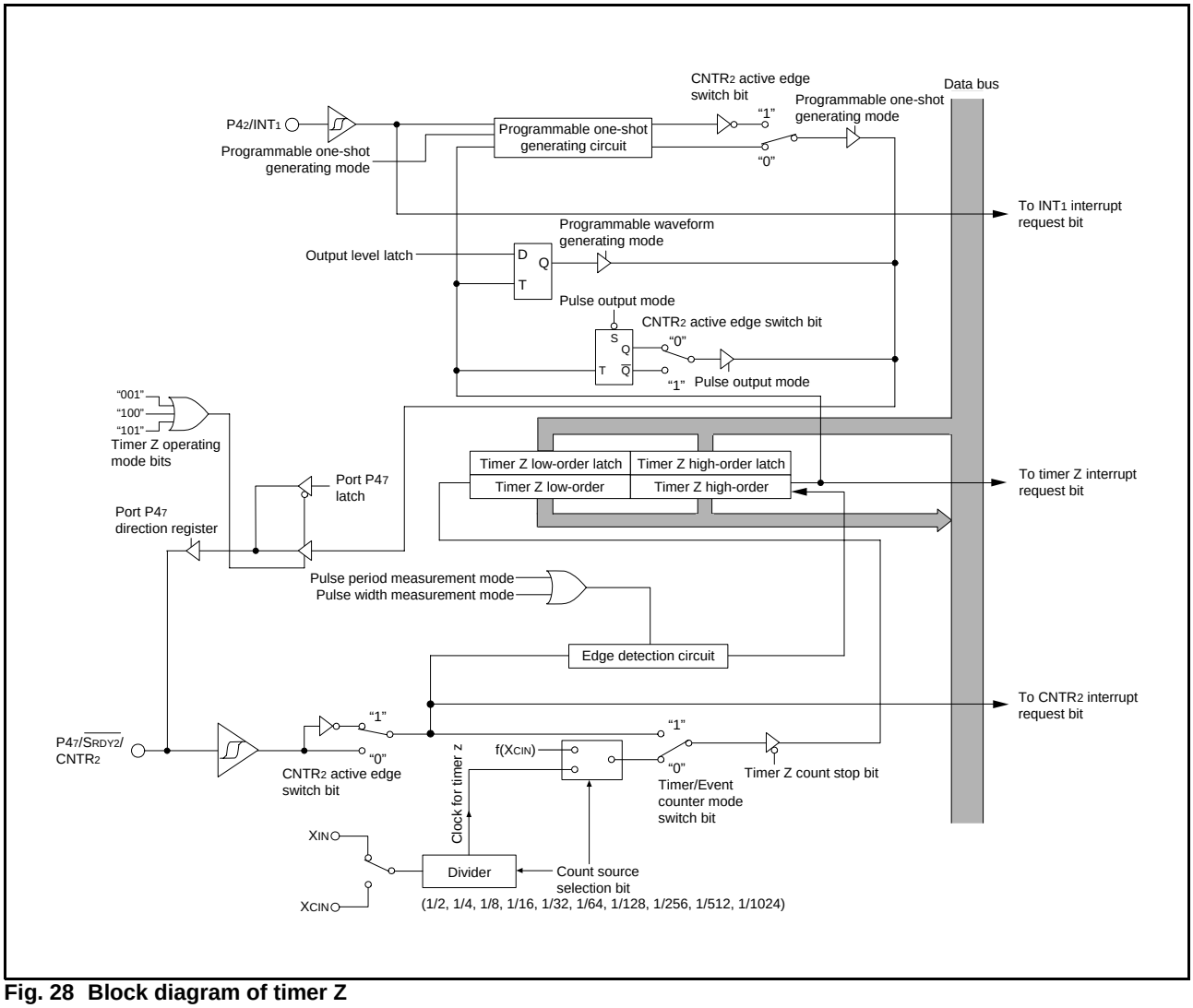
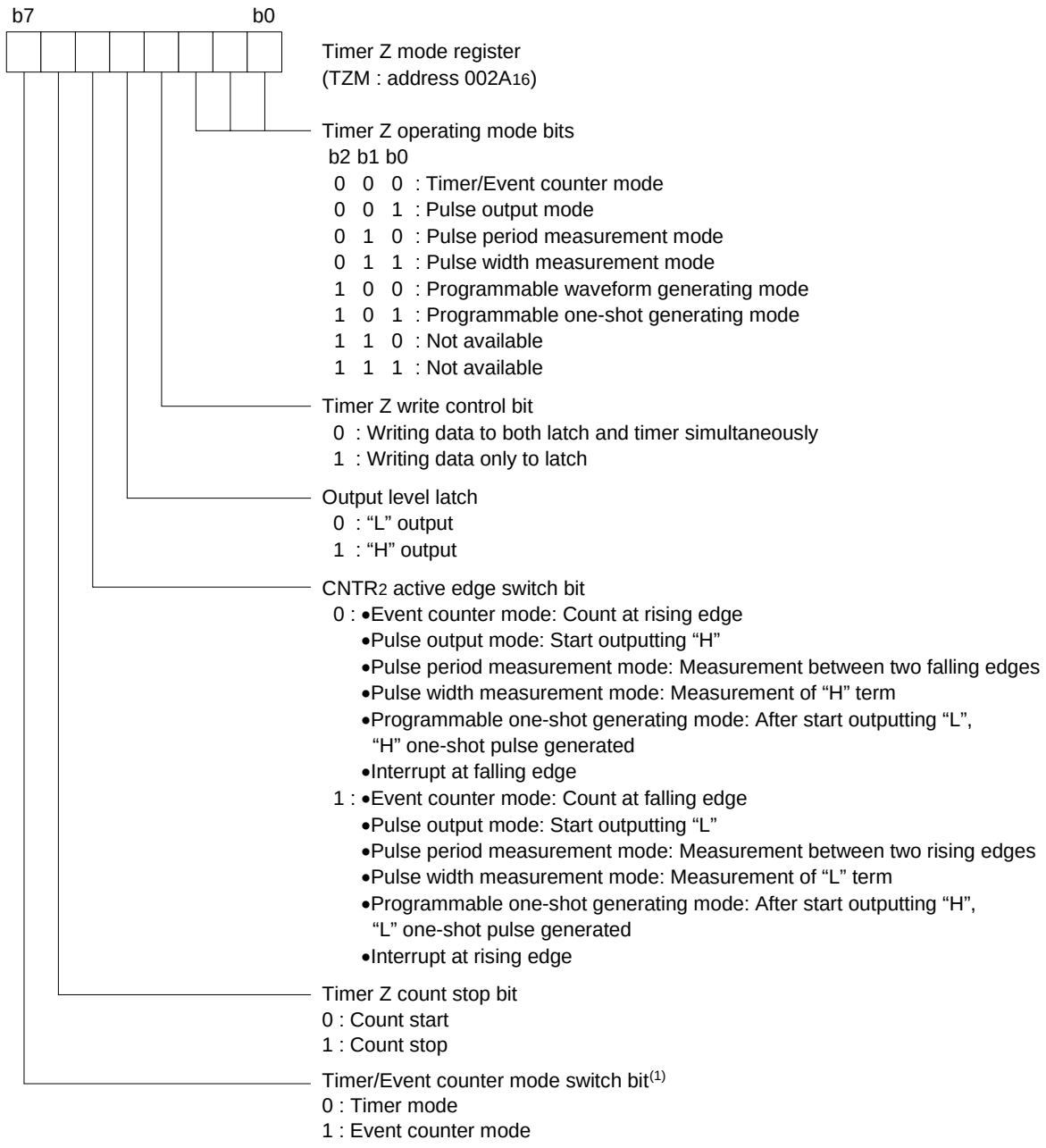
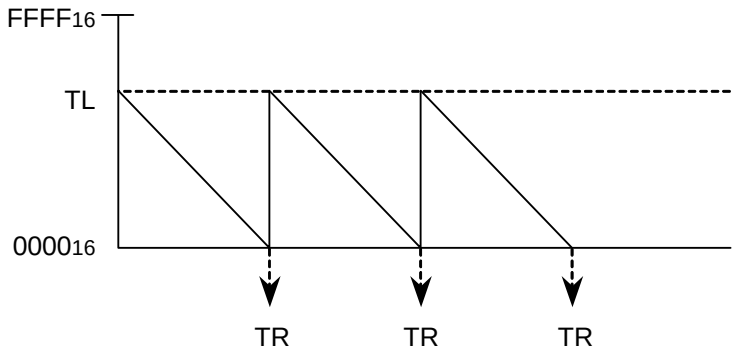


Fig. 28 Block diagram of timer Z



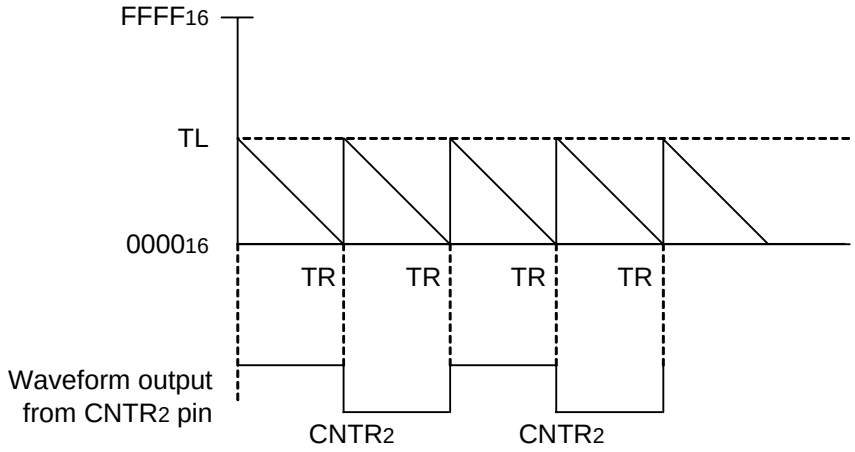
Note 1: When selecting the modes except the timer/event counter mode, set "0" to this bit.

Fig. 29 Structure of timer Z mode register



TL : Value set to timer latch
TR : Timer interrupt request

Fig. 30 Timing chart of timer/event counter mode



TL : Value set to timer latch
TR : Timer interrupt request
CNTR2 : CNTR2 interrupt request
(CNTR2 active edge switch bit = “0”; Falling edge active)

Fig. 31 Timing chart of pulse output mode

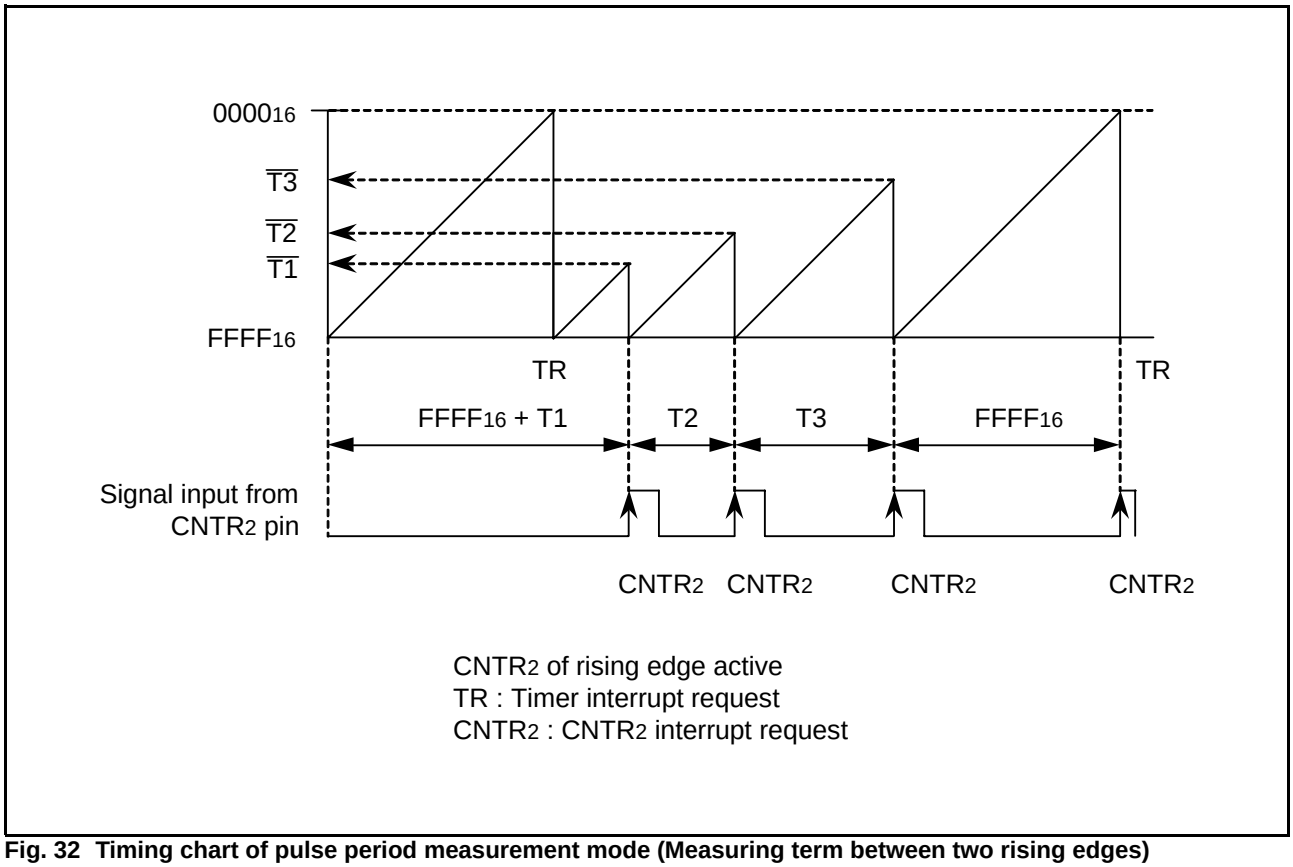


Fig. 32 Timing chart of pulse period measurement mode (Measuring term between two rising edges)

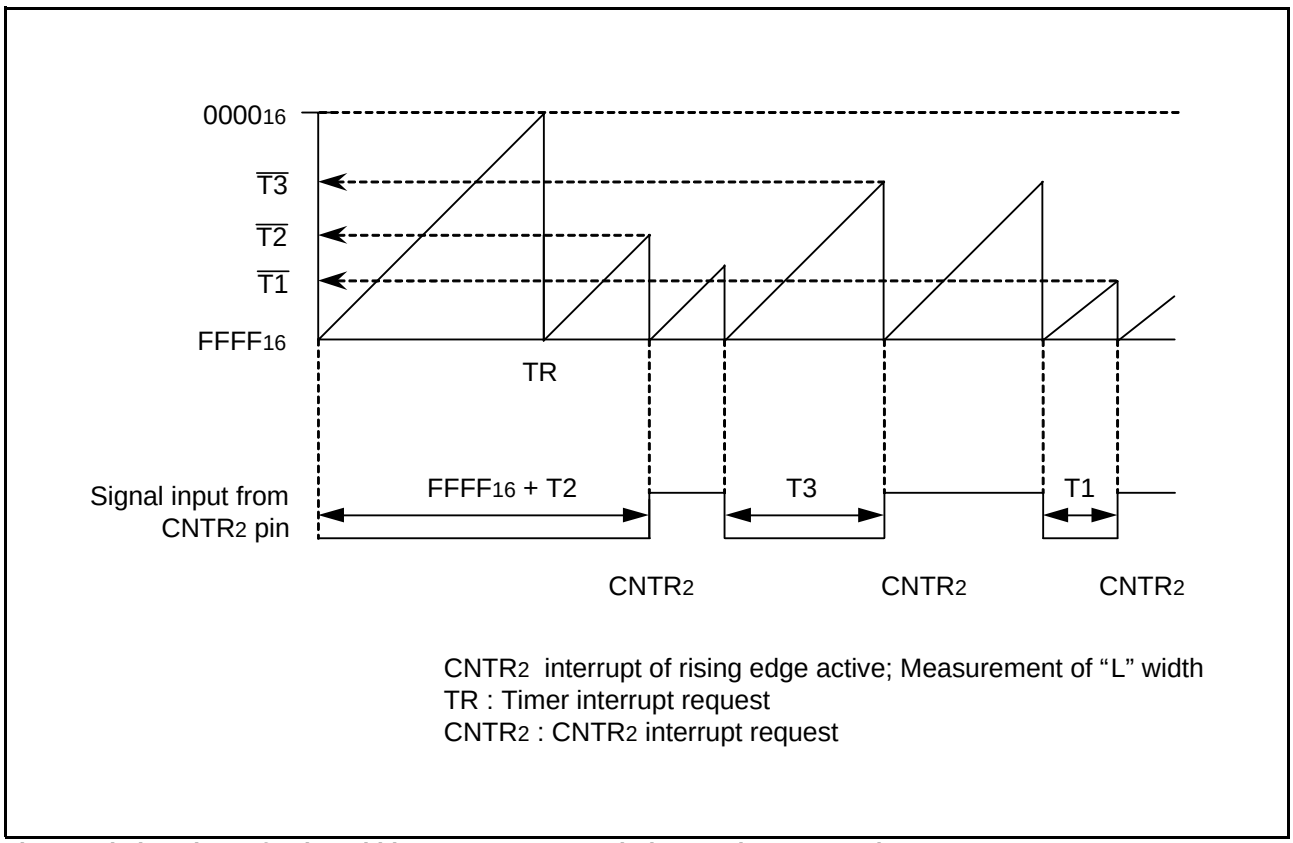
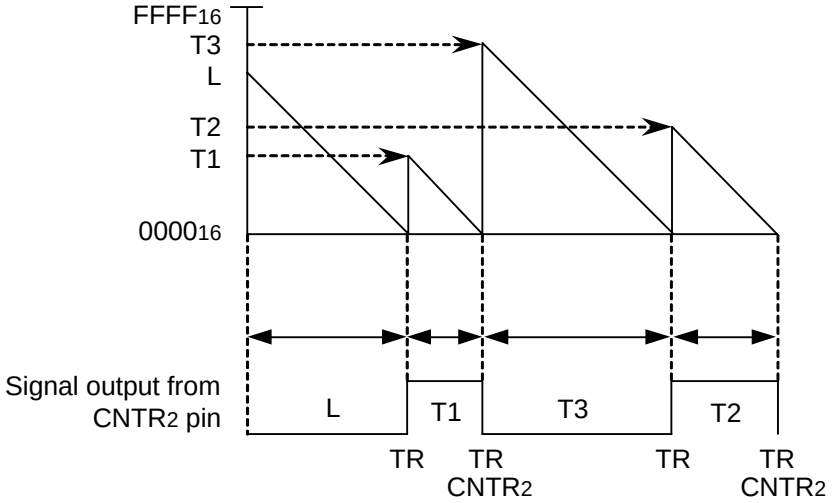
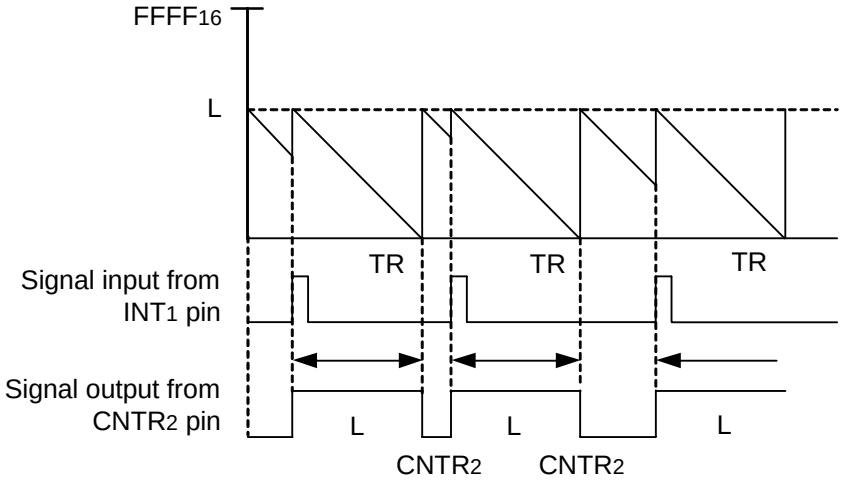


Fig. 33 Timing chart of pulse width measurement mode (Measuring “L” term)



L : Timer initial value
TR : Timer interrupt request
CNTR2 : CNTR2 interrupt request
(CNTR2 active edge switch bit = “0”; Falling edge active)

Fig. 34 Timing chart of programmable waveform generating mode



L : One-shot pulse width
TR : Timer interrupt request
CNTR2 : CNTR2 interrupt request
(CNTR2 active edge switch bit = “0”; Falling edge active)

Fig. 35 Timing chart of programmable one-shot generating mode (“H” one-shot pulse generating)

SERIAL INTERFACE

• Serial I/O1

Serial I/O1 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/O Mode

Clock synchronous serial I/O1 mode can be selected by setting the serial I/O1 mode selection bit of the serial I/O1 control register (bit 6 of address $001A_{16}$) to "1".
For clock synchronous serial I/O, 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 transmit/receive buffer register.

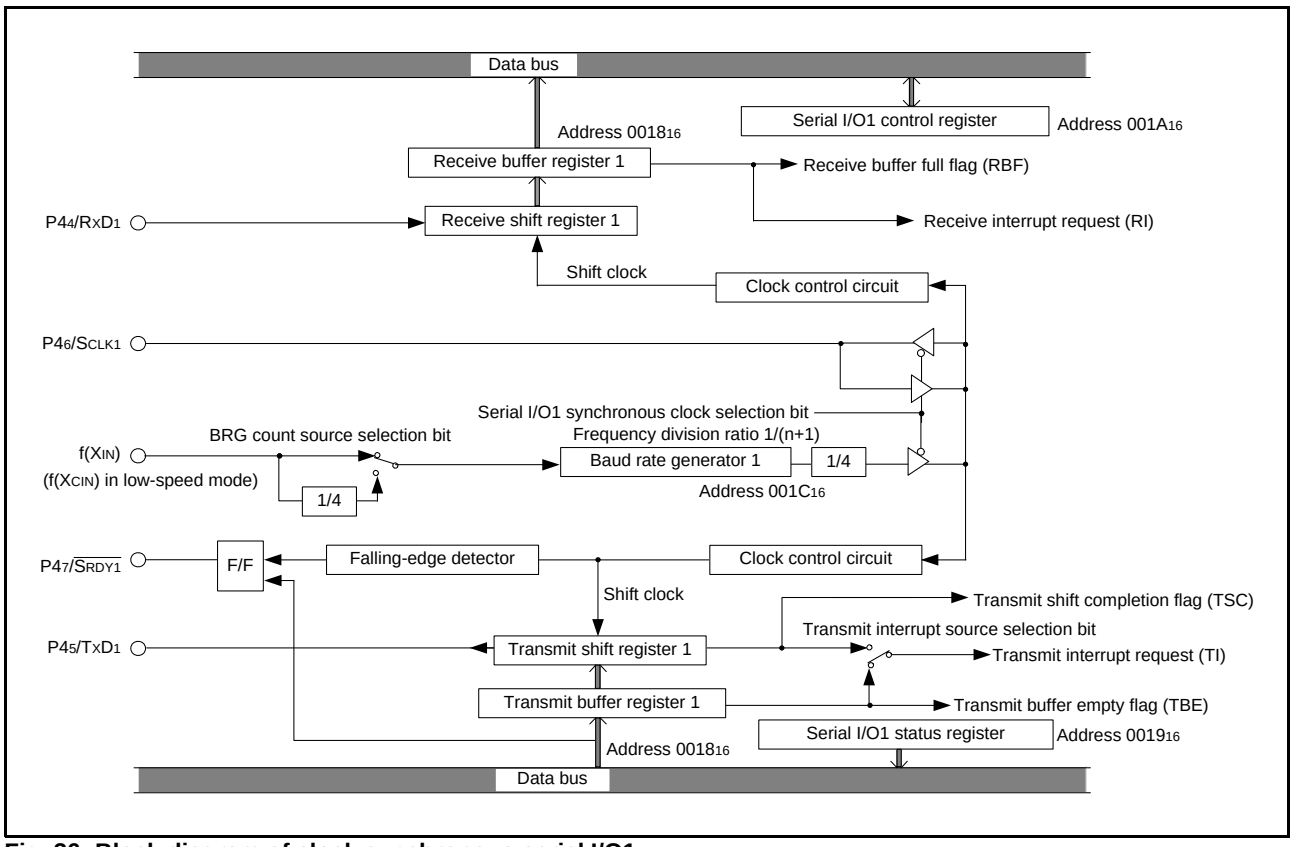
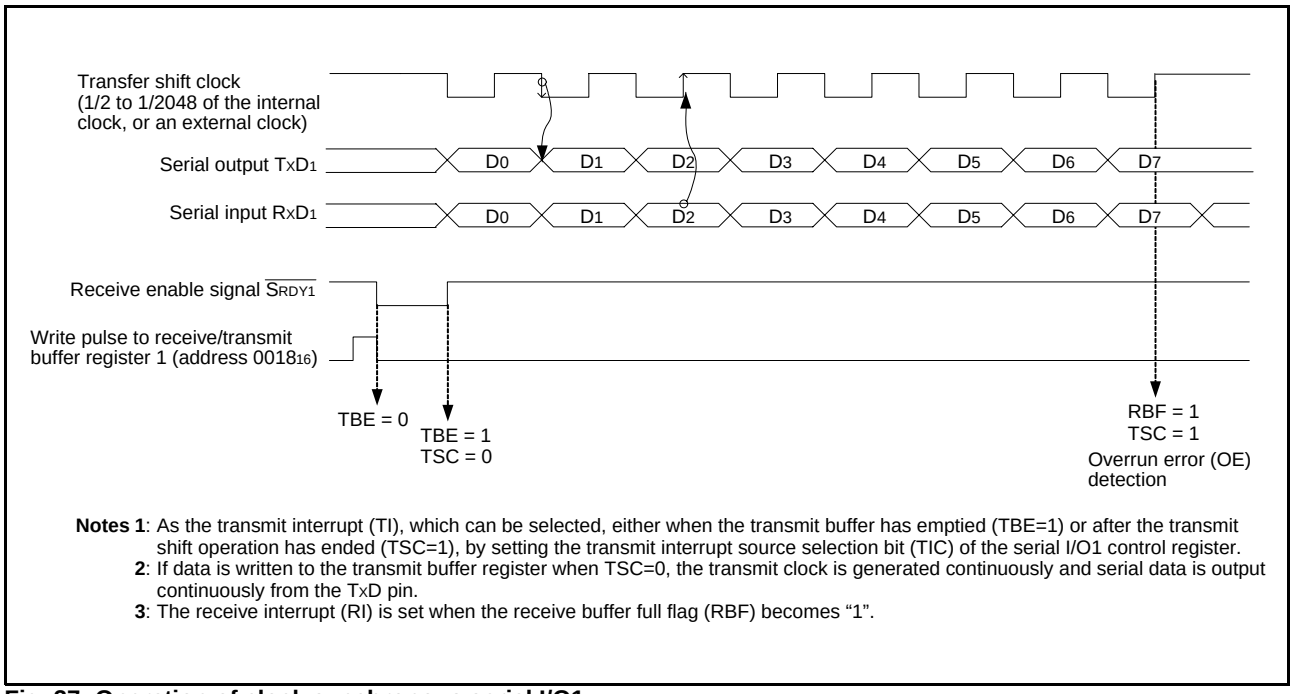


Fig. 36 Block diagram of clock synchronous serial I/O1



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/O1 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 pin.
3: The receive interrupt (RI) is set when the receive buffer full flag (RBF) becomes "1".

Fig. 37 Operation of clock synchronous serial I/O1

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

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O1 mode selection bit (b6) 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 a 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.

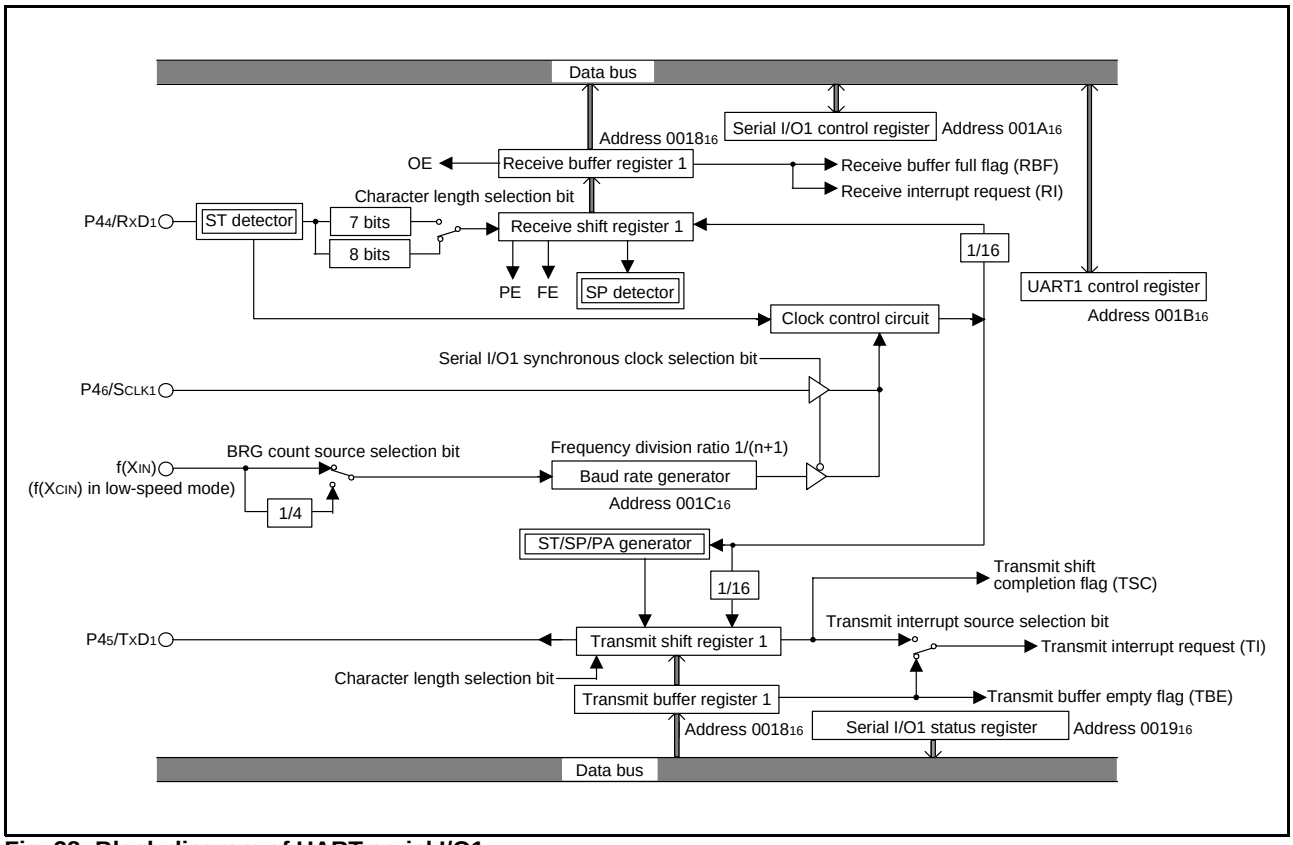
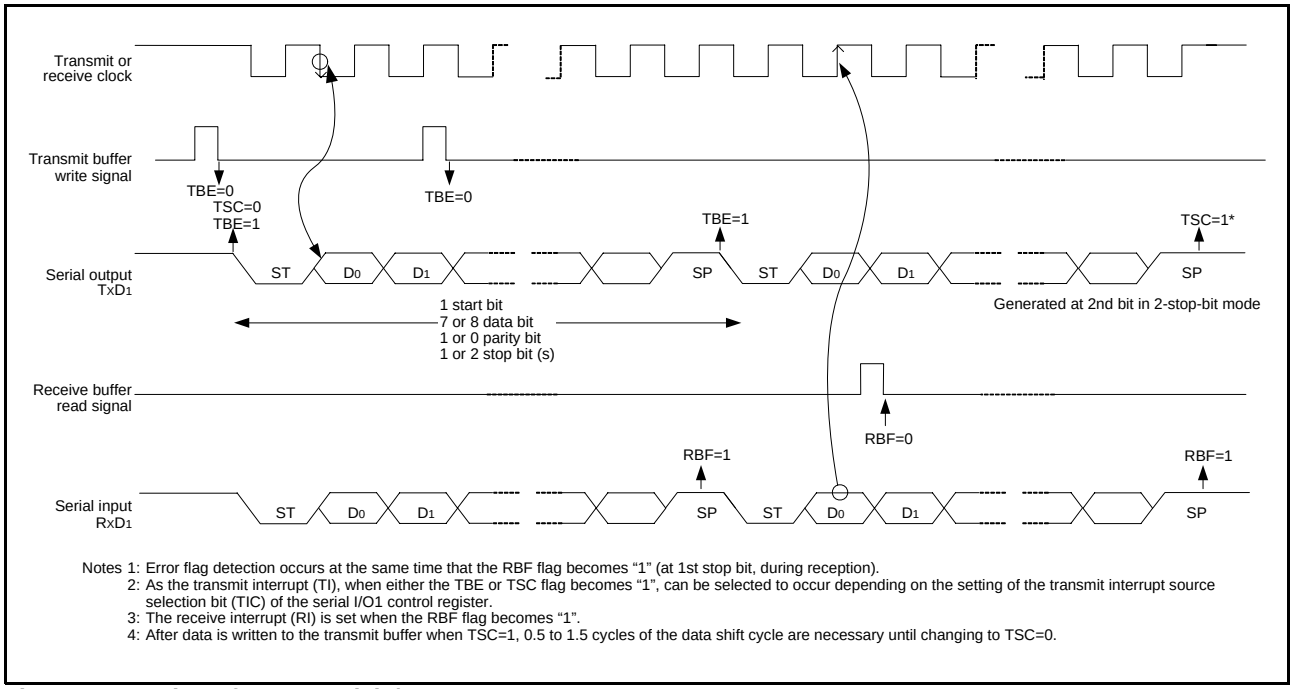


Fig. 38 Block diagram of UART serial I/O1



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 are necessary until changing to TSC=0.

Fig. 39 Operation of UART serial I/O1

[Transmit Buffer Register 1/Receive Buffer Register 1 (TB1/RB1)] 0018$_{16}$

The transmit buffer register 1 and the receive buffer register 1 are located at the same address. The transmit buffer is write-only and the receive buffer is read-only. If a character bit length is 7 bits, the MSB of data stored in the receive buffer is “0”.

[Serial I/O1 Status Register (SIO1STS)] 0019$_{16}$

The read-only serial I/O1 status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O1 function and various errors.
Three of the flags (bits 4 to 6) are valid only in UART mode.
The receive buffer full flag (bit 1) is cleared to “0” when the receive buffer register is read.
If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/O1 status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing “0” to the serial I/O1 enable bit SIOE (bit 7 of the serial I/O1 control register) also clears all the status flags, including the error flags.
Bits 0 to 6 of the serial I/O1 status register are initialized to “0” at reset, but if the transmit enable bit (bit 4) of the serial I/O1 control register has been set to “1”, the transmit shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become “1”.

[Serial I/O1 Control Register (SIO1CON)] 001A$_{16}$

The serial I/O1 control register consists of eight control bits for the serial I/O1 function.

[UART1 Control Register (UART1CON)] 001B$_{16}$

The UART control register consists of four control bits (bits 0 to 3) which are valid when asynchronous serial I/O is selected and set the data format of an data transfer, and one bit (bit 4) which is always valid and sets the output structure of the P4$_5$/TxD$_1$ pin.

[Baud Rate Generator 1 (BRG1)] 001C$_{16}$

The baud rate generator determines the baud rate for serial transfer.
The baud rate generator divides the frequency of the count source by 1/(n + 1), where n is the value written to the baud rate generator.

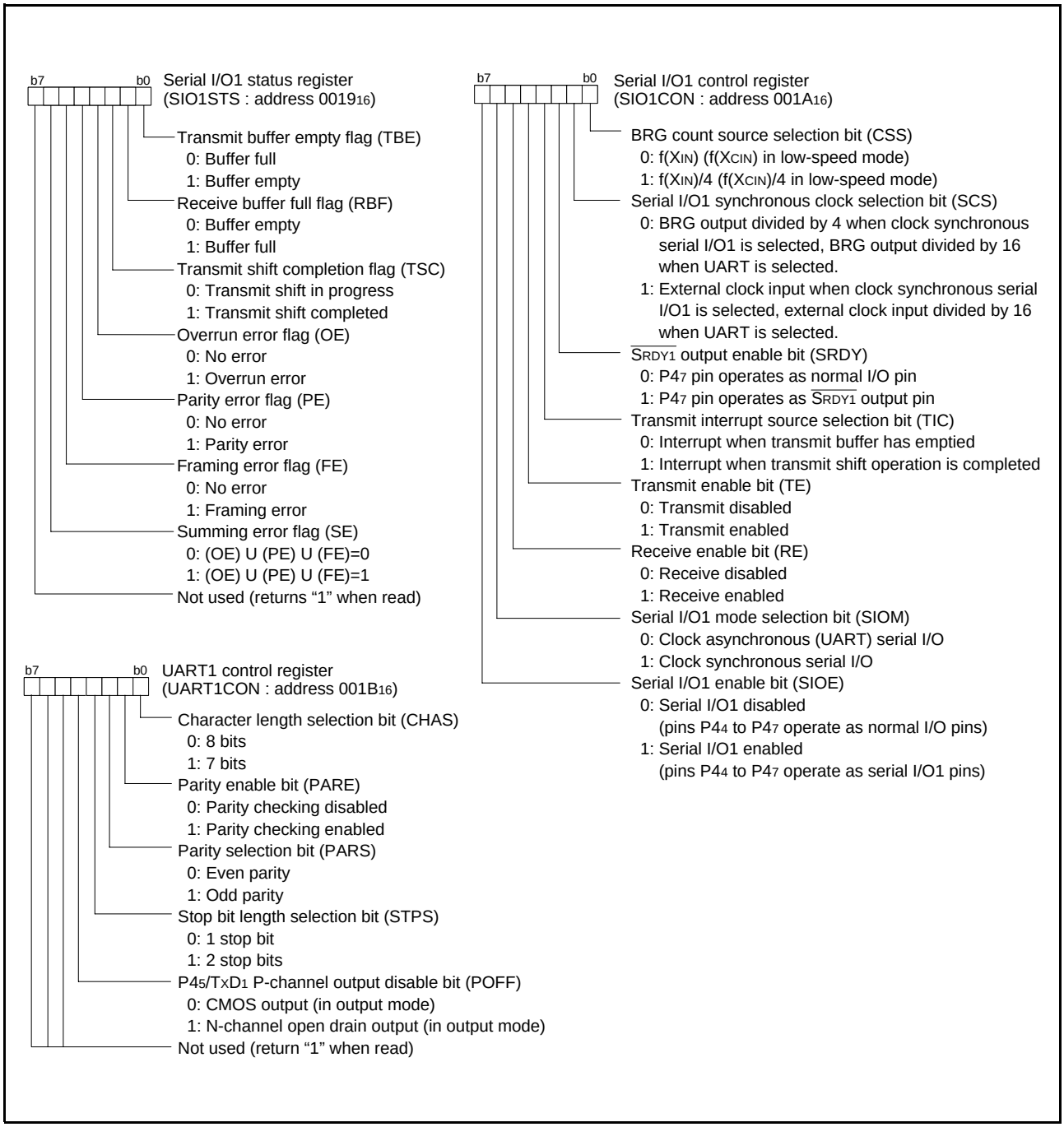


Fig. 40 Structure of serial I/O1 control registers

<Notes concerning serial I/O1>

1. Notes when selecting clock synchronous serial I/O

1.1 Stop of transmission operation

• Note

Clear the serial I/O1 enable bit and the transmit enable bit to "0" (serial I/O and transmit disabled).

• Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O1 enable bit is cleared to "0" (serial I/O disabled), the internal transmission is running (in this case, since pins TxD1, RxD1, SCLK1, and $\overline{\text{SRDY1}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O1 enable bit is set to "1" at this time, the data during internally shifting is output to the TxD1 pin and an operation failure occurs.

1.2 Stop of receive operation

• Note

Clear the receive enable bit to "0" (receive disabled), or clear the serial I/O1 enable bit to "0" (serial I/O disabled).

1.3 Stop of transmit/receive operation

• Note

Clear both the transmit enable bit and receive enable bit to "0" (transmit and receive disabled).

(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

• Reason

In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.

In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to "0" (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O1 enable bit to "0" (serial I/O disabled) (refer to 1.1).

2. Notes when selecting clock asynchronous serial I/O

2.1 Stop of transmission operation

• Note

Clear the transmit enable bit to "0" (transmit disabled). The transmission operation does not stop by clearing the serial I/O1 enable bit to "0".

• Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O1 enable bit is cleared to "0" (serial I/O disabled), the internal transmission is running (in this case, since pins TxD1, RxD1, SCLK1, and $\overline{\text{SRDY1}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O1 enable bit is set to "1" at this time, the data during internally shifting is output to the TxD1 pin and an operation failure occurs.

2.2 Stop of receive operation

• Note

Clear the receive enable bit to "0" (receive disabled).

2.3 Stop of transmit/receive operation

• Note 1 (only transmission operation is stopped)

Clear the transmit enable bit to "0" (transmit disabled). The transmission operation does not stop by clearing the serial I/O1 enable bit to "0".

• Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O1 enable bit is cleared to "0" (serial I/O disabled), the internal transmission is running (in this case, since pins TxD1, RxD1, SCLK1, and $\overline{\text{SRDY1}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O1 enable bit is set to "1" at this time, the data during internally shifting is output to the TxD1 pin and an operation failure occurs.

• Note 2 (only receive operation is stopped)

Clear the receive enable bit to "0" (receive disabled).

3. $\overline{\text{SRDY1}}$ output of reception side

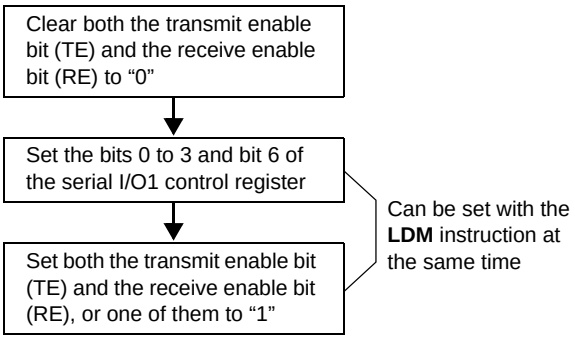
• Note

When signals are output from the $\overline{\text{SRDY1}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDY1}}$ output enable bit, and the transmit enable bit to “1” (transmit enabled).

4. Setting serial I/O1 control register again

• Note

Set the serial I/O1 control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to “0”.



5.Data transmission control with referring to transmit shift register completion flag

• Note

After the transmit data is written to the transmit buffer register, the transmit shift register completion flag changes from “1” to “0” with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

6. Transmission control when external clock is selected

• Note

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLK1 input level. Also, write data to the transmit buffer register at “H” of the SCLK1 input level.

7. Transmit interrupt request when transmit enable bit is set

• Note

When using the transmit interrupt, take the following sequence.

1. Set the serial I/O1 transmit interrupt enable bit to “0” (disabled).
2. Set the transmit enable bit to “1”.
3. Set the serial I/O1 transmit interrupt request bit to “0” after 1 or more instruction has executed.
4. Set the serial I/O1 transmit interrupt enable bit to “1” (enabled).

• Reason

When the transmit enable bit is set to “1”, the transmit buffer empty flag and the transmit shift register shift completion flag are also set to “1”. Therefore, regardless of selecting which timing for the generating of transmit interrupts, the interrupt request is generated and the transmit interrupt request bit is set at this point.

• Serial I/O2

The serial I/O2 function can be used only for clock synchronous serial I/O.

For clock synchronous serial I/O2, the transmitter and the receiver must use the same clock. If the internal clock is used, transfer is started by a write signal to the serial I/O2 register (address 001F16).

[Serial I/O2 Control Register (SIO2CON)] 001D16

The serial I/O2 control register contains eight bits which control various serial I/O2 functions.

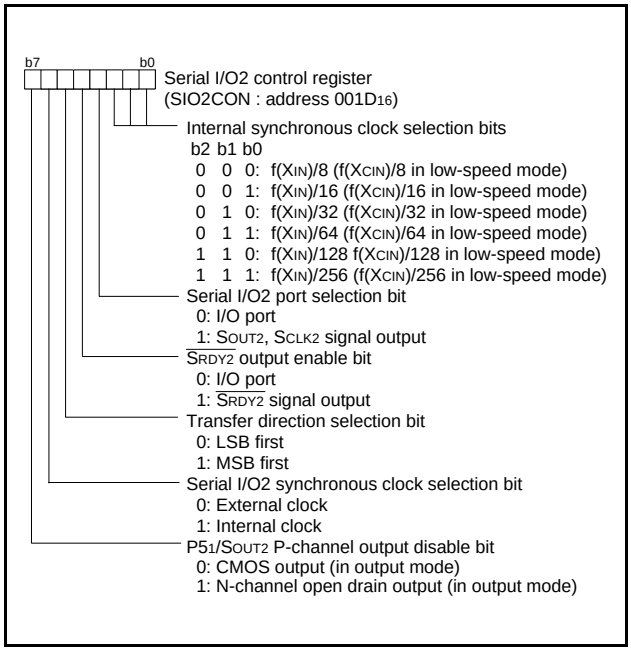


Fig. 41 Structure of Serial I/O2 control register

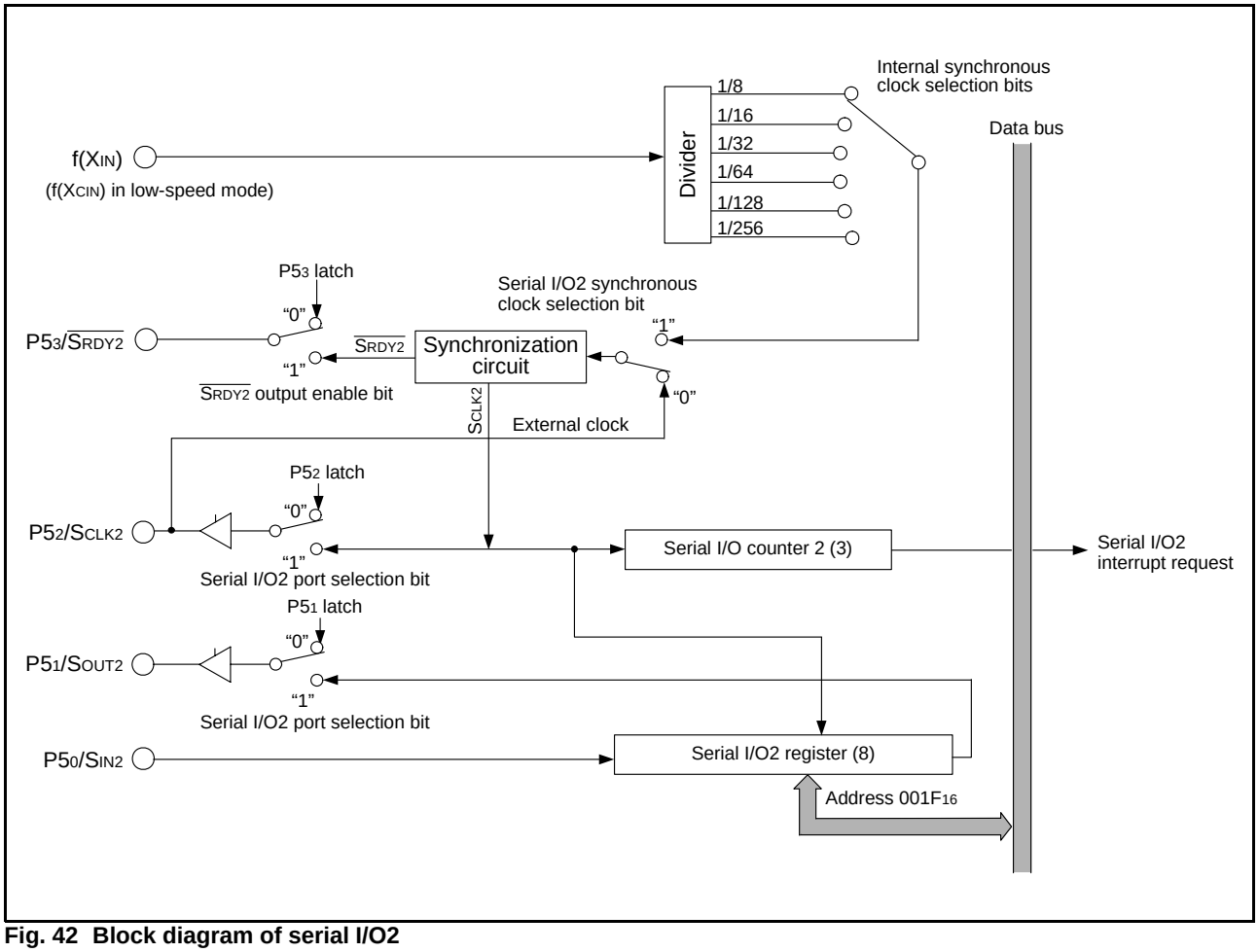
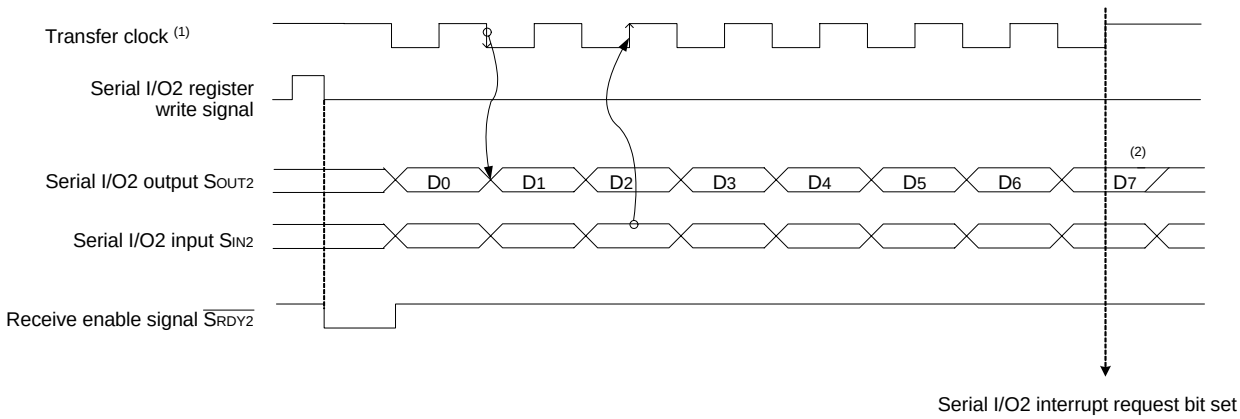


Fig. 42 Block diagram of serial I/O2



Notes 1: When the internal clock is selected as the transfer clock, the divide ratio of f(X_{IN}), or (f(X_{CIN}) in low-speed mode, can be selected by setting bits 0 to 2 of the serial I/O2 control register.
2: When the internal clock is selected as the transfer clock, the S_{OUT2} pin goes to high impedance after transfer completion.

Fig. 43 Timing of serial I/O2

• Serial I/O3

Serial I/O3 can be used as either clock synchronous or asynchronous (UART) serial I/O3. A dedicated timer is also provided for baud rate generation.

(1) Clock Synchronous Serial I/O Mode

Clock synchronous serial I/O3 mode can be selected by setting the serial I/O3 mode selection bit of the serial I/O3 control register (bit 6 of address 0032_{16}) to "1".

For clock synchronous serial I/O, 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 transmit/receive buffer register.

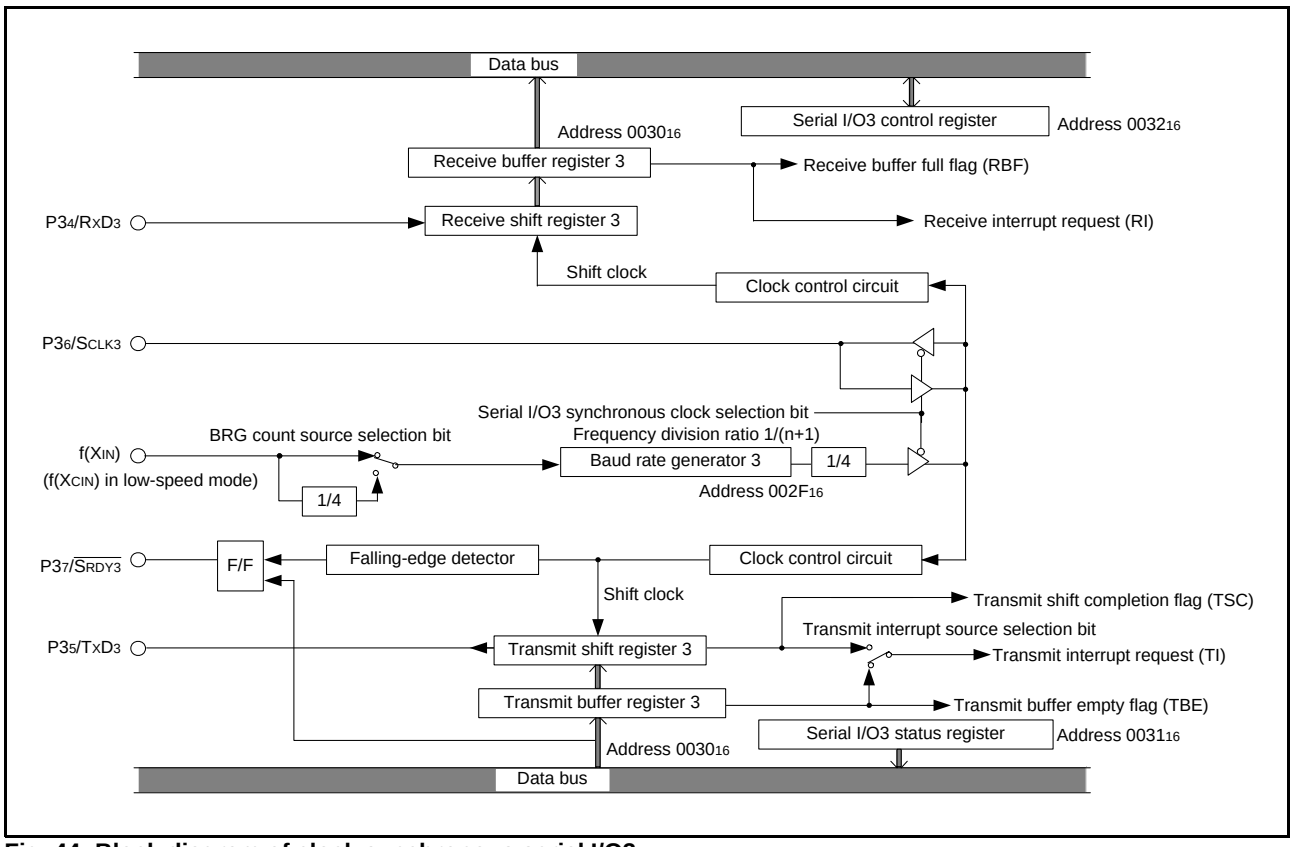
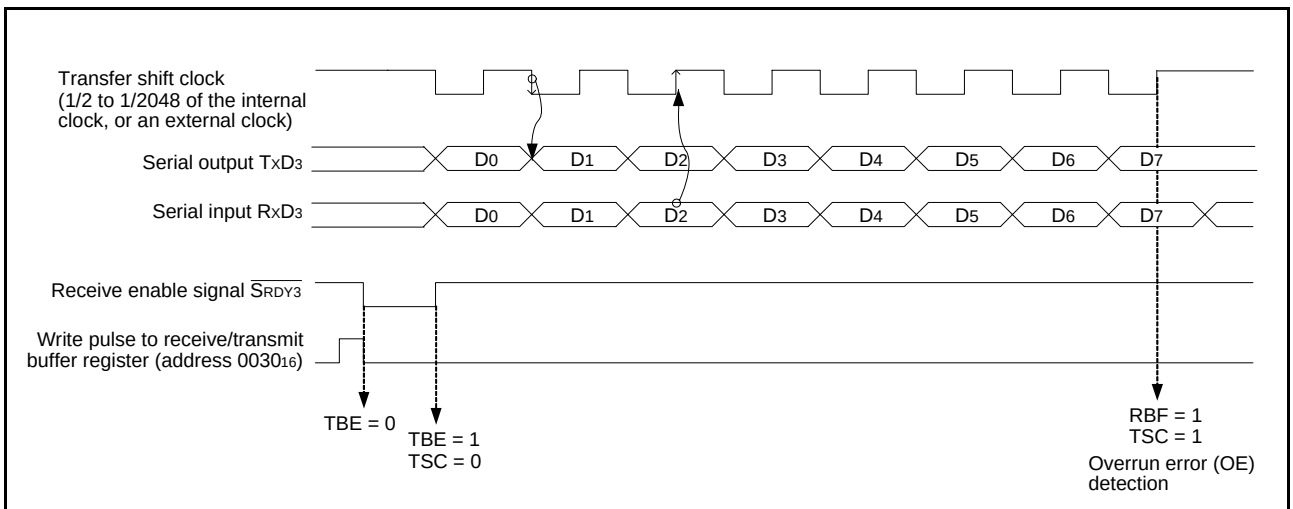


Fig. 44 Block diagram of clock synchronous serial I/O3



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/O3 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 pin.
3: The receive interrupt (RI) is set when the receive buffer full flag (RBF) becomes "1".

Fig. 45 Operation of clock synchronous serial I/O3

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

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O3 mode selection bit (b6) of the serial I/O3 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 a 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.

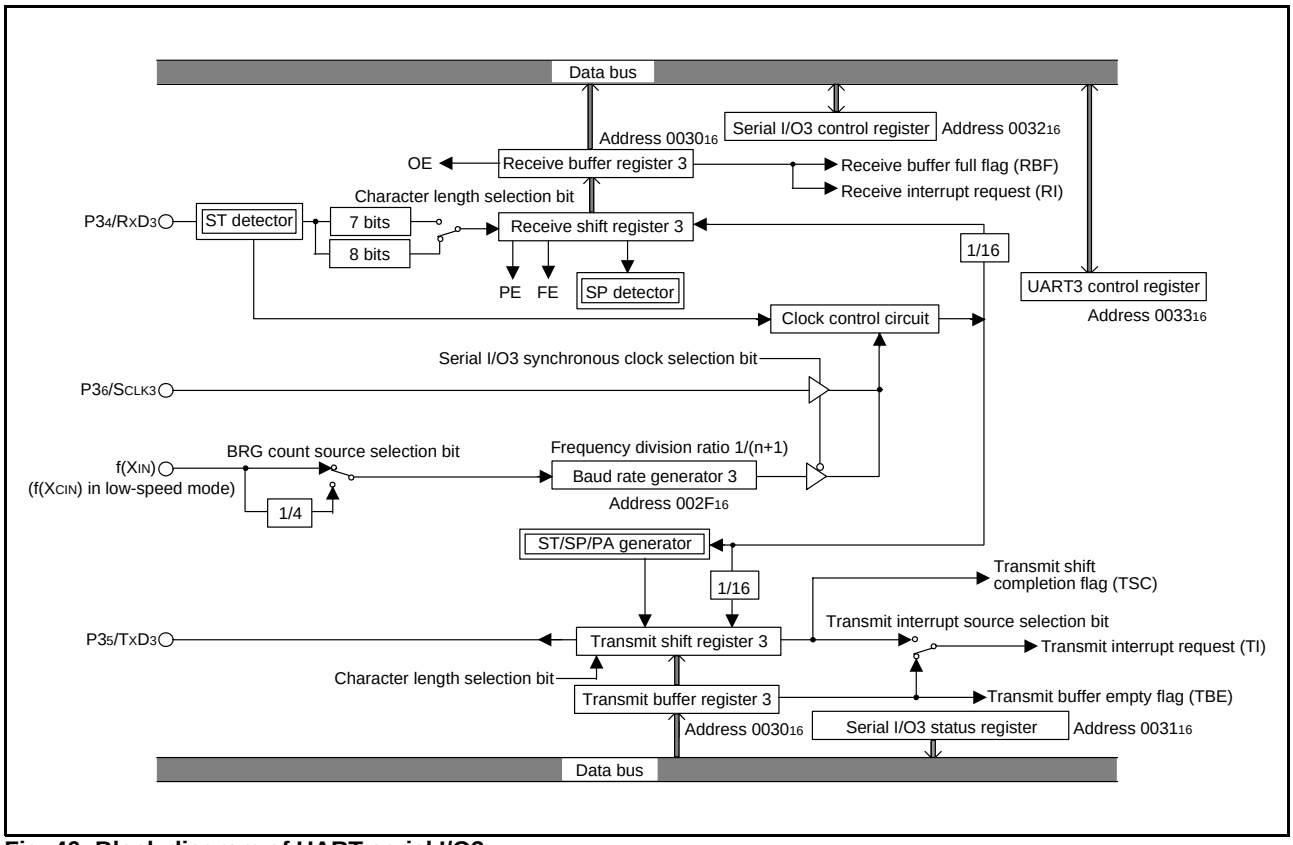
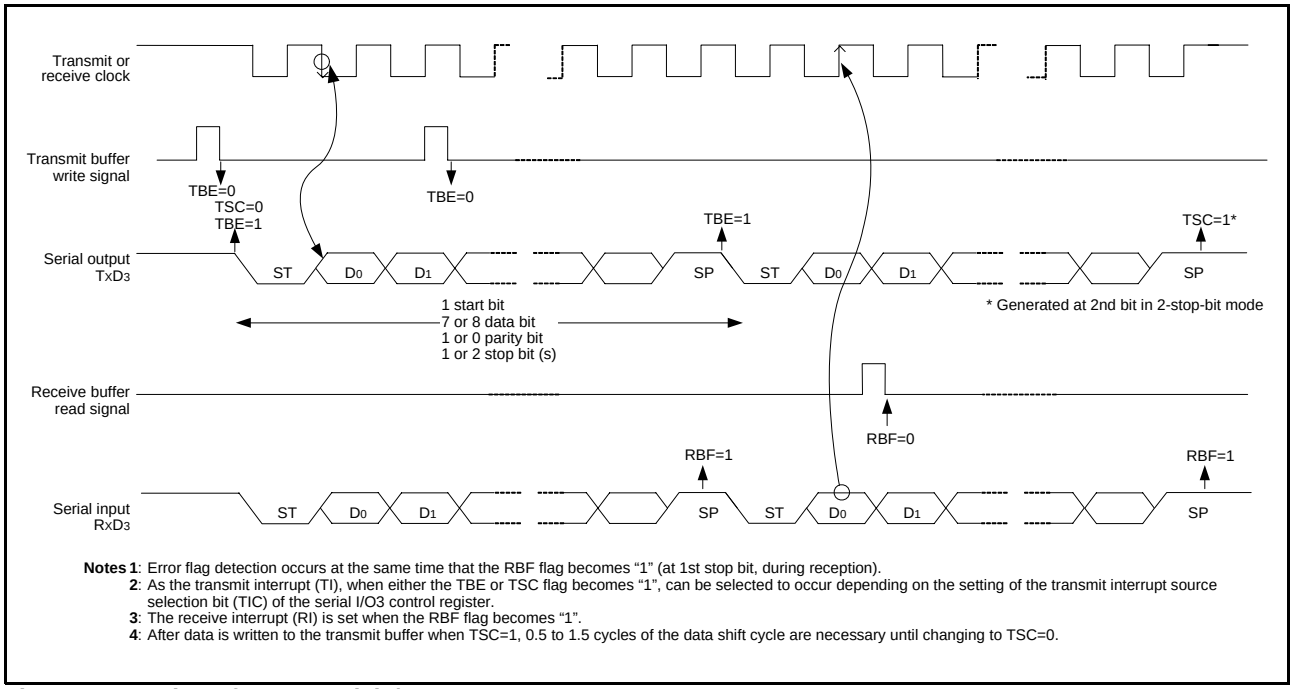


Fig. 46 Block diagram of UART serial I/O3



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/O3 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 are necessary until changing to TSC=0.

Fig. 47 Operation of UART serial I/O3

[Transmit Buffer Register 3/Receive Buffer Register 3 (TB3/RB3)] 0030$_{16}$

The transmit buffer register 3 and the receive buffer register 3 are located at the same address. The transmit buffer is write-only and the receive buffer is read-only. If a character bit length is 7 bits, the MSB of data stored in the receive buffer is "0".

[Serial I/O3 Status Register (SIO3STS)] 0031$_{16}$

The read-only serial I/O3 status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O3 function and various errors.
Three of the flags (bits 4 to 6) are valid only in UART mode.
The receive buffer full flag (bit 1) is cleared to "0" when the receive buffer register is read.
If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/O3 status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing "0" to the serial I/O3 enable bit SIOE (bit 7 of the serial I/O3 control register) also clears all the status flags, including the error flags.
Bits 0 to 6 of the serial I/O3 status register are initialized to "0" at reset, but if the transmit enable bit (bit 4) of the serial I/O3 control register has been set to "1", the transmit shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become "1".

[Serial I/O3 Control Register (SIO3CON)] 0032$_{16}$

The serial I/O3 control register consists of eight control bits for the serial I/O3 function.

[UART3 Control Register (UART3CON)] 0033$_{16}$

The UART control register consists of four control bits (bits 0 to 3) which are valid when asynchronous serial I/O is selected and set the data format of an data transfer, and one bit (bit 4) which is always valid and sets the output structure of the P3$_5$/TxD$_3$ pin.

[Baud Rate Generator 3 (BRG3)] 002F$_{16}$

The baud rate generator determines the baud rate for serial transfer.
The baud rate generator divides the frequency of the count source by 1/(n + 1), where n is the value written to the baud rate generator.

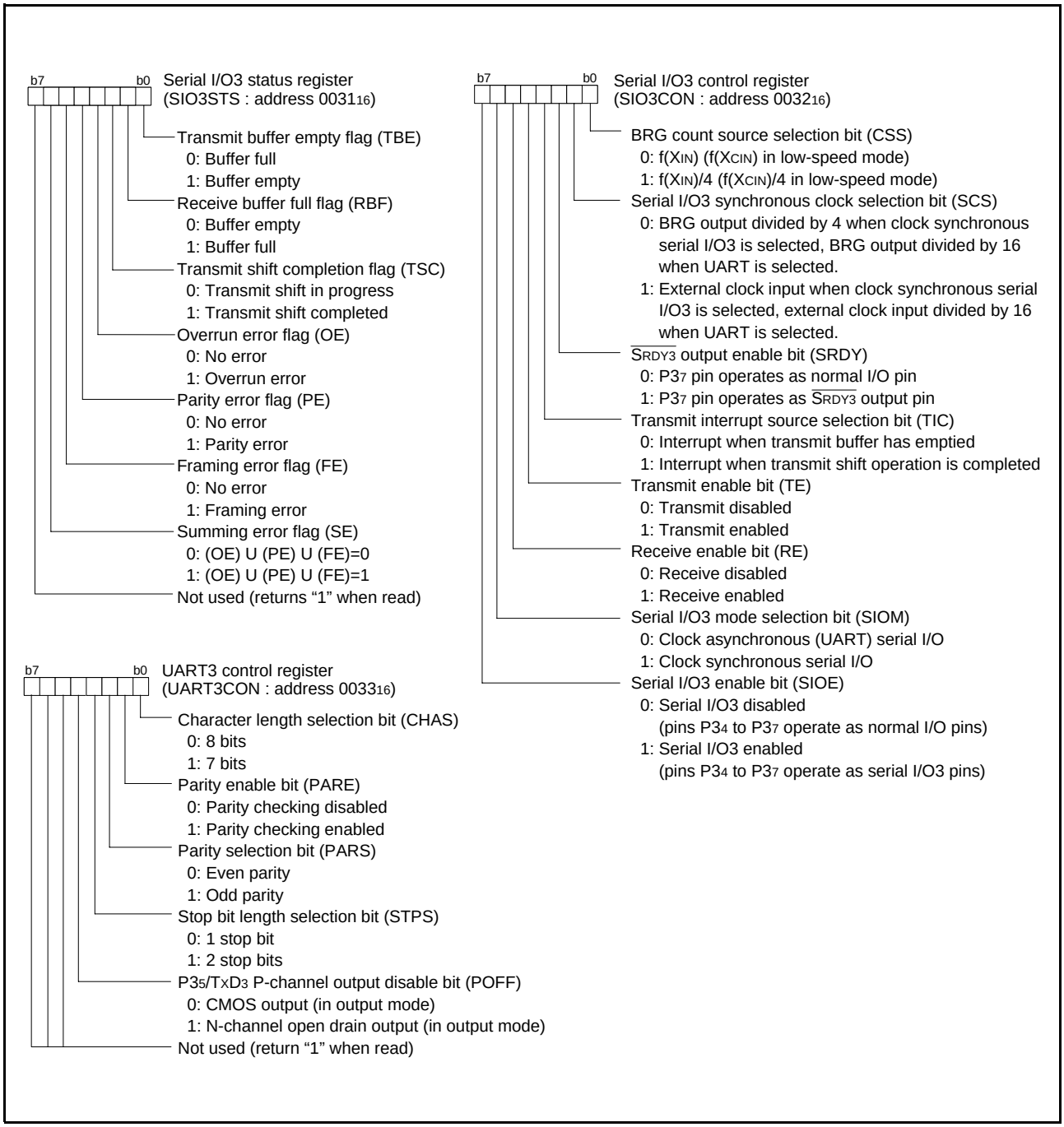


Fig. 48 Structure of serial I/O3 control registers

<Notes concerning serial I/O3>

1. Notes when selecting clock synchronous serial I/O

1.1 Stop of transmission operation

• Note

Clear the serial I/O3 enable bit and the transmit enable bit to "0" (serial I/O and transmit disabled).

• Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O3 enable bit is cleared to "0" (serial I/O disabled), the internal transmission is running (in this case, since pins TxD3, RxD3, SCLK3, and $\overline{\text{SRDY3}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O3 enable bit is set to "1" at this time, the data during internally shifting is output to the TxD3 pin and an operation failure occurs.

1.2 Stop of receive operation

• Note

Clear the receive enable bit to "0" (receive disabled), or clear the serial I/O3 enable bit to "0" (serial I/O disabled).

1.3 Stop of transmit/receive operation

• Note

Clear both the transmit enable bit and receive enable bit to "0" (transmit and receive disabled).

(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

• Reason

In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.

In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to "0" (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O3 enable bit to "0" (serial I/O disabled) (refer to 1.1).

2. Notes when selecting clock asynchronous serial I/O

2.1 Stop of transmission operation

• Note

Clear the transmit enable bit to "0" (transmit disabled). The transmission operation does not stop by clearing the serial I/O3 enable bit to "0".

• Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O3 enable bit is cleared to "0" (serial I/O disabled), the internal transmission is running (in this case, since pins TxD3, RxD3, SCLK3, and $\overline{\text{SRDY3}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O3 enable bit is set to "1" at this time, the data during internally shifting is output to the TxD3 pin and an operation failure occurs.

2.2 Stop of receive operation

• Note

Clear the receive enable bit to "0" (receive disabled).

2.3 Stop of transmit/receive operation

• Note 1 (only transmission operation is stopped)

Clear the transmit enable bit to "0" (transmit disabled). The transmission operation does not stop by clearing the serial I/O3 enable bit to "0".

• Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O3 enable bit is cleared to "0" (serial I/O disabled), the internal transmission is running (in this case, since pins TxD3, RxD3, SCLK3, and $\overline{\text{SRDY3}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O3 enable bit is set to "1" at this time, the data during internally shifting is output to the TxD3 pin and an operation failure occurs.

• Note 2 (only receive operation is stopped)

Clear the receive enable bit to "0" (receive disabled).

3. $\overline{\text{SRDY3}}$ output of reception side

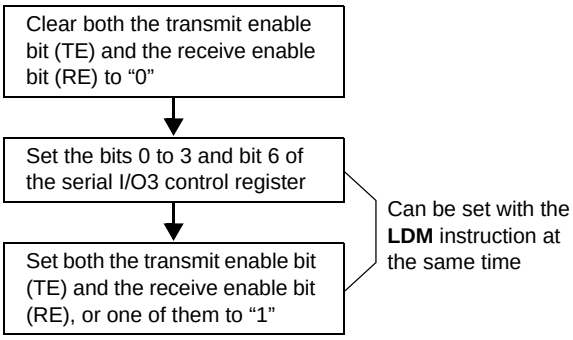
• Note

When signals are output from the $\overline{\text{SRDY3}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDY3}}$ output enable bit, and the transmit enable bit to "1" (transmit enabled).

4. Setting serial I/O3 control register again

• Note

Set the serial I/O3 control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to "0".



5.Data transmission control with referring to transmit shift register completion flag

• Note

After the transmit data is written to the transmit buffer register, the transmit shift register completion flag changes from "1" to "0" with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

6. Transmission control when external clock is selected

• Note

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to "1" at "H" of the SCLK3 input level. Also, write data to the transmit buffer register at "H" of the SCLK input level.

7. Transmit interrupt request when transmit enable bit is set

• Note

When using the transmit interrupt, take the following sequence.

1. Set the serial I/O3 transmit interrupt enable bit to "0" (disabled).
2. Set the transmit enable bit to "1".
3. Set the serial I/O3 transmit interrupt request bit to "0" after 1 or more instruction has executed.
4. Set the serial I/O3 transmit interrupt enable bit to "1" (enabled).

• Reason

When the transmit enable bit is set to "1", the transmit buffer empty flag and the transmit shift register shift completion flag are also set to "1". Therefore, regardless of selecting which timing for the generating of transmit interrupts, the interrupt request is generated and the transmit interrupt request bit is set at this point.

PWM (PWM: Pulse Width Modulation)

The 3804 group (Spec.L) has PWM functions with an 8-bit resolution, based on a signal that is the clock input XIN or that clock input divided by 2 or the clock input XCIN or that clock input divided by 2 in low-speed mode.

• Data Setting

The PWM output pin also functions as port P56. Set the PWM period by the PWM prescaler, and set the "H" term of output pulse by the PWM register.
If the value in the PWM prescaler is n and the value in the PWM register is m (where n = 0 to 255 and m = 0 to 255):
PWM period = 255 × (n+1) / f(XIN)
= 31.875 × (n+1) μs
(when f(XIN) = 8 MHz, count source selection bit = "0")
Output pulse "H" term = PWM period × m / 255
= 0.125 × (n+1) × m μs
(when f(XIN) = 8 MHz, count source selection bit = "0")

• PWM Operation

When bit 0 (PWM enable bit) of the PWM control register is set to "1", operation starts by initializing the PWM output circuit, and pulses are output starting at an "H".
If the PWM register or PWM prescaler is updated during PWM output, the pulses will change in the cycle after the one in which the change was made.

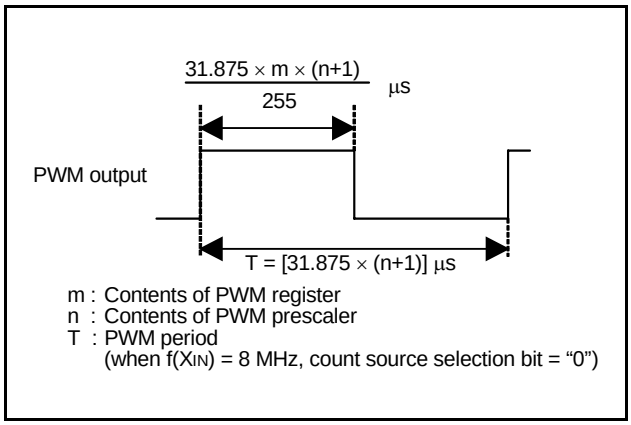


Fig. 49 Timing of PWM period

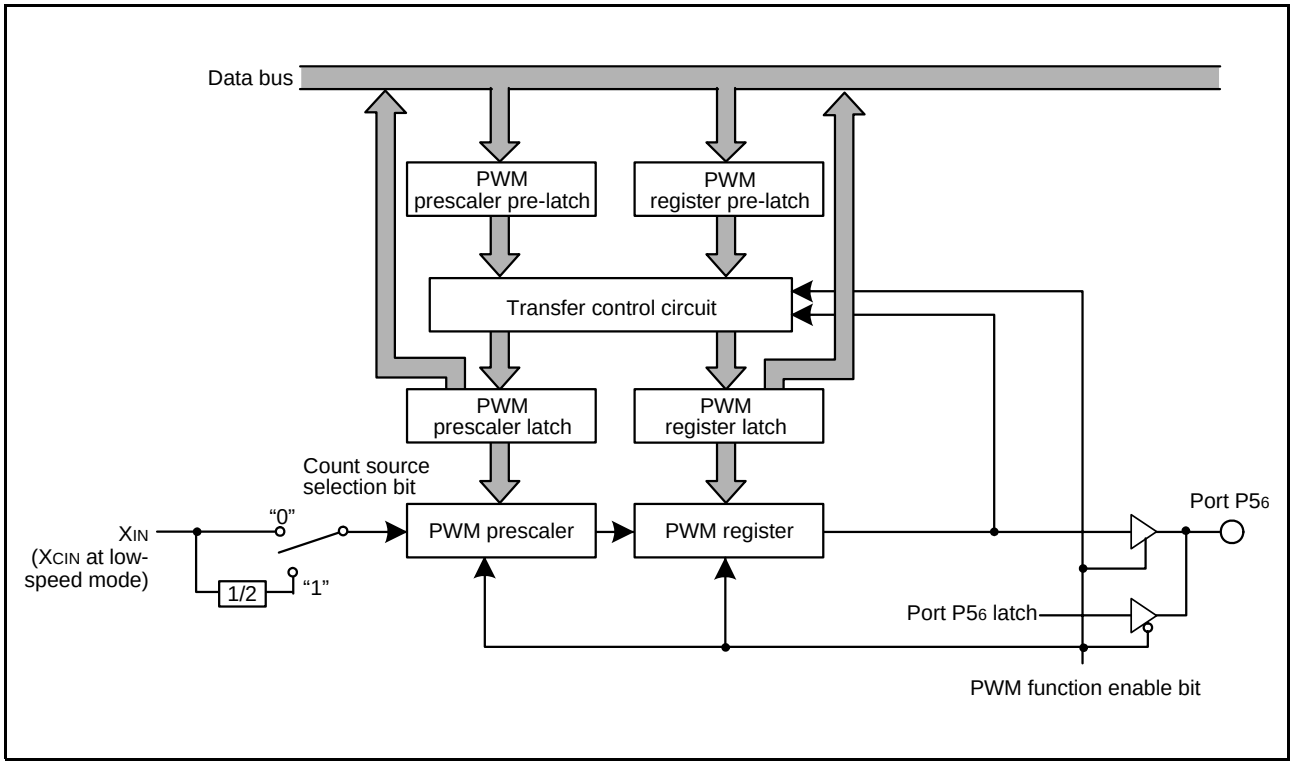


Fig. 50 Block diagram of PWM function

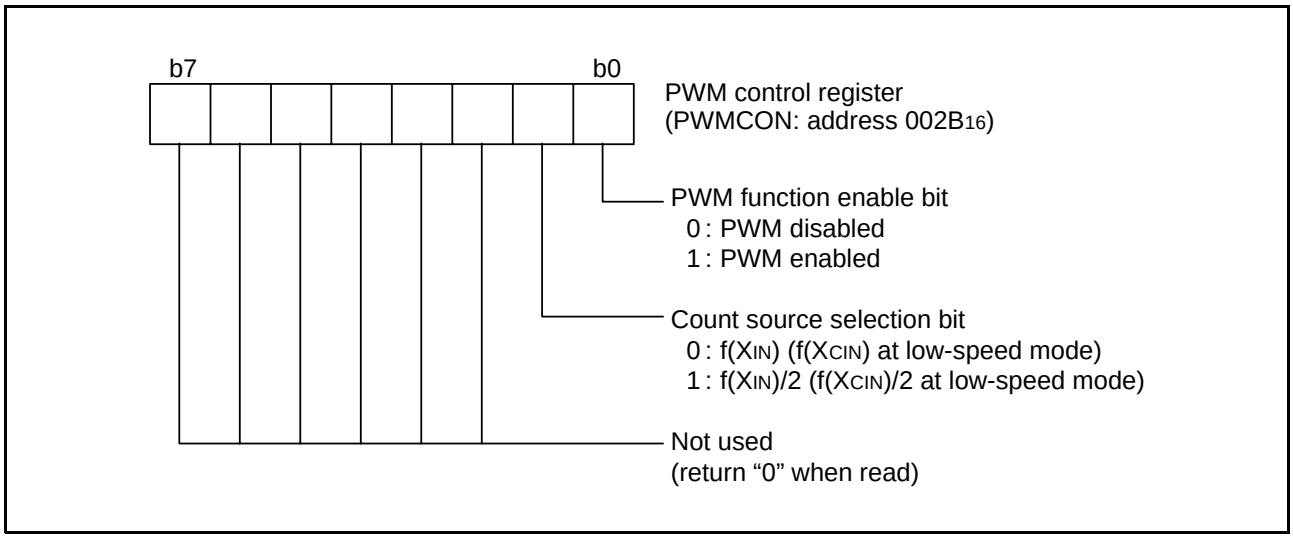


Fig. 51 Structure of PWM control register

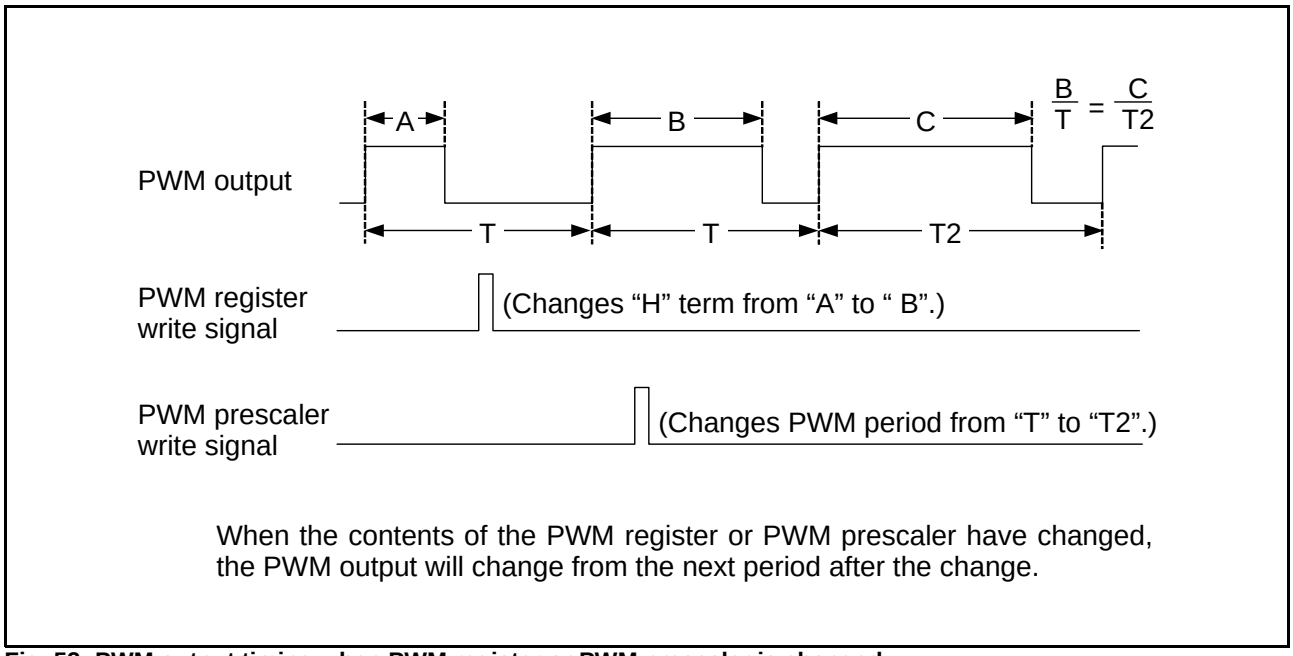


Fig. 52 PWM output timing when PWM register or PWM prescaler is changed

<Notes>

The PWM starts after the PWM function enable bit is set to enable and “L” level is output from the PWM pin.
The length of this “L” level output is as follows:

$\frac{n+1}{2 \times f(X_{IN})}$ sec (Count source selection bit = 0, where n is the value set in the prescaler)

$\frac{n+1}{f(X_{IN})}$ sec (Count source selection bit = 1, where n is the value set in the prescaler)

A/D CONVERTER (successive approximation type)

[AD Conversion Register 1, 2 (AD1, AD2)] 0035$_{16}$, 0038$_{16}$

The AD conversion register is a read-only register that stores the result of an A/D conversion. When reading this register during an A/D conversion, the previous conversion result is read.

Bit 7 of the AD conversion register 2 is the conversion mode selection bit. When this bit is set to "0", the A/D converter becomes the 10-bit A/D mode. When this bit is set to "1", that becomes the 8-bit A/D mode. The conversion result of the 8-bit A/D mode is stored in the AD conversion register 1. As for 10-bit A/D mode, not only 10-bit reading but also only high-order 8-bit reading of conversion result can be performed by selecting the reading procedure of the AD conversion registers 1, 2 after A/D conversion is completed (in Figure 54).

As for 10-bit A/D mode, the 8-bit reading inclined to MSB is performed when reading the AD converter register 1 after A/D conversion is started; and when the AD converter register 1 is read after reading the AD converter register 2, the 8-bit reading inclined to LSB is performed.

[AD/DA Control Register (ADCON)] 0034$_{16}$

The AD/DA control register controls the A/D conversion process. Bits 0 to 2 and bit 4 select a specific analog input pin. Bit 3 signals the completion of an A/D conversion. The value of this bit remains at "0" during an A/D conversion, and changes to "1" when an A/D conversion ends. Writing "0" to this bit starts the A/D conversion.

• Comparison Voltage Generator

The comparison voltage generator divides the voltage between AVSS and VREF into 1024, and that outputs the comparison voltage in the 10-bit A/D mode (256 division in 8-bit A/D mode). The A/D converter successively compares the comparison voltage Vref in each mode, dividing the VREF voltage (see below), with the input voltage.

- 10-bit A/D mode (10-bit reading)

$$Vref = \frac{VREF}{1024} \times n \ (n = 0 – 1023)$$

- 10-bit A/D mode (8-bit reading)

$$Vref = \frac{VREF}{256} \times n \ (n = 0 – 255)$$

- 8-bit A/D mode

$$Vref = \frac{VREF}{256} \times (n – 0.5) \ (n = 1 – 255)$$

$$= 0 \qquad (n = 0)$$

• Channel Selector

The channel selector selects one of ports P67/AN7 to P60/AN0 or P07/AN15 to P00/AN8, and inputs the voltage to the comparator.

• Comparator and Control Circuit

The comparator and control circuit compares an analog input voltage with the comparison voltage, and then stores the result in the AD conversion registers 1, 2. When an A/D conversion is completed, the control circuit sets the AD conversion completion bit and the AD interrupt request bit to "1".

Note that because the comparator consists of a capacitor coupling, set f(XIN) to 500 kHz or more during an A/D conversion.

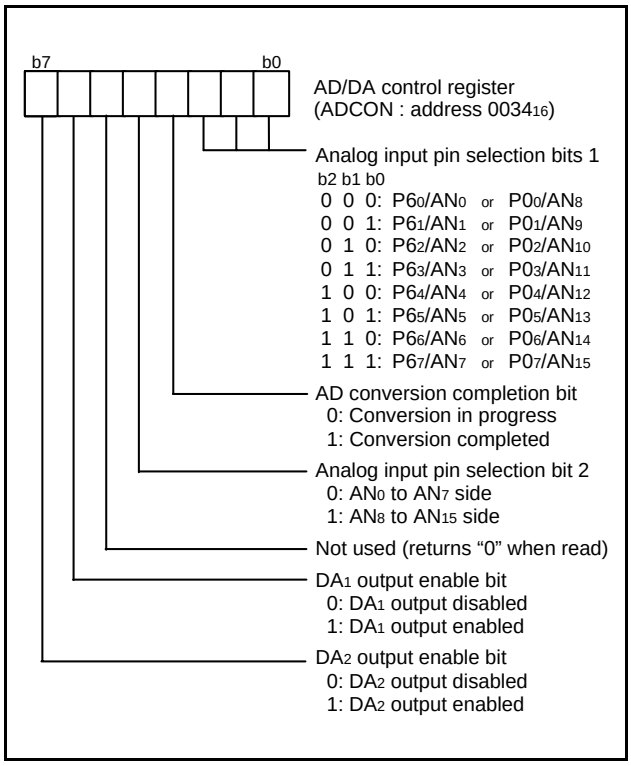


Fig. 53 Structure of AD/DA control register

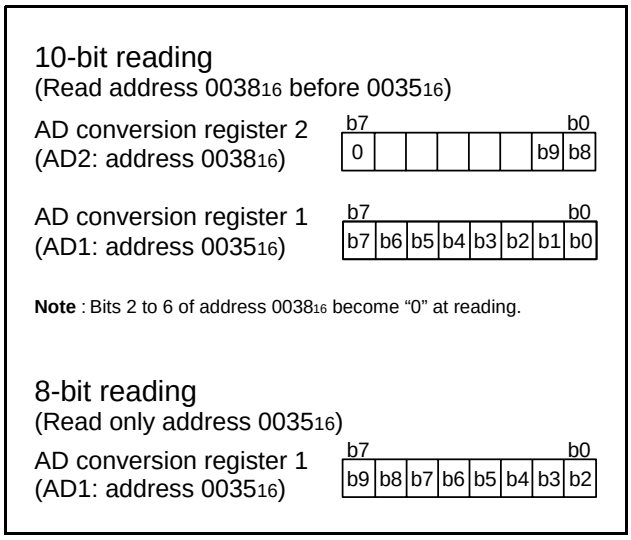


Fig. 54 Structure of 10-bit A/D mode reading

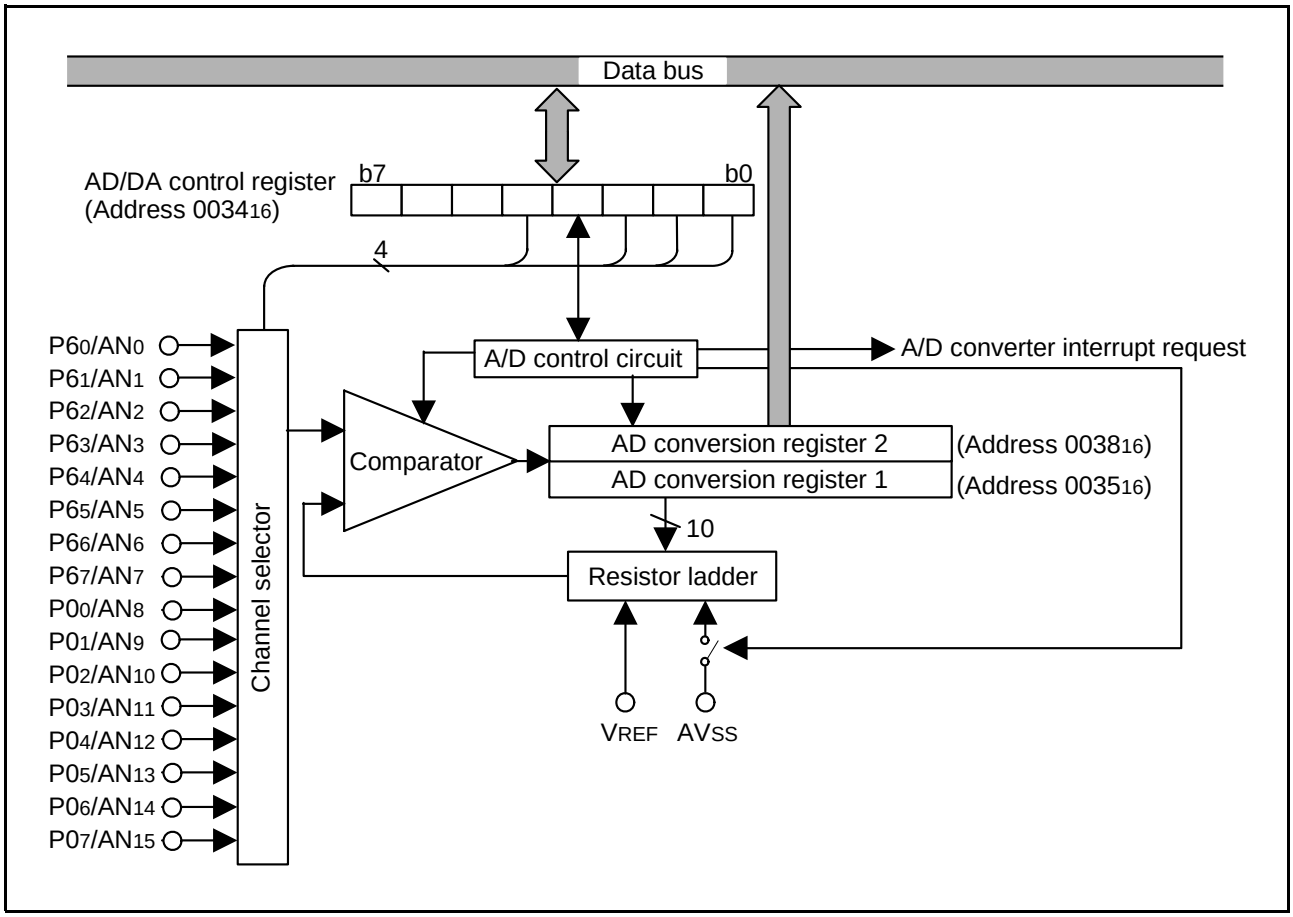


Fig. 55 Block diagram of A/D converter

D/A CONVERTER

The 3804 group (Spec.L) has two internal D/A converters (DA_1 and DA_2) with 8-bit resolution.
The D/A conversion is performed by setting the value in each DA conversion register. The result of D/A conversion is output from the DA_1 or DA_2 pin by setting the DA output enable bit to "1".
When using the D/A converter, the corresponding port direction register bit ($P3_0/DA_1$ or $P3_1/DA_2$) must be set to "0" (input status).
The output analog voltage V is determined by the value n (decimal notation) in the DA conversion register as follows:

$V = V_{REF} \times n/256$ (n = 0 to 255)
Where V_{REF} is the reference voltage.

At reset, the DA conversion registers are cleared to "00_{16}", and the DA output enable bits are cleared to "0", and the $P3_0/DA_1$ and $P3_1/DA_2$ pins become high impedance.
The DA output does not have buffers. Accordingly, connect an external buffer when driving a low-impedance load.

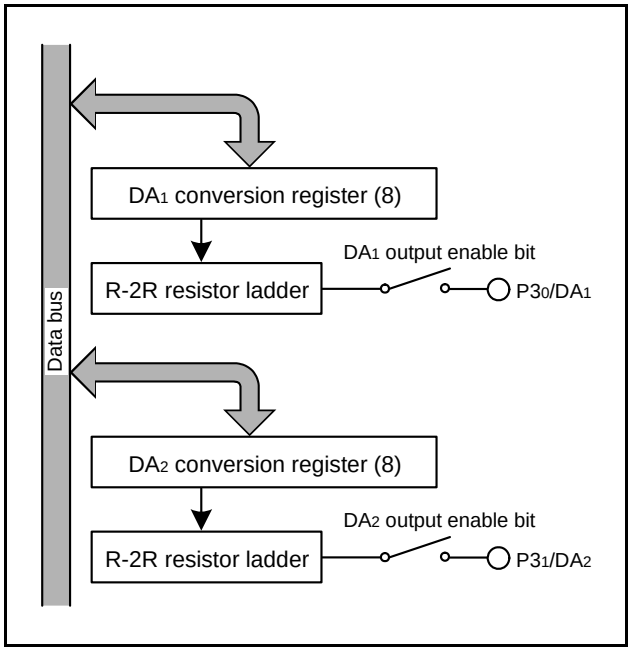


Fig. 56 Block diagram of D/A converter

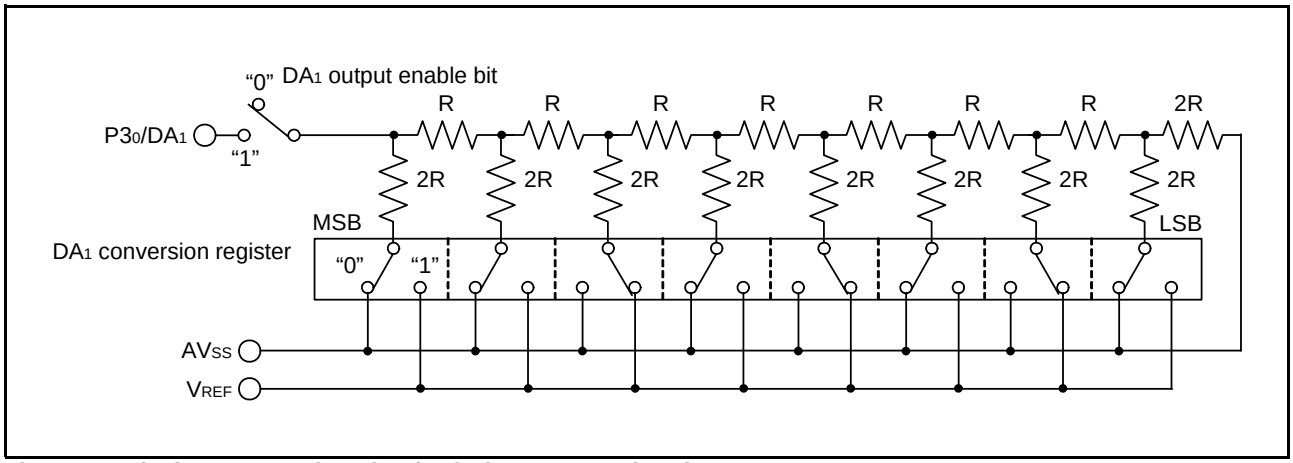


Fig. 57 Equivalent connection circuit of D/A converter (DA1)

WATCHDOG TIMER

The watchdog timer gives a mean of returning to the reset status when a program cannot run on a normal loop (for example, because of a software run-away). The watchdog timer consists of an 8-bit watchdog timer L and an 8-bit watchdog timer H.

• Watchdog Timer Initial Value

Watchdog timer L is set to "FF16" and watchdog timer H is set to "FF16" by writing to the watchdog timer control register (address 001E16) or at a reset. Any write instruction that causes a write signal can be used, such as the STA, LDM, CLB, etc. Data can only be written to bits 6 and 7 of the watchdog timer control register. Regardless of the value written to bits 0 to 5, the above-mentioned value will be set to each timer.
Bit 6 can be written only once after releasing reset. After rewriting it is disable to write any data to this bit.

• Watchdog Timer Operations

The watchdog timer stops at reset and starts to count down by writing to the watchdog timer control register (address 001E16). An internal reset occurs at an underflow of the watchdog timer H. The reset is released after waiting for a reset release time and the program is processed from the reset vector address. Accordingly, programming is usually performed so that writing to the watchdog timer control register may be started before an underflow. If writing to the watchdog timer control register is not performed once, the watchdog timer does not function.

• Bit 6 of Watchdog Timer Control Register

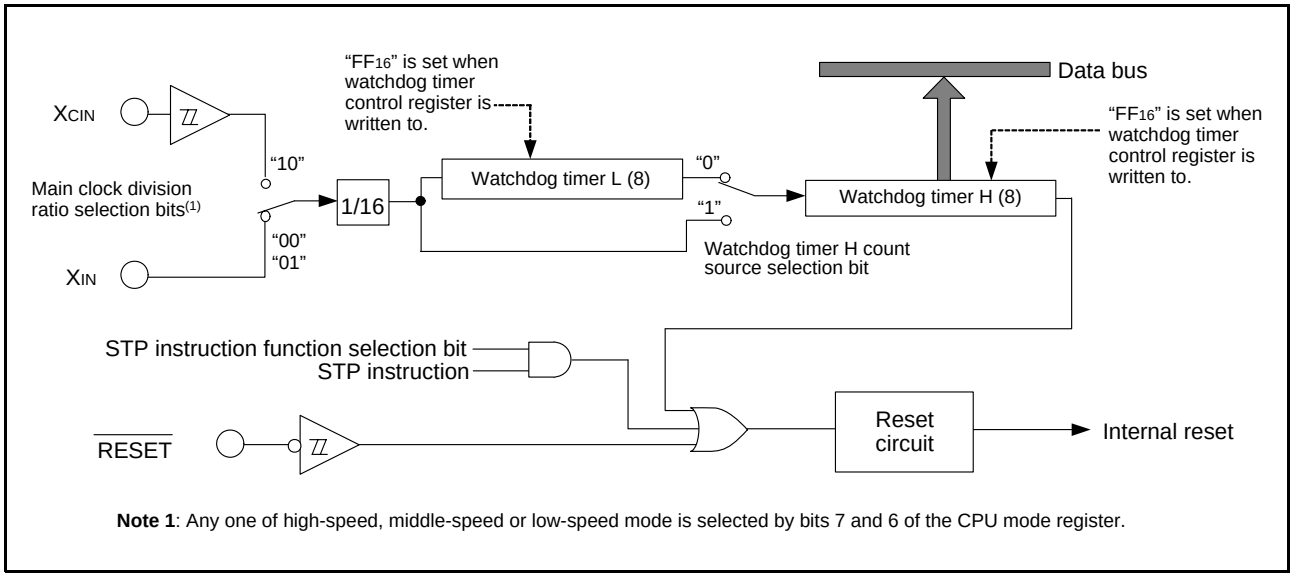
- When bit 6 of the watchdog timer control register is "0", the MCU enters the stop mode by execution of STP instruction. Just after releasing the stop mode, the watchdog timer restarts counting(Note.). When executing the WIT instruction, the watchdog timer does not stop.
- When bit 6 is "1", execution of STP instruction causes an internal reset. When this bit is set to "1" once, it cannot be rewritten to "0" by program. Bit 6 is "0" at reset.

The following shows the period between the write execution to the watchdog timer control register and the underflow of watchdog timer H.

Bit 7 of the watchdog timer control register is "0":
when XCIN = 32.768 kHz; 32 s
when XIN = 16 MHz; 65.536 ms

Bit 7 of the watchdog timer control register is "1":
when XCIN = 32.768 kHz; 125 ms
when XIN = 16 MHz; 256 μs

Note. The watchdog timer continues to count even during the wait time set by timer 1 and timer 2 to release the stop state and in the wait mode. Accordingly, write to the watchdog timer control register to not underflow the watchdog timer in this time.



Note 1: Any one of high-speed, middle-speed or low-speed mode is selected by bits 7 and 6 of the CPU mode register.

Fig. 58 Block diagram of Watchdog timer

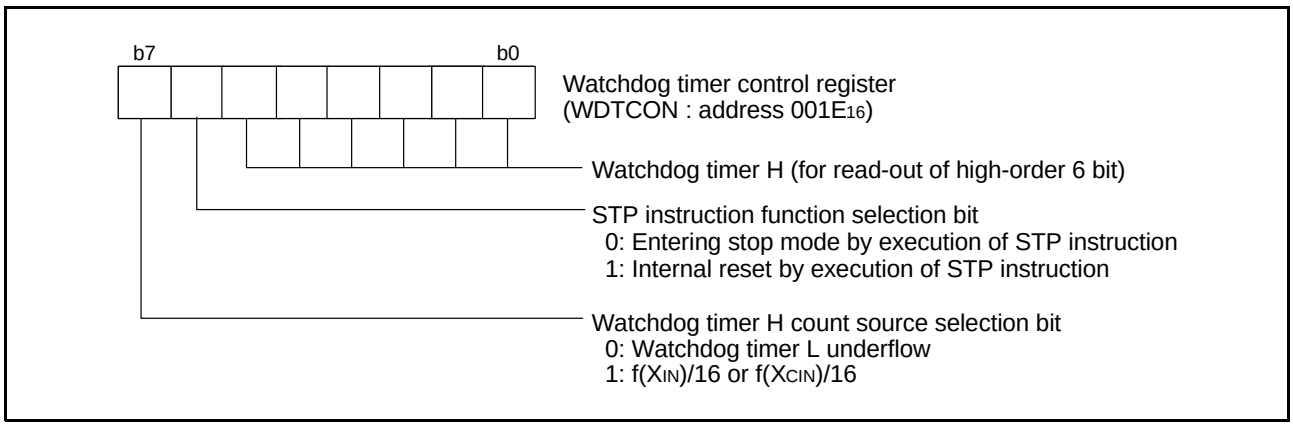


Fig. 59 Structure of Watchdog timer control register

MULTI-MASTER I²C-BUS INTERFACE

The 3804 group (Spec. L) has the multi-master I²C-BUS interface. The multi-master I²C-BUS interface is a serial communications circuit, conforming to the Philips I²C-BUS data transfer format. This interface, offering both arbitration lost detection and a synchronous functions, is useful for the multi-master serial communications.

Figure 60 shows a block diagram of the multi-master I²C-BUS interface and Table 9 lists the multi-master I²C-BUS interface functions.

This multi-master I²C-BUS interface consists of the I²C slave address registers 0 to 2, the I²C data shift register, the I²C clock control register, the I²C control register, the I²C status register, the I²C START/STOP condition control register, the I²C special mode control register, the I²C special mode status register, and other control circuits.

When using the multi-master I²C-BUS interface, set 1 MHz or more to the internal clock ϕ.

Table 9 Multi-master I²C-BUS interface functions

Item	Function
Format	In conformity with Philips I²C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	In conformity with Philips I²C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequency	16.1 kHz to 400 kHz (at ϕ = 4 MHz)

System clock ϕ = f(XIN)/2 (high-speed mode)
ϕ = f(XIN)/8 (middle-speed mode)

NOTE:
1. We are not responsible for any third party's infringement of patent rights or other rights attributable to the use of the control function (bit 6 of the I²C control register at address 002E16) for connections between the I²C-BUS interface and ports (SCL1, SCL2, SDA1, and SDA2).

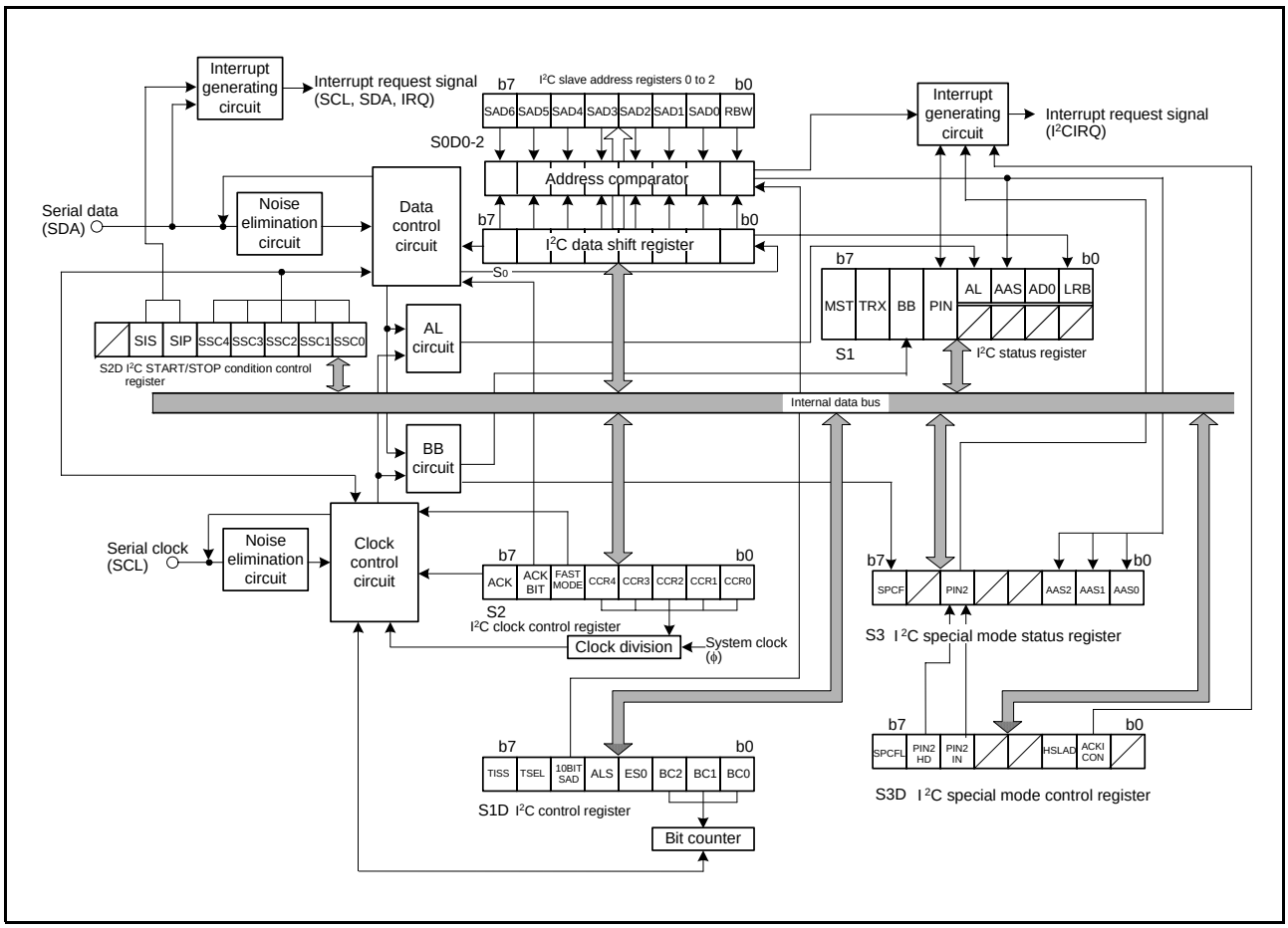


Fig. 60 Block diagram of multi-master I²C-BUS interface

*: Purchase of Renesas Technology Corporation's I²C components conveys a license under the Philips I²C Patent Rights to use these components an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

[I²C Data Shift Register (S0)] 0011₁₆

The I²C data shift register (S0: address 0011₁₆) is an 8-bit shift register to store receive data and write transmit data.
When transmit data is written into this register, it is transferred to the outside from bit 7 in synchronization with the SCL, and each time one-bit data is output, the data of this register are shifted by one bit to the left. When data is received, it is input to this register from bit 0 in synchronization with the SCL, and each time one-bit data is input, the data of this register are shifted by one bit to the left. The minimum 2 cycles of the internal clock ϕ are required from the rising of the SCL until input to this register.
The I²C data shift register is in a write enable status only when the I²C-BUS interface enable bit (ES0 bit) of the I²C control register (S1D: address 0014₁₆) is "1". The bit counter is reset by a write instruction to the I²C data shift register. When both the ES0 bit and the MST bit of the I²C status register (S1: address 0013₁₆) are "1", the SCL is output by a write instruction to the I²C data shift register. Reading data from the I²C data shift register is always enabled regardless of the ES0 bit value.

[I²C Slave Address Registers 0 to 2 (S0D0 to S0D2)] 0FF7₁₆ to 0FF9₁₆

The I²C slave address registers 0 to 2 (S0D0 to S0D2: addresses 0FF7₁₆ to 0FF9₁₆) consists of a 7-bit slave address and a read/write bit. In the addressing mode, the slave address written in this register is compared with the address data to be received immediately after the START condition is detected.

• Bit 0: Read/write bit (RWB)

This is not used in the 7-bit addressing mode. In the 10-bit addressing mode, set RWB to "0" because the first address data to be received is compared with the contents (SAD6 to SAD0 + RWB) of the I²C slave address registers 0 to 2.
When 2-byte address data match slave address, a 7-bit slave address which is received after restart condition has detected and R/W data can be matched by setting "1" to RWB with software. The RWB is cleared to "0" automatically when the stop condition is detected.

• Bits 1 to 7: Slave address (SAD0-SAD6)

These bits store slave addresses. Regardless of the 7-bit addressing mode or the 10-bit addressing mode, the address data transmitted from the master is compared with these bits' contents.

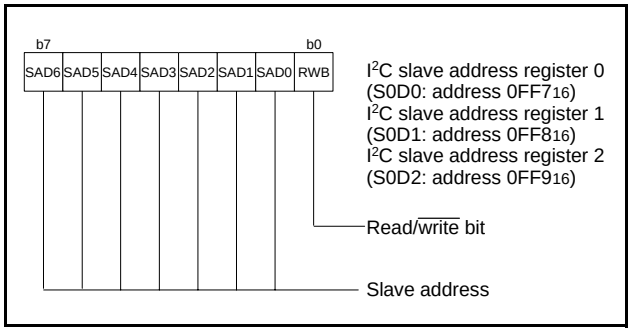


Fig. 61 Structure of I²C slave address registers 0 to 2

[I²C Clock Control Register (S2)] 0015₁₆

The I²C clock control register (S2: address 0015₁₆) is used to set ACK control, SCL mode and SCL frequency.

• Bits 0 to 4: SCL frequency control bits (CCR0-CCR4)

These bits control the SCL frequency. Refer to Table 10.

• Bit 5: SCL mode specification bit (FAST MODE)

This bit specifies the SCL mode. When this bit is set to "0", the standard clock mode is selected. When the bit is set to "1", the high-speed clock mode is selected.

When connecting the bus of the high-speed mode I²C bus standard (maximum 400 kbits/s), use 8 MHz or more oscillation frequency f(XIN) in the high-speed mode (2 division clock).

• Bit 6: ACK bit (ACK BIT)

This bit sets the SDA status when an ACK clock* is generated. When this bit is set to "0", the ACK return mode is selected and SDA goes to "L" at the occurrence of an ACK clock. When the bit is set to "1", the ACK non-return mode is selected. The SDA is held in the "H" status at the occurrence of an ACK clock.

However, when the slave address agree with the address data in the reception of address data at ACK BIT = "0", the SDA is automatically made "L" (ACK is returned). If there is a disagreement between the slave address and the address data, the SDA is automatically made "H" (ACK is not returned).

* ACK clock: Clock for acknowledgment

• Bit 7: ACK clock bit (ACK)

This bit specifies the mode of acknowledgment which is an acknowledgment response of data transfer. When this bit is set to "0", the no ACK clock mode is selected. In this case, no ACK clock occurs after data transmission. When the bit is set to "1", the ACK clock mode is selected and the master generates an ACK clock each completion of each 1-byte data transfer. The device for transmitting address data and control data releases the SDA at the occurrence of an ACK clock (makes SDA "H") and receives the ACK bit generated by the data receiving device.

Note. Do not write data into the I²C clock control register during transfer. If data is written during transfer, the I²C clock generator is reset, so that data cannot be transferred normally.

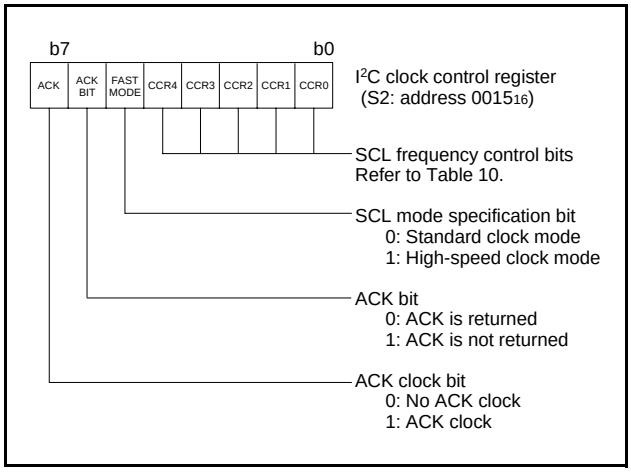


Fig. 62 Structure of I²C clock control register

Table 10 Set values of I²C clock control register and SCL frequency

Setting value of CCR4-CCR0					SCL frequency (at ϕ = 4 MKz, unit: kHz) **(Note 1)**	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard clock mode	High-speed clock mode
0	0	0	0	0	Setting disabled	Setting disabled
0	0	0	0	1	Setting disabled	Setting disabled
0	0	0	1	0	Setting disabled	Setting disabled
0	0	0	1	1	– **(Note 2)**	333
0	0	1	0	0	– **(Note 2)**	250
0	0	1	0	1	100	400 **(Note 3)**
0	0	1	1	0	83.3	166
:	:	:	:	:	500/CCR value **(Note 3)**	1000/CCR value **(Note 3)**
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

NOTES:
1. Duty of SCL output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at ϕ = 4 MHz). "H" duration of the clock fluctuates from -4 to +2 machine cycles in the standard clock mode, and fluctuates from -2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because "L" duration is extended instead of "H" duration reduction. These are values when SCL synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.
2. Each value of SCL frequency exceeds the limit at ϕ = 4 MHz or more. When using these setting value, use ϕ of 4 MHz or less.
3. The data formula of SCL frequency is described below:
   ϕ/(8 × CCR value) Standard clock mode
   ϕ/(4 × CCR value) High-speed clock mode (CCR value ≠ 5)
   ϕ/(2 × CCR value) High-speed clock mode (CCR value = 5)
   Do not set 0 to 2 as CCR value regardless of ϕ frequency. Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

[I²C Control Register (S1D)] 0014₁₆

The I²C control register (S1D: address 0014₁₆) controls data communication format.

• Bits 0 to 2: Bit counter (BC0-BC2)

These bits decide the number of bits for the next 1-byte data to be transmitted. The I²C interrupt request signal occurs immediately after the number of count specified with these bits (ACK clock is added to the number of count when ACK clock is selected by ACK clock bit (bit 7 of S2, address 0015₁₆) have been transferred, and BC0 to BC2 are returned to "000₂".
Also when a START condition is received, these bits become "000₂" and the address data is always transmitted and received in 8 bits.

• Bit 3: I²C interface enable bit (ES0)

This bit enables to use the multi-master I²C-BUS interface. When this bit is set to "0", the use disable status is provided, so that the SDA and the SCL become high-impedance. When the bit is set to "1", use of the interface is enabled.
When ES0 = "0", the following is performed.

- PIN = "1", BB = "0" and AL = "0" are set (which are bits of the I²C status register, S1, at address 0013₁₆).
- Writing data to the I²C data shift register (S0: address 0011₁₆) is disabled.

• Bit 4: Data format selection bit (ALS)

This bit decides whether or not to recognize slave addresses. When this bit is set to "0", the addressing format is selected, so that address data is recognized. When a match is found between a slave address and address data as a result of comparison or when a general call (refer to "I²C Status Register", bit 1) is received, transfer processing can be performed. When this bit is set to "1", the free data format is selected, so that slave addresses are not recognized.

• Bit 5: Addressing format selection bit (10BIT SAD)

This bit selects a slave address specification format. When this bit is set to "0", the 7-bit addressing format is selected. In this case, only the high-order 7 bits (slave address) of the I²C slave address registers 0 to 2 are compared with address data. When this bit is set to "1", the 10-bit addressing format is selected, and all the bits of the I²C slave address registers 0 to 2 are compared with address data.

• Bit 7: I²C-BUS interface pin input level selection bit (TISS)

This bit selects the input level of the SCL and SDA pins of the multi-master I²C-BUS interface.

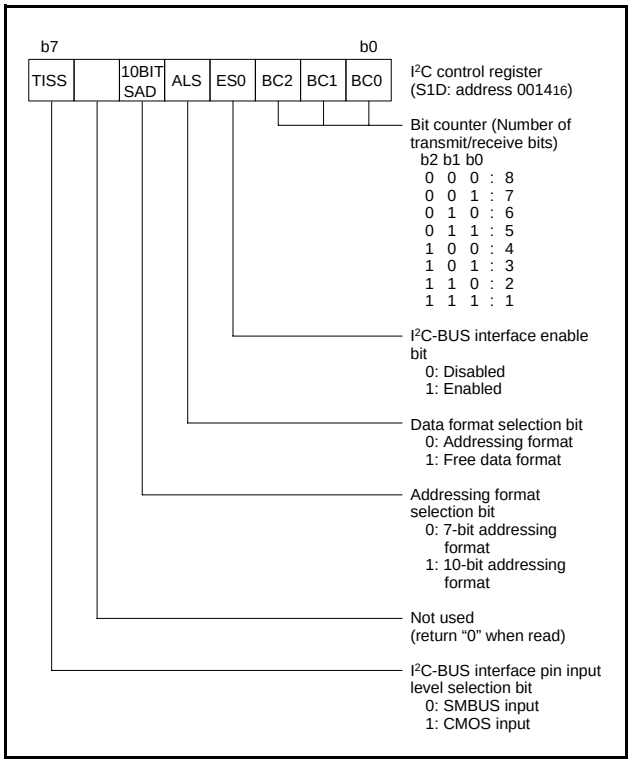


Fig. 63 Structure of I²C clock control register

[I²C Status Register (S1)] 0013₁₆

The I²C status register (S1: address 0013₁₆) controls the I²C-BUS interface status. The low-order 4 bits are read-only bits and the high-order 4 bits can be read out and written to.
Set "0000₂" to the low-order 4 bits, because these bits become the reserved bits at writing.

• Bit 0: Last receive bit (LRB)

This bit stores the last bit value of received data and can also be used for ACK receive confirmation. If ACK is returned when an ACK clock occurs, the LRB bit is set to "0". If ACK is not returned, this bit is set to "1". Except in the ACK mode, the last bit value of received data is input. The state of this bit is changed from "1" to "0" by executing a write instruction to the I²C data shift register (S0: address 0011₁₆).

• Bit 1: General call detecting flag (AD0)

When the ALS bit is "0", this bit is set to "1" when a general call* whose address data is all "0" is received in the slave mode. By a general call of the master device, every slave device receives control data after the general call. The AD0 bit is set to "0" by detecting the STOP condition or START condition, or reset.

* General call: The master transmits the general call address "00₁₆" to all slaves.

• Bit 2: Slave address comparison flag (AAS)

This flag indicates a comparison result of address data when the ALS bit is "0".

(1) In the slave receive mode, when the 7-bit addressing format is selected, this bit is set to "1" in one of the following conditions:
  • The address data immediately after occurrence of a START condition agrees with the slave address stored in the high-order 7 bits of the I²C slave address register.
  • A general call is received.

(2) In the slave receive mode, when the 10-bit addressing format is selected, this bit is set to "1" with the following condition:
  • When the address data is compared with the I²C slave address register (8 bits consisting of slave address and RWB bit), the first bytes agree.

(3) This bit is set to"0" by executing a write instruction to the I²C data shift register (S0: address 0011₁₆) when ES0 is set to "1" or reset.

• Bit 3: Arbitration lost* detecting flag (AL)

In the master transmission mode, when the SDA is made "L" by any other device, arbitration is judged to have been lost, so that this bit is set to "1". At the same time, the TRX bit is set to "0", so that immediately after transmission of the byte whose arbitration was lost is completed, the MST bit is set to "0". The arbitration lost can be detected only in the master transmission mode. When arbitration is lost during slave address transmission, the TRX bit is set to "0" and the reception mode is set. Consequently, it becomes possible to detect the agreement of its own slave address and address data transmitted by another master device.

The AL bit is set to "0" in one of the following conditions:
- Executing a write instruction to the I²C data shift register (S0: address 0011₁₆)
- When the ES0 bit is "0"
- At reset

* Arbitration lost: The status in which communication as a master is disabled.

• Bit 4: SCL pin low hold bit (PIN)

This bit generates an interrupt request signal. Each time 1-byte data is transmitted, the PIN bit changes from "1" to "0". At the same time, an interrupt request signal occurs to the CPU. The PIN bit is set to "0" in synchronization with a falling of the last clock (including the ACK clock) of an internal clock and an interrupt request signal occurs in synchronization with a falling of the PIN bit. When the PIN bit is "0", the SCL is kept in the "0" state and clock generation is disabled. Figure 65 shows an interrupt request signal generating timing chart.
The PIN bit is set to "1" in one of the following conditions:
- Executing a write instruction to the I²C data shift register (S0: address 0011₁₆). (This is the only condition which the prohibition of the internal clock is released and data can be communicated except for the start condition detection.)
- When the ES0 bit is "0"
- At reset
- When writing "1" to the PIN bit by software

The PIN bit is set to "0" in one of the following conditions:
- Immediately after completion of 1-byte data transmission (including when arbitration lost is detected)
- Immediately after completion of 1-byte data reception
- In the slave reception mode, with ALS = "0" and immediately after completion of slave address agreement or general call address reception
- In the slave reception mode, with ALS = "1" and immediately after completion of address data reception

• Bit 5: Bus busy flag (BB)

This bit indicates the status of use of the bus system. When this bit is set to "0", this bus system is not busy and a START condition can be generated. The BB flag is set/reset by the SCL, SDA pins input signal regardless of master/slave. This flag is set to "1" by detecting the START condition, and is set to "0" by detecting the STOP condition. The condition of these detecting is set by the START/STOP condition setting bits (SSC4-SSC0) of the I²C START/STOP condition control register (S2D: address 0016₁₆). When the ES0 bit of the I²C control register (bit 3 of S1D, address 0014₁₆) is "0" or reset, the BB flag is set to "0".
For the writing function to the BB flag, refer to the sections "START Condition Generating Method" and "STOP Condition Generating Method" described later.

• Bit 6: Communication mode specification bit (transfer direction specification bit: TRX)

This bit decides a direction of transfer for data communication. When this bit is “0”, the reception mode is selected and the data of a transmitting device is received. When the bit is “1”, the transmission mode is selected and address data and control data are output onto the SDA in synchronization with the clock generated on the SCL.

This bit is set/reset by software and hardware. About set/reset by hardware is described below. This bit is set to “1” by hardware when all the following conditions are satisfied:

- When ALS is “0”
- In the slave reception mode or the slave transmission mode
- When the R/$\overline{W}$ bit reception is “1”
- This bit is set to “0” in one of the following conditions:
- When arbitration lost is detected.
- When a STOP condition is detected.
- When writing “1” to this bit by software is invalid by the START condition duplication preventing function (Note).
- With MST = “0” and when a START condition is detected.
- With MST = “0” and when ACK non-return is detected.
- At reset

• Bit 7: Communication mode specification bit (master/slave specification bit: MST)

This bit is used for master/slave specification for data communication. When this bit is “0”, the slave is specified, so that a START condition and a STOP condition generated by the master are received, and data communication is performed in synchronization with the clock generated by the master. When this bit is “1”, the master is specified and a START condition and a STOP condition are generated. Additionally, the clocks required for data communication are generated on the SCL.

This bit is set to “0” in one of the following conditions.

- Immediately after completion of the byte which has lost arbitration when arbitration lost is detected
- When a STOP condition is detected.
- Writing “1” to this bit by software is invalid by the START condition duplication preventing function (Note).
- At reset

Note. START condition duplication preventing function
The MST, TRX, and BB bits is set to “1” at the same time after confirming that the BB flag is “0” in the procedure of a START condition occurrence. However, when a START condition by another master device occurs and the BB flag is set to “1” immediately after the contents of the BB flag is confirmed, the START condition duplication preventing function makes the writing to the MST and TRX bits invalid. The duplication preventing function becomes valid from the rising of the BB flag to reception completion of slave address.

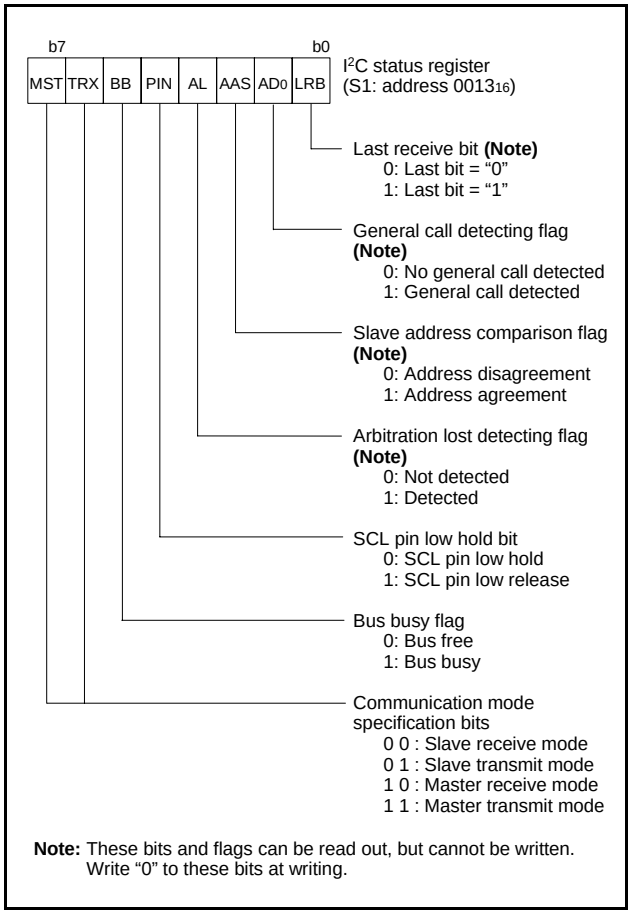


Fig. 64 Structure of I²C status register

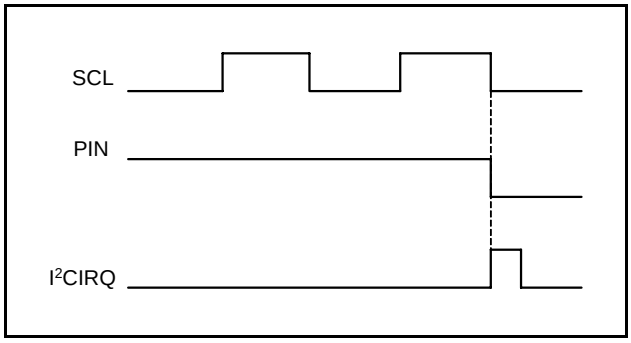


Fig. 65 Interrupt request signal generating timing

START Condition Generating Method

When writing "1" to the MST, TRX, and BB bits of the I²C status register (S1: address 001316) at the same time after writing the slave address to the I²C data shift register (S0: address 001116) with the condition in which the ES0 bit of the I²C control register (S1D: address 001416) is "1" and the BB flag is "0", a START condition occurs. After that, the bit counter becomes "0002" and an SCL for 1 byte is output. The START condition generating timing is different in the standard clock mode and the high-speed clock mode. Refer to Figure 66, the START condition generating timing diagram, and Table 11, the START condition generating timing table.

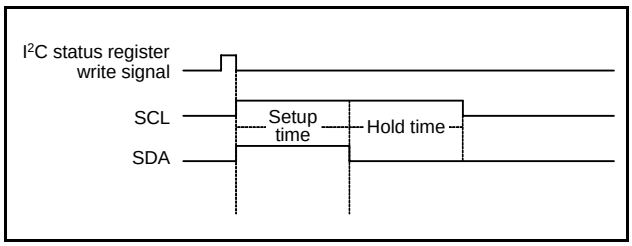


Fig. 66 START condition generating timing diagram

Table 11 START condition generating timing table

Item	Standard clock mode	High-speed clock mode
Setup time	5.0 μs (20 cycles)	2.5 μs (10 cycles)
Hold time	5.0 μs (20 cycles)	2.5 μs (10 cycles)

NOTE:
1. Absolute time at ϕ = 4 MHz. The value in parentheses denotes the number of ϕ cycles.

STOP Condition Generating Method

When the ES0 bit of the I²C control register (S1D: address 001416) is "1", write "1" to the MST and TRX bits, and write "0" to the BB bit of the I²C status register (S1: address 001316) simultaneously. Then a STOP condition occurs. The STOP condition generating timing is different in the standard clock mode and the high-speed clock mode. Refer to Figure 67, the STOP condition generating timing diagram, and Table 12, the STOP condition generating timing table.

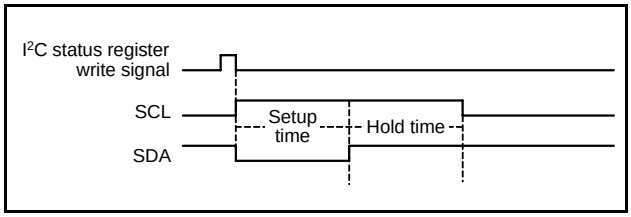


Fig. 67 STOP condition generating timing diagram

Table 12 STOP condition generating timing table

Item	Standard clock mode	High-speed clock mode
Setup time	5.0 μs (20 cycles)	3.0 μs (12 cycles)
Hold time	4.5 μs (18 cycles)	2.5 μs (10 cycles)

NOTE:
1. Absolute time at ϕ = 4 MHz. The value in parentheses denotes the number of ϕ cycles.

START/STOP Condition Detecting Operation

The START/STOP condition detection operations are shown in Figures 68, 69, and Table 13. The START/STOP condition is set by the START/STOP condition set bit.

The START/STOP condition can be detected only when the input signal of the SCL and SDA pins satisfy three conditions: SCL release time, setup time, and hold time (see Table 13).

The BB flag is set to "1" by detecting the START condition and is reset to "0" by detecting the STOP condition.

The BB flag set/reset timing is different in the standard clock mode and the high-speed clock mode. Refer to Table 13, the BB flag set/reset time.

Note. When a STOP condition is detected in the slave mode (MST = 0), an interrupt request signal "I^2CIRQ" occurs to the CPU.

Table 13 START condition/STOP condition detecting conditions

	Standard clock mode	High-speed clock mode
SCL release time	SSC value + 1 cycle (6.25 μs)	4 cycle (1.0 μs)
Setup time	$\frac{\text{SSC value} + 1}{2}$ cycle < 4 μs (3.125 μs)	2 cycle (0.5 μs)
Hold time	$\frac{\text{SSC value} + 1}{2}$ cycle < 4 μs (3.125 μs)	2 cycle (0.5 μs)
BB flag set/ reset time	$\frac{\text{SSC value} - 1}{2}$ + 2 cycles (3.375 μs)	3.5 cycle (0.875 μs)

NOTE:
1. Unit : Cycle number of system clock ϕ
   SSC value is the decimal notation value of the START/STOP condition set bits SSC4 to SSC0. Do not set "0" or an odd number to SSC value. The value in parentheses is an example when the I^2C START/STOP condition control register is set to "18_{16}" at ϕ = 4 MHz.

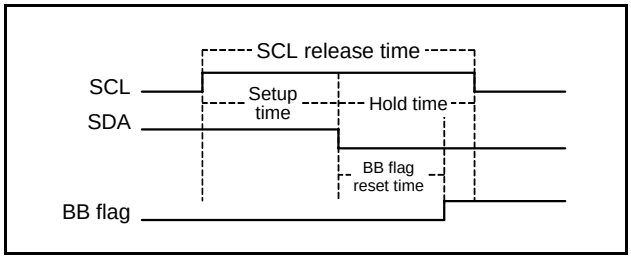


Fig. 68 START/STOP condition detecting timing diagram

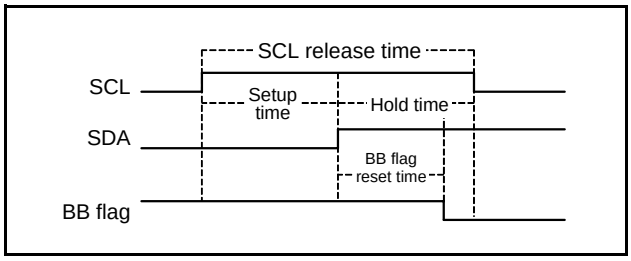


Fig. 69 STOP condition detecting timing diagram

[I²C START/STOP Condition Control Register (S2D)] 0016₁₆

The I²C START/STOP condition control register (S2D: address 0016₁₆) controls START/STOP condition detection.

• Bits 0 to 4: START/STOP condition set bits (SSC4-SSC0)

SCL release time, setup time, and hold time change the detection condition by value of the main clock divide ratio selection bit and the oscillation frequency f(XIN) because these time are measured by the internal system clock. Accordingly, set the proper value to the START/STOP condition set bits (SSC4 to SSC0) in considered of the system clock frequency. Refer to Table 13.
Do not set "00000₂" or an odd number to the START/STOP condition set bits (SSC4 to SSC0).
Refer to Table 14, the recommended set value to START/STOP condition set bits (SSC4-SSC0) for each oscillation frequency.

• Bit 5: SCL/SDA interrupt pin polarity selection bit (SIP)

An interrupt can occur when detecting the falling or rising edge of the SCL or SDA pin. This bit selects the polarity of the SCL or SDA pin interrupt pin.

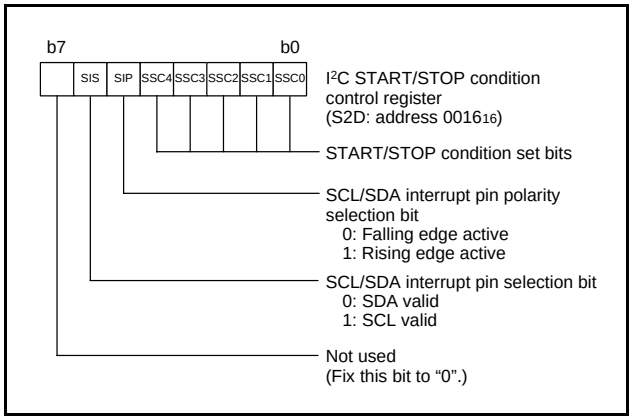


Fig. 70 Structure of I²C START/STOP condition control register

• Bit 6: SCL/SDA interrupt pin selection bit (SIS)

This bit selects the pin of which interrupt becomes valid between the SCL pin and the SDA pin.

Note. When changing the setting of the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0, the SCL/SDA interrupt request bit may be set. When selecting the SCL/SDA interrupt source, disable the interrupt before the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0 is set. Reset the request bit to "0" after setting these bits, and enable the interrupt.

Table 14 Recommended set value to START/STOP condition set bits (SSC4-SSC0) for each oscillation frequency

Oscillation frequency f(XIN)(MHz)	Main clock divide ratio	Internal clock φ (MHz)	START/STOP condition control register	SCL release time (μs)	Setup time (μs)	Hold time (μs)
8	2	4	XXX11010	6.75 μs (27 cycles)	3.5 μs (14 cycles)	3.25 μs (13 cycles)
			XXX11000	6.25 μs (25 cycles)	3.25 μs (13 cycles)	3.0 μs (12 cycles)
8	8	1	XXX00100	5.0 μs (5 cycles)	3.0 μs (3 cycles)	2.0 μs (2 cycles)
4	2	2	XXX01100	6.5 μs (13 cycles)	3.5 μs (7 cycles)	3.0 μs (6 cycles)
			XXX01010	5.5 μs (11 cycles)	3.0 μs (6 cycles)	2.5 μs (5 cycles)
2	2	1	XXX00100	5.0 μs (5 cycles)	3.0 μs (3 cycles)	2.0 μs (2 cycles)

NOTE:
1. Do not set an odd number to the START/STOP condition set bits (SSC4 to SSC0) and "00000₂".

[I²C Special Mode Status Register (S3)] 0012₁₆

The I²C special mode status register (S3: address 0012₁₆) consists of the flags indicating I²C operating state in the I²C special mode, which is set by the I²C special mode control register (S3D: address 0017₁₆).
The stop condition flag is valid in all operating modes.

• Bit 0: Slave address 0 comparison flag (AAS0)
Bit 1: Slave address 1 comparison flag (AAS1)
Bit 2: Slave address 2 comparison flag (AAS2)

These flags indicate a comparison result of address data. These flags are valid only when the slave address control bit (MSLAD) is "1".
In the 7-bit addressing format of the slave reception mode, the respective slave address i (i = 0, 1, 2) comparison flags corresponding to the I²C slave address registers 0 to 2 are set to "1" when an address data immediately after an occurrence of a START condition agrees with the high-order 7-bit slave address stored in the I²C slave address registers 0 to 2 (addresses 0FF7₁₆ to 0FF9₁₆).
In the 10-bit addressing format of the slave mode, the respective slave address i (i = 0, 1, 2) comparison flags corresponding to the I²C slave address registers are set to "1" when an address data is compared with the 8 bits consisting of the slave address stored in the I²C slave address registers 0 to 2 and the RWB bit, and the first byte agrees.
These flags are initialized to "0" at reset, when the slave address control bit (MSLAD) is "0", or when writing data to the I²C data shift register (S0: address 0011₁₆).

• Bit 5: SCL pin low hold 2 flag (PIN2)

When the ACK interrupt control bit (ACKICON) and the ACK clock bit (ACK) are "1", this flag is set to "0" in synchronization with the falling of the data's last SCL clock, just before the ACK clock. The SCL pin is simultaneously held low, and the I²C interrupt request occurs.
This flag is initialized to "1" at reset, when the ACK interrupt control bit (ACKICON) is "0", or when writing "1" to the SCL pin low hold 2 flag set bit (PIN2IN).
The SCL pin is held low when either the SCL pin low hold bit (PIN) or the SCL pin low hold 2 flag (PIN2) becomes "0". The low hold state of the SCL pin is released when both the SCL pin low hold bit (PIN) and the SCL pin low hold 2 flag (PIN2) are "1".

• Bit 7: Stop condition flag (SPCF)

This flag is set to "1" when a STOP condition occurs.
This flag is initialized to "0" at reset, when the I²C-BUS interface enable bit (ES0) is "0", or when writing "1" to the STOP condition flag clear bit (SPFCL).

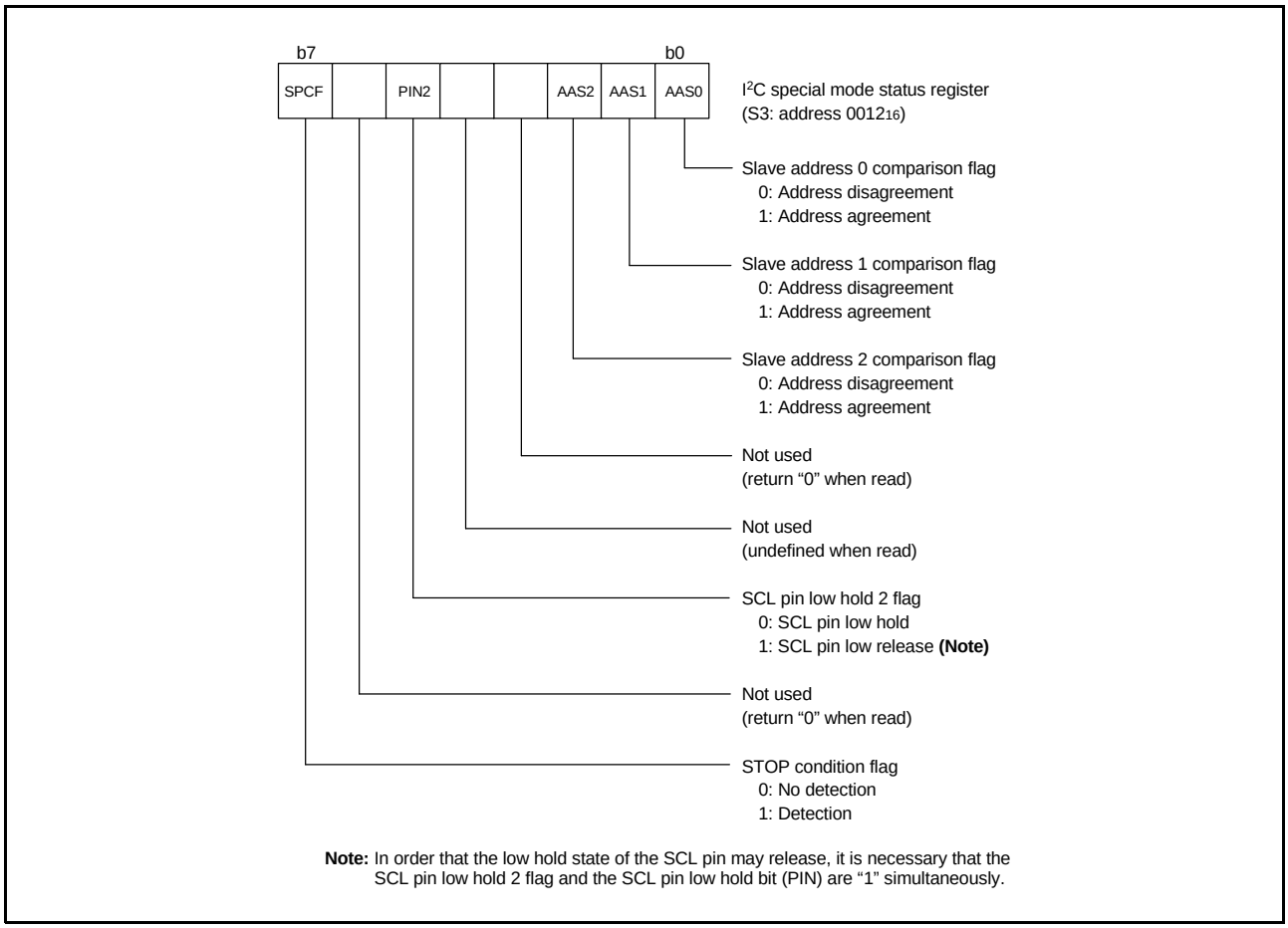


Fig. 71 Structure of I²C special mode status register

[I²C Special Mode Control Register (S3D)] 0017₁₆

The I²C special mode control register (S3D: address 0017₁₆) controls special functions such as occurrence timing of reception interrupt request and extending slave address comparison to 3 bytes.

• Bit 1: ACK interrupt control bit (ACKICON)

This bit controls the timing of I²C interrupt request occurrence at completion of data receiving due to master reception or slave reception.
When this bit is "0", the SCL pin low hold bit (PIN) is set to "0" in synchronization with the falling of the last SCL clock, including the ACK clock. The SCL pin is simultaneously held low, and the I²C interrupt request occurs.
When this bit is "1" and the ACK clock bit (ACK) is "1", the SCL pin low hold 2 flag (PIN2) is set to "0" in synchronization with the falling of the data's last SCL clock, just before the ACK clock. The SCL pin is simultaneously held low, and the I²C interrupt request occurs again. The ACK bit can be changed after the contents of data are confirmed by using this function.

• Bit 2: I²C slave address control bit (MSLAD)

This bit controls a slave address. When this bit is "0", only the I²C slave address register 0 (address 0FF7₁₆) becomes valid as a slave address and a read/write bit.
When this bit is "1", all of the I²C slave address registers 0 to 2 (addresses 0FF7₁₆ to 0FF9₁₆) become valid as a slave address and a read/write bit. In this case, when an address data agrees with any one of the I²C slave address registers 0 to 2, the slave address comparison flag (AAS) is set to "1" and the I²C slave address comparison flag corresponding to the agreed I²C slave address registers 0 to 2 is also set to "1".

• Bit 5: SCL pin low hold 2 flag set bit (PIN2IN)

Writing "1" to this bit initializes the SCL pin low hold 2 flag (PIN2) to "1".
When writing "0", nothing is generated.

• Bit 6: SCL pin low hold set bit (PIN2HD)

When the SCL pin low hold bit (PIN) becomes "0", the SCL pin is held low. However, the SCL pin low hold bit (PIN) cannot be set to "0" by software. The SCL pin low hold set bit (PIN2HD) is used to, hold the SCL pin in the low state by software. When writing "1" to this bit, the SCL pin low hold 2 flag (PIN2) becomes "0", and the SCL pin is held low. When writing "0", nothing occurs.

• Bit 7: STOP condition flag clear bit (SPFCL)

Writing "1" to this bit initializes the STOP condition flag (SPCF) to "0".
When writing "0", nothing is generated.

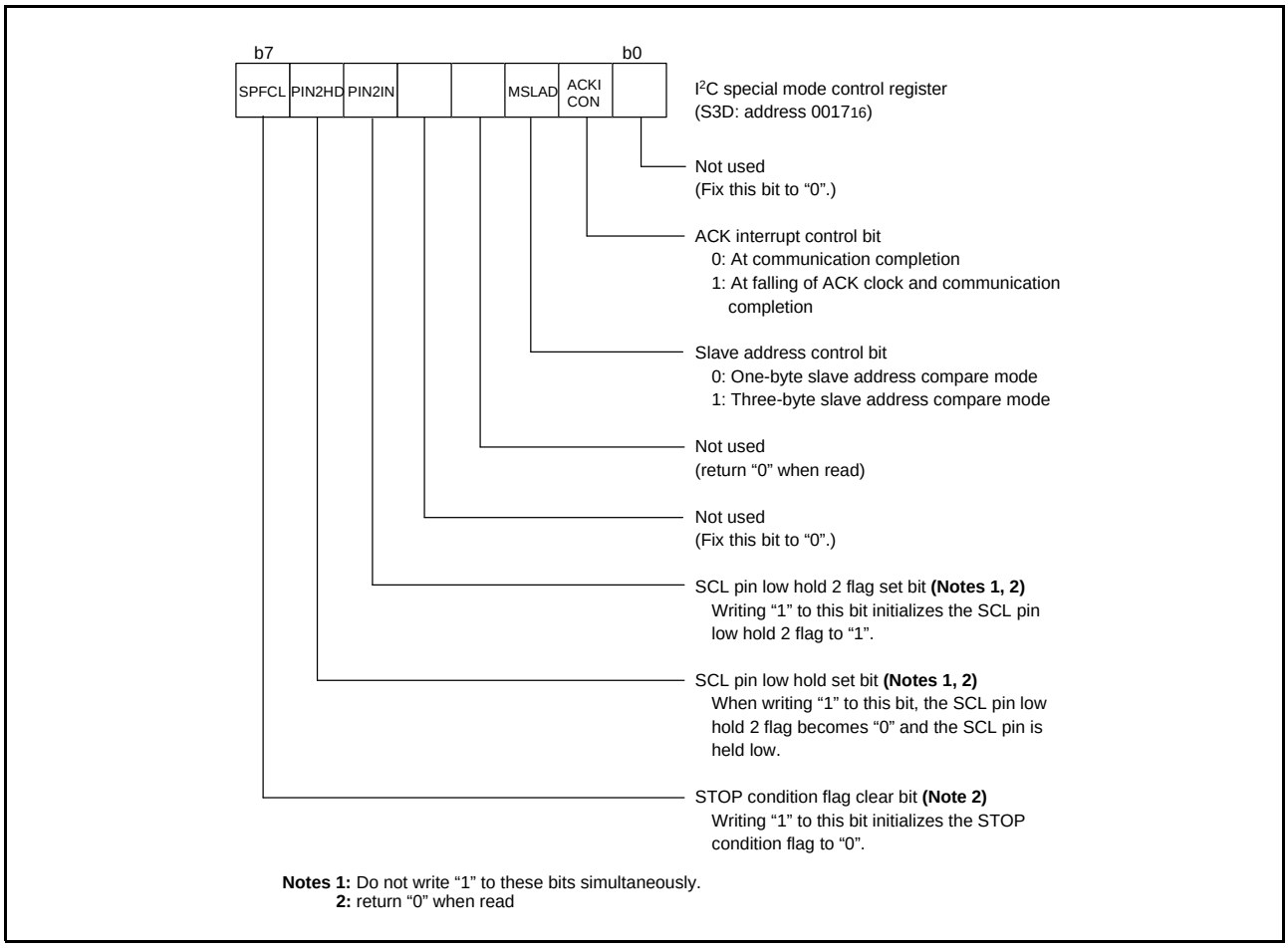


Fig. 72 Structure of I²C special mode control register

Address Data Communication

There are two address data communication formats, namely, 7-bit addressing format and 10-bit addressing format. The respective address communication formats are described below.

• 7-bit addressing format

To adapt the 7-bit addressing format, set the 10BIT SAD bit of the I²C control register (S1D: address 0014₁₆) to "0". The first 7-bit address data transmitted from the master is compared with the high-order 7-bit slave address stored in the I²C slave address register. At the time of this comparison, address comparison of the RWB bit of the I²C slave address register is not performed. For the data transmission format when the 7-bit addressing format is selected, refer to Figure 73, (1) and (2).

• 10-bit addressing format

To adapt the 10-bit addressing format, set the 10BIT SAD bit of the I²C control register (S1D: address 0014₁₆) to "1". An address comparison is performed between the first-byte address data transmitted from the master and the 8-bit slave address stored in the I²C slave address register. At the time of this comparison, an address comparison between the RWB bit of the I²C slave address register and the R/$\overline{W}$ bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the RWB bit which is the last bit of the address data not only specifies the direction of communication for control data, but also is processed as an address data bit.

When the first-byte address data agree with the slave address, the AAS bit of the I²C status register (S1: address 0013₁₆) is set to "1". After the second-byte address data is stored into the I²C data shift register (S0: address 0011₁₆), perform an address comparison between the second-byte data and the slave address by software. When the address data of the 2 bytes agree with the slave address, set the RWB bit of the I²C slave address register to "1" by software. This processing can make the 7-bit slave address and R/$\overline{W}$ data agree, which are received after a RESTART condition is detected, with the value of the I²C slave address register. For the data transmission format when the 10-bit addressing format is selected, refer to Figure 73, (3) and (4)
.

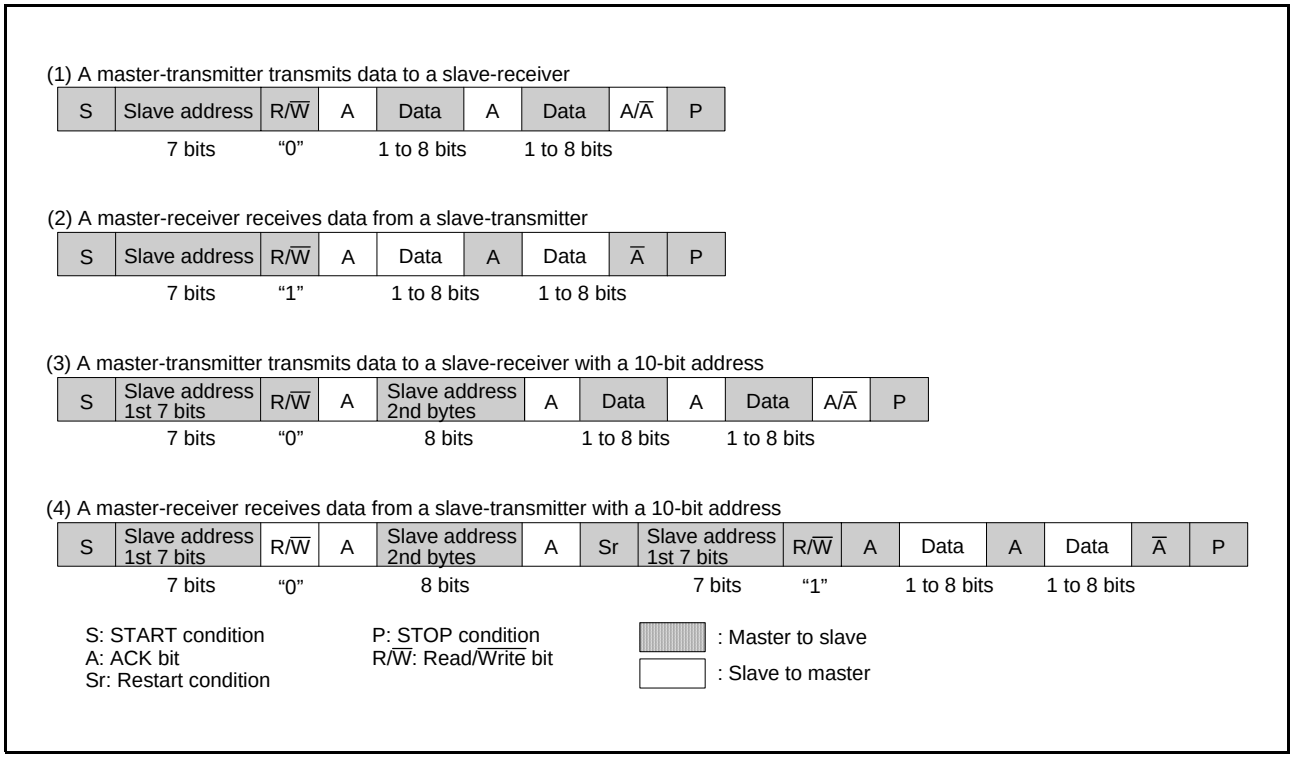


Fig. 73 Address data communication format

Example of Master Transmission

An example of master transmission in the standard clock mode, at the SCL frequency of 100 kHz and in the ACK return mode is shown below.

(1) Set a slave address in the high-order 7 bits of the I²C slave address register and "0" into the RWB bit.
(2) Set the ACK return mode and SCL = 100 kHz by setting "85₁₆" in the I²C clock control register (S2: address 0015₁₆).
(3) Set "00₁₆" in the I²C status register (S1: address 0013₁₆) so that transmission/reception mode can become initializing condition.
(4) Set a communication enable status by setting "08₁₆" in the I²C control register (S1D: address 0014₁₆).
(5) Confirm the bus free condition by the BB flag of the I²C status register (S1: address 0013₁₆).
(6) Set the address data of the destination of transmission in the high-order 7 bits of the I²C data shift register (S0: address 0011₁₆) and set "0" in the least significant bit.
(7) Set "F0₁₆" in the I²C status register (S1: address 0013₁₆) to generate a START condition. At this time, an SCL for 1 byte and an ACK clock automatically occur.
(8) Set transmit data in the I²C data shift register (S0: address 0011₁₆). At this time, an SCL and an ACK clock automatically occur.
(9) When transmitting control data of more than 1 byte, repeat step (8).
(10) Set "D0₁₆" in the I²C status register (S1: address 0013₁₆) to generate a STOP condition if ACK is not returned from slave reception side or transmission ends.

Example of Slave Reception

An example of slave reception in the high-speed clock mode, at the SCL frequency of 400 kHz, in the ACK non-return mode and using the addressing format is shown below.

(1) Set a slave address in the high-order 7 bits of the I²C slave address register and "0" in the RWB bit.
(2) Set the no ACK clock mode and SCL = 400 kHz by setting "25₁₆" in the I²C clock control register (S2: address 0015₁₆).
(3) Set "00₁₆" in the I²C status register (S1: address 0013₁₆) so that transmission/reception mode can become initializing condition.
(4) Set a communication enable status by setting "08₁₆" in the I²C control register (S1D: address 0014₁₆).
(5) When a START condition is received, an address comparison is performed.
(6)
- When all transmitted addresses are "0" (general call):
  AD0 of the I²C status register (S1: address 0013₁₆) is set to "1" and an interrupt request signal occurs.
- When the transmitted addresses agree with the address set in (1):
  AAS of the I²C status register (S1: address 0013₁₆) is set to "1" and an interrupt request signal occurs.
- In the cases other than the above AD0 and AAS of the I²C status register (S1: address 0013₁₆) are set to "0" and no interrupt request signal occurs.

(7) Set dummy data in the I²C data shift register (S0: address 0011₁₆).
(8) When receiving control data of more than 1 byte, repeat step (7).
(9) When a STOP condition is detected, the communication ends.

Precautions when using multi-master I²C BUS interface

(1) Read-modify-write instruction

The precautions when the read-modify-write instruction such as SEB, CLB etc. is executed for each register of the multi-master I²C-BUS interface are described below.

- I²C data shift register (S0: address 0011₁₆)
  When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.
- I²C slave address registers 0 to 2 (S0D0 to S0D2: addresses 0FF7₁₆ to0FF9₁₆)
  When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended. It is because H/W changes the read/write bit (RWB) at the above timing.
- I²C status register (S1: address 0013₁₆)
  Do not execute the read-modify-write instruction for this register because all bits of this register are changed by H/W.
- I²C control register (S1D: address 0014₁₆)
  When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended. Because H/W changes the bit counter (BC0-BC2) at the above timing.
- I²C clock control register (S2: address 0015₁₆)
  The read-modify-write instruction can be executed for this register.
- I²C START/STOP condition control register (S2D: address 0016₁₆)
  The read-modify-write instruction can be executed for this register.

(2) START condition generating procedure using multi-master

1. Procedure example (The necessary conditions of the generating procedure are described as the following 2 to 5.

```
    :
  LDA –                 (Taking out of slave address value)
  SEI                   (Interrupt disabled)
  BBS 5, S1, BUSBUSY    (BB flag confirming and branch process)
BUSFREE:
  STA S0                (Writing of slave address value)
  LDM #$F0, S1          (Trigger of START condition
  generating)
  CLI                   (Interrupt enabled)
    :
BUSBUSY:
  CLI                   (Interrupt enabled)
    :
```

2. Use “Branch on Bit Set” of “BBS 5, S1, -” for the BB flag confirming and branch process.
3. Use “STA $12, STX $12” or “STY $12” of the zero page addressing instruction for writing the slave address value to the I²C data shift register.
4. Execute the branch instruction of above 2 and the store instruction of above 3 continuously shown the above procedure example.
5. Disable interrupts during the following three process steps:
   • BB flag confirming
   • Writing of slave address value
   • Trigger of START condition generating
   When the condition of the BB flag is bus busy, enable interrupts immediately.

(3) RESTART condition generating procedure

1. Procedure example (The necessary conditions of the generating procedure are described as the following 2 to 4.)
   Execute the following procedure when the PIN bit is “0”

```
    :
  LDM #$00, S1      (Select slave receive mode)
  LDA –             (Taking out of slave address value)
  SEI               (Interrupt disabled)
  STA S0            (Writing of slave address value)
  LDM #$F0, S1      (Trigger of RESTART condition generating)
  CLI               (Interrupt enabled)
    :
```

2. Select the slave receive mode when the PIN bit is “0”. Do not write “1” to the PIN bit. Neither “0” nor “1” is specified for the writing to the BB bit.
   The TRX bit becomes “0” and the SDA pin is released.
3. The SCL pin is released by writing the slave address value to the I²C data shift register.
4. Disable interrupts during the following two process steps:
   • Writing of slave address value
   • Trigger of RESTART condition generating

(4) Writing to I²C status register

Do not execute an instruction to set the PIN bit to “1” from “0” and an instruction to set the MST and TRX bits to “0” from “1” simultaneously. It is because it may enter the state that the SCL pin is released and the SDA pin is released after about one machine cycle. Do not execute an instruction to set the MST and TRX bits to “0” from “1” simultaneously when the PIN bit is “1”. It is because it may become the same as above.

(5) Process of after STOP condition generating

Do not write data in the I²C data shift register S0 and the I²C status register S1 until the bus busy flag BB becomes “0” after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers does not have the problem.

RESET CIRCUIT

To reset the microcomputer, $\overline{\text{RESET}}$ pin should be held at an "L" level for 16 cycles or more of XIN. Then the $\overline{\text{RESET}}$ pin is returned to an "H" level (the power source voltage should be between 2.7 V and 5.5 V, and the oscillation should be stable), reset is released. After the reset is completed, the program starts from the address contained in address FFFD16 (high-order byte) and address FFFC16 (low-order byte).
Input to the $\overline{\text{RESET}}$ pin in the following procedure.

- When power source is stabilized

(1) Input "L" level to $\overline{\text{RESET}}$ pin.
(2) Input "L" level for 16 cycles or more to XIN pin.
(3) Input "H" level to $\overline{\text{RESET}}$ pin.

- At power-on

(1) Input "L" level to $\overline{\text{RESET}}$ pin.
(2) Increase the power source voltage to 2.7 V.
(3) Wait for td(P-R) until internal power source has stabilized.
(4) Input "L" level for 16 cycles or more to XIN pin.
(5) Input "H" level to $\overline{\text{RESET}}$ pin.

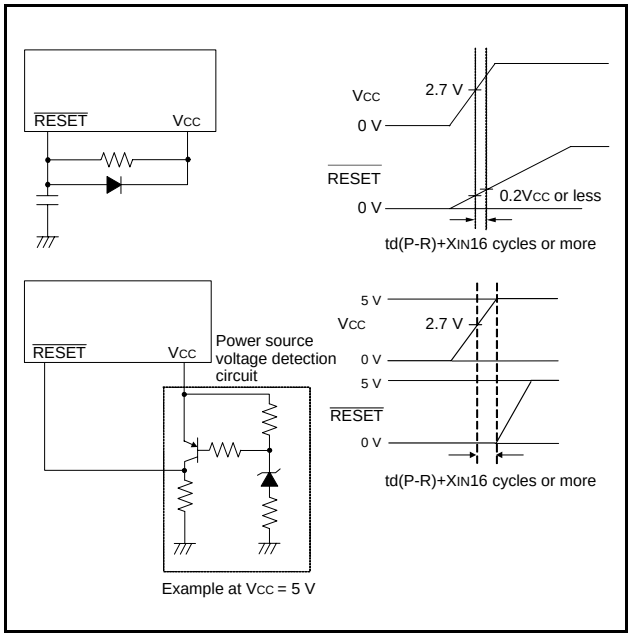
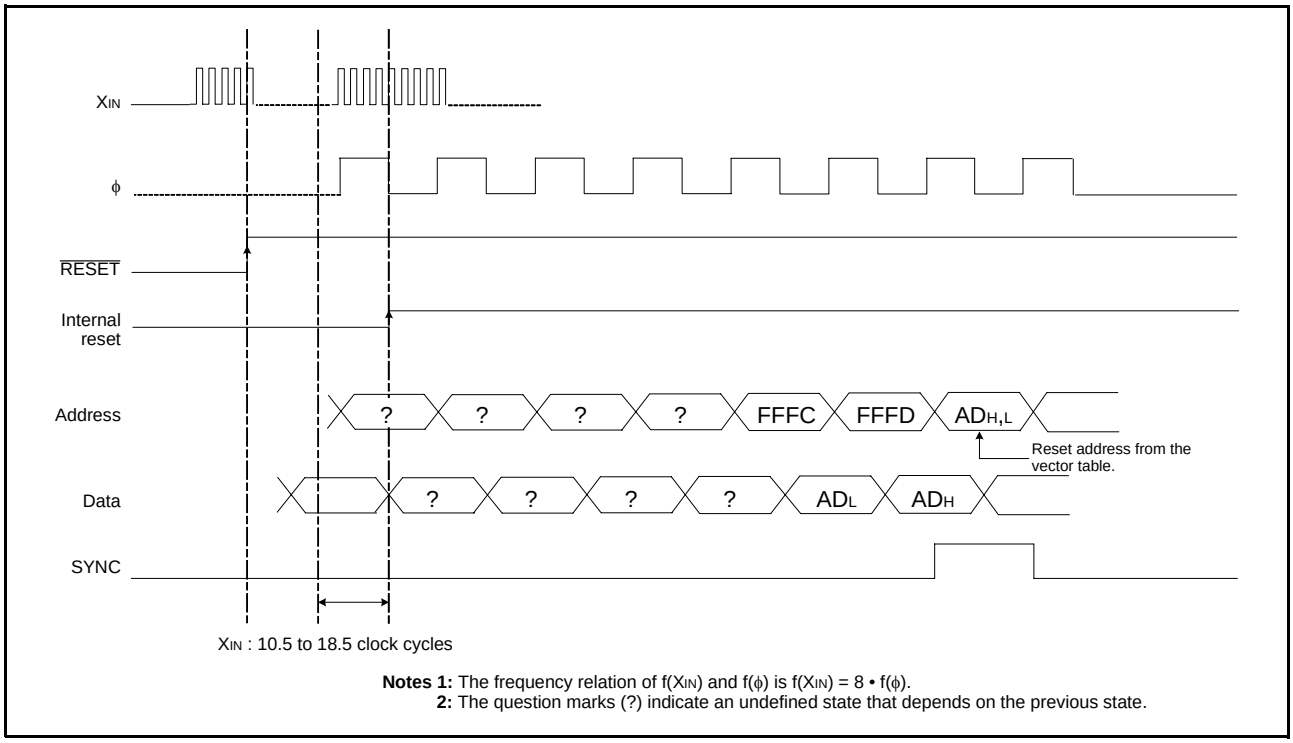


Fig. 74 Reset circuit example



Notes 1: The frequency relation of f(XIN) and f(ϕ) is f(XIN) = 8 • f(ϕ).
2: The question marks (?) indicate an undefined state that depends on the previous state.

Fig. 75 Reset sequence

		Address	Register contents
(1)	Port P0 (P0)	0000_{16}	00_{16}
(2)	Port P0 direction register (P0D)	0001_{16}	00_{16}
(3)	Port P1 (P1)	0002_{16}	00_{16}
(4)	Port P1 direction register (P1D)	0003_{16}	00_{16}
(5)	Port P2 (P2)	0004_{16}	00_{16}
(6)	Port P2 direction register (P2D)	0005_{16}	00_{16}
(7)	Port P3 (P3)	0006_{16}	00_{16}
(8)	Port P3 direction register (P3D)	0007_{16}	00_{16}
(9)	Port P4 (P4)	0008_{16}	00_{16}
(10)	Port P4 direction register (P4D)	0009_{16}	00_{16}
(11)	Port P5 (P5)	$000A_{16}$	00_{16}
(12)	Port P5 direction register (P5D)	$000B_{16}$	00_{16}
(13)	Port P6 (P6)	$000C_{16}$	00_{16}
(14)	Port P6 direction register (P6D)	$000D_{16}$	00_{16}
(15)	Timer 12, X count source selection register (T12XCSS)	$000E_{16}$	0 0 1 1 0 0 1 1
(16)	Timer Y, Z count source selection register (TYZCSS)	$000F_{16}$	0 0 1 1 0 0 1 1
(17)	MISRG	0010_{16}	00_{16}
(18)	I^2C data shift register (S0)	0011_{16}	X X X X X X X X
(19)	I^2C special mode status register (S3)	0012_{16}	0 0 1 0 0 0 0 0
(20)	I^2C status register (S1)	0013_{16}	0 0 0 1 0 0 0 X
(21)	I^2C control register (S1D)	0014_{16}	00_{16}
(22)	I^2C clock control register (S2)	0015_{16}	00_{16}
(23)	I^2C START/STOP condition control register (S2D)	0016_{16}	0 0 0 1 1 0 1 0
(24)	I^2C special mode control register (S3D)	0017_{16}	00_{16}
(25)	Transmit/Receive buffer register 1 (TB1/RB1)	0018_{16}	X X X X X X X X
(26)	Serial I/O1 status register (SIO1STS)	0019_{16}	1 0 0 0 0 0 0 0
(27)	Serial I/O1 control register (SIO1CON)	$001A_{16}$	00_{16}
(28)	UART1 control register (UART1CON)	$001B_{16}$	1 1 1 0 0 0 0 0
(29)	Baud rate generator 1 (BRG1)	$001C_{16}$	X X X X X X X X
(30)	Serial I/O2 control register (SIO2CON)	$001D_{16}$	00_{16}
(31)	Watchdog timer control register (WDTCON)	$001E_{16}$	0 0 1 1 1 1 1 1
(32)	Serial I/O2 register (SIO2)	$001F_{16}$	X X X X X X X X
(33)	Prescaler 12 (PRE12)	0020_{16}	FF_{16}
(34)	Timer 1 (T1)	0021_{16}	01_{16}
(35)	Timer 2 (T2)	0022_{16}	FF_{16}
(36)	Timer XY mode register (TM)	0023_{16}	00_{16}
(37)	Prescaler X (PREX)	0024_{16}	FF_{16}
(38)	Timer X (TX)	0025_{16}	FF_{16}
(39)	Prescaler Y (PREY)	0026_{16}	FF_{16}
(40)	Timer Y (TY)	0027_{16}	FF_{16}

		Address	Register contents
(41)	Timer Z (low-order) (TZL)	0028_{16}	FF_{16}
(42)	Timer Z (high-order) (TZH)	0029_{16}	FF_{16}
(43)	Timer Z mode register (TZM)	$002A_{16}$	00_{16}
(44)	PWM control register (PWMCON)	$002B_{16}$	00_{16}
(45)	PWM prescaler (PREPWM)	$002C_{16}$	X X X X X X X X
(46)	PWM register (PWM)	$002D_{16}$	X X X X X X X X
(47)	Baud rate generator 3 (BRG3)	$002F_{16}$	X X X X X X X X
(48)	Transmit/Receive buffer register 3 (TB3/RB3)	0030_{16}	X X X X X X X X
(49)	Serial I/O3 status register (SIO3STS)	0031_{16}	1 0 0 0 0 0 0 0
(50)	Serial I/O3 control register (SIO3CON)	0032_{16}	00_{16}
(51)	UART3 control register (UART3CON)	0033_{16}	1 1 1 0 0 0 0 0
(52)	AD/DA control register (ADCON)	0034_{16}	0 0 0 0 1 0 0 0
(53)	AD conversion register 1 (AD1)	0035_{16}	X X X X X X X X
(54)	DA1 conversion register (DA1)	0036_{16}	00_{16}
(55)	DA2 conversion register (DA2)	0037_{16}	00_{16}
(56)	AD conversion register 2 (AD2)	0038_{16}	0 0 0 0 0 0 X X
(57)	Interrupt source selection register (INTSEL)	0039_{16}	00_{16}
(58)	Interrupt edge selection register (INTEDGE)	$003A_{16}$	00_{16}
(59)	CPU mode register (CPUM)	$003B_{16}$	0 1 0 0 1 0 0 0
(60)	Interrupt request register 1 (IREQ1)	$003C_{16}$	00_{16}
(61)	Interrupt request register 2 (IREQ2)	$003D_{16}$	00_{16}
(62)	Interrupt control register 1 (ICON1)	$003E_{16}$	00_{16}
(63)	Interrupt control register 2 (ICON2)	$003F_{16}$	00_{16}
(64)	Flash memory control register 0 (FMCR0)	$0FE0_{16}$	0 0 0 0 0 0 0 1
(65)	Flash memory control register 1 (FMCR1)	$0FE1_{16}$	0 1 0 0 0 0 0 0
(66)	Flash memory control register 2 (FMCR2)	$0FE2_{16}$	0 1 0 0 0 1 0 1
(67)	Port P0 pull-up control register (PULL0)	$0FF0_{16}$	00_{16}
(68)	Port P1 pull-up control register (PULL1)	$0FF1_{16}$	00_{16}
(69)	Port P2 pull-up control register (PULL2)	$0FF2_{16}$	00_{16}
(70)	Port P3 pull-up control register (PULL3)	$0FF3_{16}$	00_{16}
(71)	Port P4 pull-up control register (PULL4)	$0FF4_{16}$	00_{16}
(72)	Port P5 pull-up control register (PULL5)	$0FF5_{16}$	00_{16}
(73)	Port P6 pull-up control register (PULL6)	$0FF6_{16}$	00_{16}
(74)	I^2C slave address register 0 (S0D0)	$0FF7_{16}$	00_{16}
(75)	I^2C slave address register 1 (S0D1)	$0FF8_{16}$	00_{16}
(76)	I^2C slave address register 2 (S0D2)	$0FF9_{16}$	00_{16}
(77)	Processor status register	(PS)	X X X X X 1 X X
(78)	Program counter	(PC_H)	$FFFD_{16}$ contents
		(PC_L)	$FFFC_{16}$ contents

Note : X: Not fixed.
Since the initial values for other than above mentioned registers and RAM contents are indefinite at reset, they must be set.

Fig. 76 Internal status at reset

CLOCK GENERATING CIRCUIT

The 3804 group (Spec.L) has two built-in oscillation circuits: main clock XIN-XOUT oscillation circuit and sub clock XCIN-XCOUT oscillation circuit. An oscillation circuit can be formed by connecting a resonator between XIN and XOUT (XCIN and XCOUT). 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.) However, an external feed-back resistor is needed between XCIN and XCOUT.
Immediately after power on, only the XIN oscillation circuit starts oscillating, and XCIN and XCOUT pins function as I/O ports.

• Frequency Control

(1) Middle-speed mode

The internal clock ϕ is the frequency of XIN divided by 8. After reset is released, this mode is selected.

(2) High-speed mode

The internal clock ϕ is half the frequency of XIN.

(3) Low-speed mode

The internal clock ϕ is half the frequency of XCIN.

(4) Low power dissipation mode

The low power consumption operation can be realized by stopping the main clock XIN in low-speed mode. To stop the main clock, set bit 5 of the CPU mode register to "1". When the main clock XIN is restarted (by setting the main clock stop bit to "0"), set sufficient time for oscillation to stabilize.
The sub-clock XCIN-XCOUT oscillating circuit can not directly input clocks that are generated externally. Accordingly, make sure to cause an external resonator to oscillate.

Oscillation Control

(1) Stop mode

If the STP instruction is executed, the internal clock ϕ stops at an "H" level, and XIN and XCIN oscillators stop. When the oscillation stabilizing time set after STP instruction released bit (bit 0 of address 0010_{16}) is "0", the prescaler 12 is set to "FF_{16}" and timer 1 is set to "01_{16}". When the oscillation stabilizing time set after STP instruction released bit is "1", set the sufficient time for oscillation of used oscillator to stabilize since nothing is set to the prescaler 12 and timer 1.
After STP instruction is released, the input of the prescaler 12 is connected to count source which had set at executing the STP instruction, and the output of the prescaler 12 is connected to timer 1. Oscillator restarts when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU (remains at "H") until timer 1 underflows. The internal clock ϕ is supplied for the first time, when timer 1 underflows. This ensures time for the clock oscillation using the ceramic resonators to be stabilized. When the oscillator is restarted by reset, apply "L" level to the $\overline{\text{RESET}}$ pin until the oscillation is stable since a wait time will not be generated.
In the flash memory L version, the built-in power source circuit is switched to the low power dissipation mode at executing the STP instruction to reduce consumption current. At returning from the STP instruction, the built-in power source circuit is switched to the normal mode, but a specified time is required from when the power supply to the flash memory is started until the flash memory operation is enabled. In this version, set a wait time of 100 μs or more with the oscillation stabilizing time set after STP instruction released function by using timer 1.

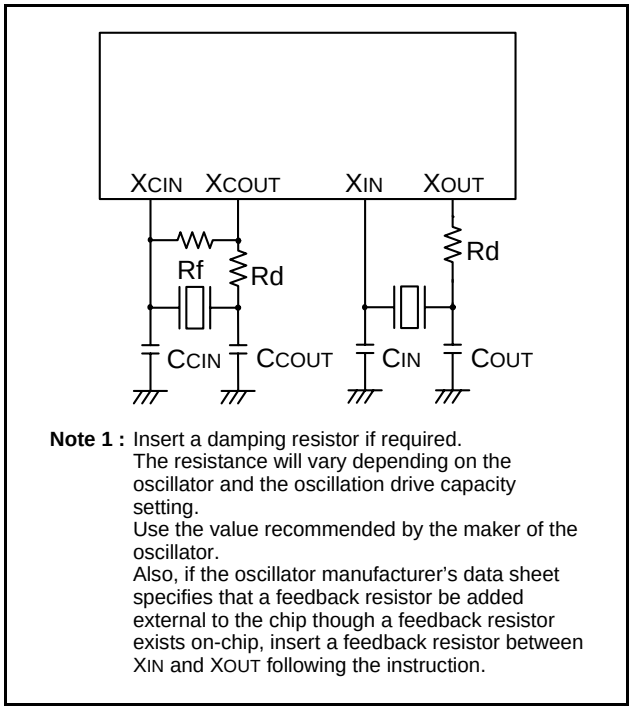
(2) Wait mode

If the WIT instruction is executed, the internal clock ϕ stops at an "H" level, but the oscillator does not stop. The internal clock ϕ restarts at reset 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 the interrupts will be received to release the STP or WIT state, their interrupt enable bits must be set to "1" before executing of the STP or WIT instruction.
When releasing the STP state, the input of the prescaler 12 and timer 1 is connected to the count source which had set at executing the STP instruction and the prescaler 12 and timer 1 will start counting. Set the timer 1 interrupt enable bit to "0" before executing the STP instruction.

<Notes>

- If you switch the mode between middle/high-speed and low-speed, stabilize both XIN and XCIN oscillations. The sufficient time is required for the sub clock to stabilize, especially immediately after power on and at returning from stop mode. When switching the mode between middle/high-speed and low-speed, set the frequency on condition that f(XIN) > 3×f(XCIN).
- When using the quartz-crystal oscillator of high frequency, such as 16 MHz etc., it may be necessary to select a specific oscillator with the specification demanded.
- When using the oscillation stabilizing time set after STP instruction released bit set to "1", evaluate time to stabilize oscillation of the used oscillator and set the value to the timer 1 and prescaler 12.



Note 1 : 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. 77 Ceramic resonator circuit

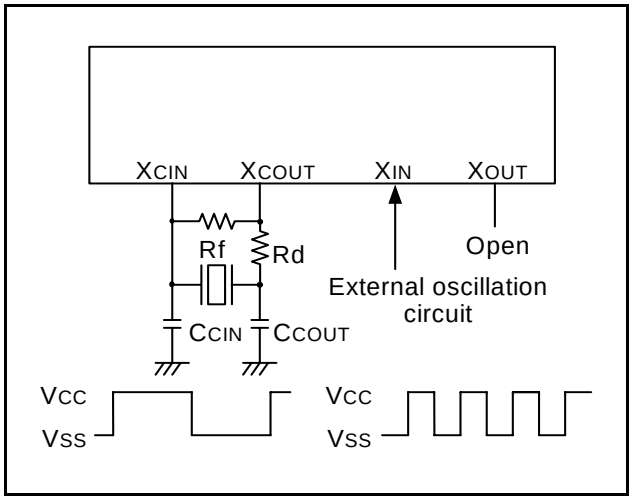
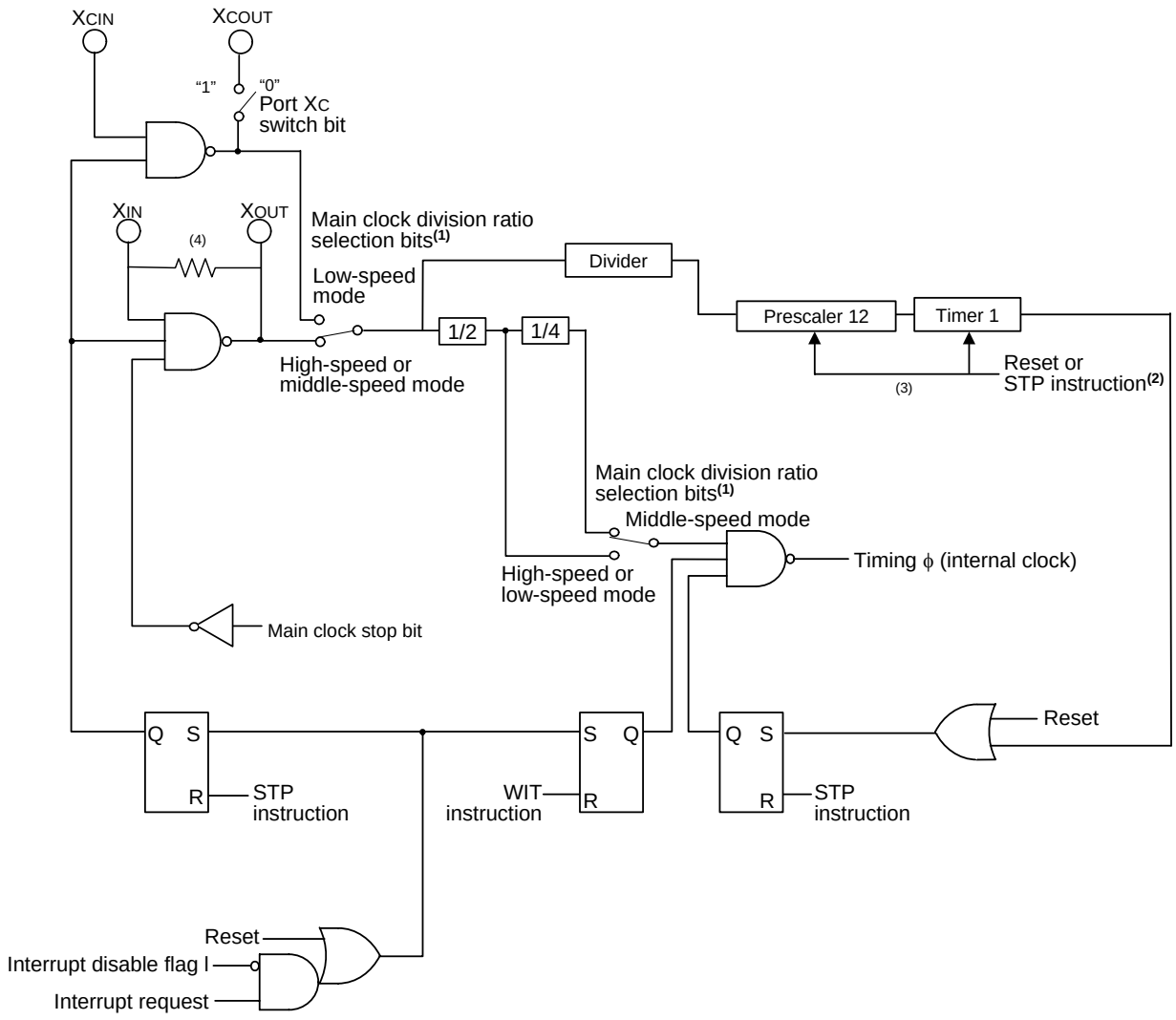
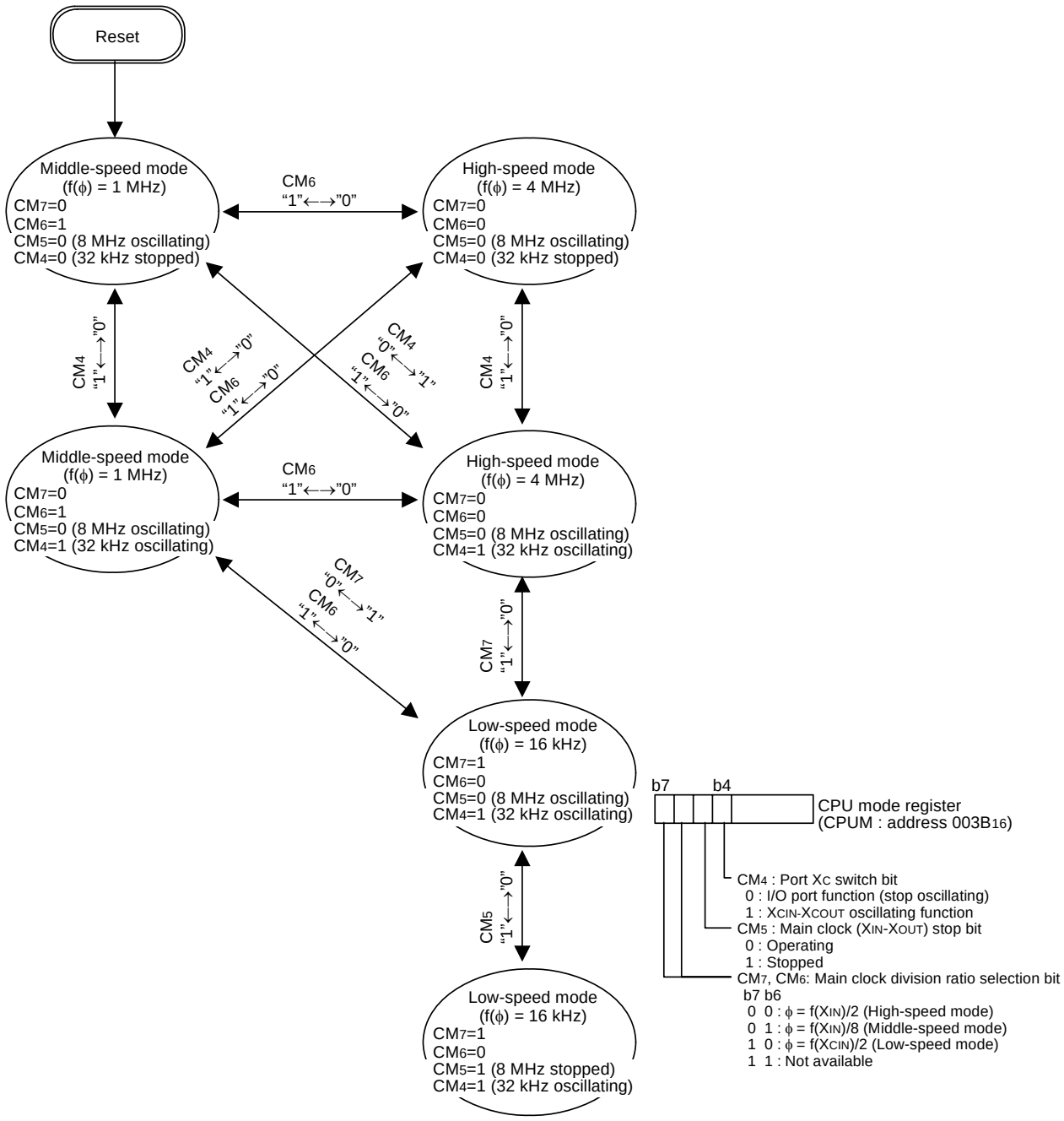


Fig. 78 External clock input circuit



Notes 1: Either high-speed, middle-speed or low-speed mode is selected by bits 7 and 6 of the CPU mode register. When low-speed mode is selected, set port XC switch bit (b4) to "1".
2: f(XIN)/16 is supplied as the count source to the prescaler 12 at reset, the count source before executing the STP instruction is supplied as the count source at executing STP instruction.
3: When bit 0 of MISRG is "0", timer 1 is set "01_{16}" and prescaler 12 is set "FF_{16}" automatically. When bit 0 of MISRG is "1" , set the appropriate value to them in accordance with oscillation stabilizing time required by the using oscillator because nothing is automatically set into timer 1 and prescaler 12.
4: Although a feed-back resistor exists on-chip, an external feed-back resistor may be needed depending on conditions.

Fig. 79 System clock generating circuit block diagram (Single-chip mode)



Notes 1: Switch the mode by the allows shown between the mode blocks. (Do not switch between the modes directly without an allow.)
2: The all modes can be switched to the stop mode or the wait mode and return to the source mode when the stop mode or the wait mode is ended.
3: Timer operates in the wait mode.
4: When the stop mode is ended, a delay of approximately 1 ms occurs by connecting prescaler 12 and Timer 1 in middle/high-speed mode.
5: When the stop mode is ended, a delay of approximately 0.25 s occurs by Timer 1 and Timer 2 in low-speed mode.
6: Wait until oscillation stabilizes after oscillating the main clock XIN before the switching from the low-speed mode to middle/high-speed mode.
7: The example assumes that 8 MHz is being applied to the XIN pin and 32 kHz to the XCIN pin. ϕ indicates the internal clock.

Fig. 80 State transitions of system clock

FLASH MEMORY MODE

The 3804 group (Spec.L)'s flash memory version has the flash memory that can be rewritten with a single power source.
For this flash memory, three flash memory modes are available in which to read, program, and erase: the parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and the CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU).

This flash memory version has some blocks on the flash memory as shown in Figure 81 and each block can be erased.
In addition to the ordinary User ROM area to store the MCU operation control program, the flash memory has a Boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This Boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This Boot ROM area can be rewritten in only parallel I/O mode.

Summary

Table 15 lists the summary of the 3804 group (Spec.L) flash memory version.

Table 15 Summary of 3804 group (Spec.L)'s flash memory version

Item		Specifications
Power source voltage (V_{CC})		V_{CC} = 2.7 to 5.5 V
Program/Erase V_{PP} voltage (V_{PP})		V_{CC} = 2.7 to 5.5 V
Flash memory mode		3 modes; Parallel I/O mode, Standard serial I/O mode, CPU rewrite mode
Erase block division	User ROM area/Data ROM area	Refer to Figure 81.
	Boot ROM area (1)	Not divided (4 Kbytes)
Program method		In units of bytes
Erase method		Block erase
Program/Erase control method		Program/Erase control by software command
Number of commands		5 commands
Number of program/Erase times		100(Max.)
ROM code protection		Available in parallel I/O mode and standard serial I/O mode

NOTE:
1. The Boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. This Boot ROM area can be erased and written in only parallel I/O mode.

Table 16 Electrical characteristics of flash memory (program ROM)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
–	Byte programming time		V_{CC} = 5.0 V, Topr = 25 °C	–	60	400	μs
–	Block erase time	(Block 1)	V_{CC} = 5.0 V, Topr = 25 °C	–	0.5	9	s
		(Block 2)		–	0.9	9	s
		(Block 3)		–	1.3	9	s
		(Block A, B)		–	0.3	9	s

NOTES:
1. V_{CC} = AV_{CC} = 2.7 V to 5.5 V, Topr = 0 °C to 60 °C, unless otherwise noted.
2. Definition of programming/erase count
   The programming/erase count refers to the number of erase operations per block. For example, if block A is a 2 K-byte block and 2,048 1-byte writes are performed, all to different addresses, after which block A is erased, the programming/erase count is 1. Note that for each erase operation it is not possible to perform more than one programming (write) operation to the same address (overwrites prohibited).
3. This is the number of times for which all electrical characteristics are guaranteed after a programming or erase operation. (The guarantee covers the range from 1 to maximum value.)
4. On systems where reprogramming is performed a large number of times, it is possible to reduce the effective number of overwrites by sequentially shifting the write address, so that as much of the available area of the block is used up through successive programming (write) operations before an erase operation is performed. For example, if each programming operation uses 16 bytes of space, a maximum of 128 programming operations may be performed before it becomes necessary to erase the block in order to continue. In this way the effective number of overwrites can be kept low. The effective overwrite count can be further reduced by evenly dividing operations between block A and block B. It is recommended that data be retained on the number of times each block has been erased and a limit count set.
5. If a block erase error occurs, execute the clear status register command followed by the block erase command a minimum of three times and until the erase error is no longer generated.

Boot Mode

The control program for CPU rewrite mode must be written into the User ROM or Boot ROM area in parallel I/O mode beforehand. (If the control program is written into the Boot ROM area, the standard serial I/O mode becomes unusable.)
See Figure 81 for details about the Boot ROM area.
Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVSS pin low. In this case, the CPU starts operating using the control program in the User ROM area.
When the microcomputer is reset and the CNVSS pin high after pulling the P45/TxD1 pin and CNVSS pin high, the CPU starts operating (start address of program is stored into addresses FFFC16 and FFFD16) using the control program in the Boot ROM area. This mode is called the “Boot mode”. Also, User ROM area can be rewritten using the control program in the Boot ROM area.

Block Address

Block addresses refer to the maximum address of each block. These addresses are used in the block erase command.

CPU Rewrite Mode

In CPU rewrite mode, the internal flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU).
In CPU rewrite mode, only the User ROM area shown in Figure 81 can be rewritten; the Boot ROM area cannot be rewritten. Make sure the program and block erase commands are issued for only the User ROM area and each block area.
The control program for CPU rewrite mode can be stored in either User ROM or Boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to internal RAM area before it can be executed.

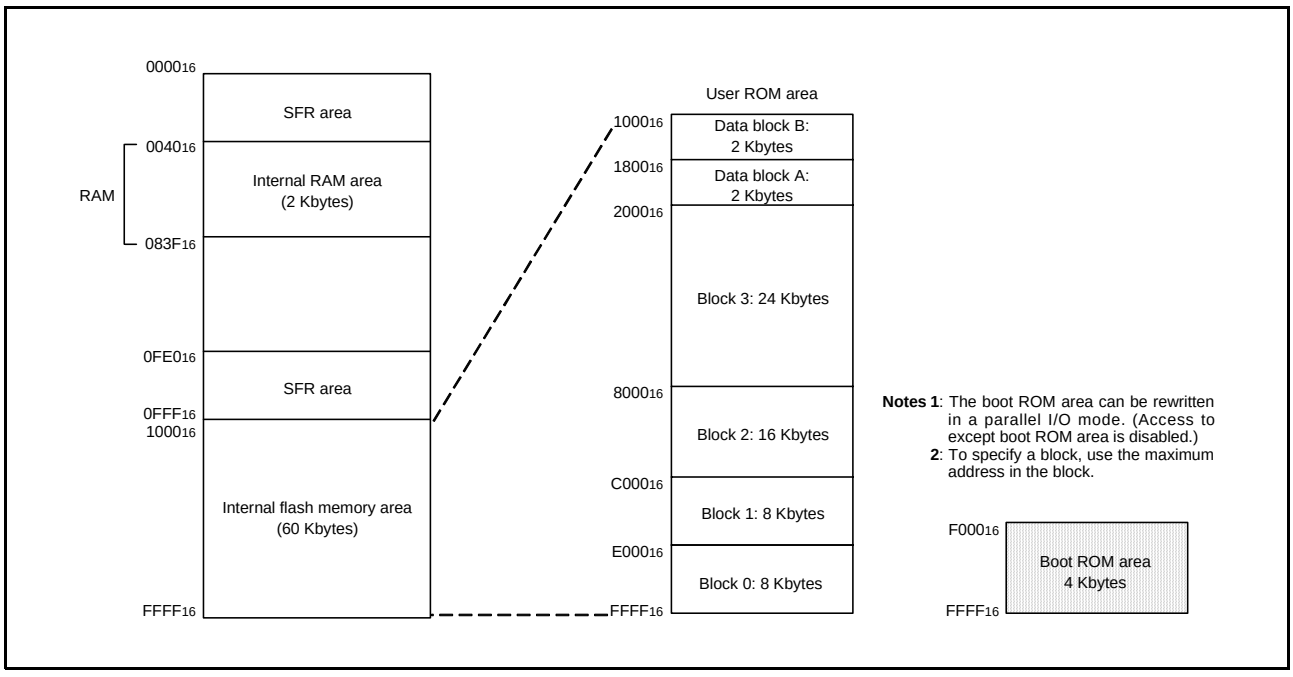


Fig. 81 Block diagram of built-in flash memory

Outline Performance

CPU rewrite mode is usable in the single-chip or Boot mode. The only User ROM area can be rewritten.

In CPU rewrite mode, the CPU erases, programs and reads the internal flash memory as instructed by software commands. This rewrite control program must be transferred to internal RAM area before it can be executed.

The MCU enters CPU rewrite mode by setting “1” to the CPU rewrite mode select bit (bit 1 of address 0FE0$_{16}$). Then, software commands can be accepted.

Use software commands to control program and erase operations. Whether a program or erase operation has terminated normally or in error can be verified by reading the status register.

Figure 82 shows the flash memory control register 0.

Bit 0 of the flash memory control register 0 is the RY/$\overline{\text{BY}}$ 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 8 KB user block E/W enable bit. By setting combination of bit 4 of the flash memory control register 2 and this bit as shown in Table 17, E/W is disabled to user block in the CPU rewriting mode.

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 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”. 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.

Figure 83 shows the flash memory control register 1.

Bit 0 of the flash memory control register 1 is the Erase suspend enable bit. By setting this bit to “1”, the erase suspend mode to suspend erase processing temporary when block erase command is executed can be used. In order to set this bit to “1”, writing “0” and “1” in succession to bit 0. In order to set this bit to “0”, write “0” only to bit 0.

Bit 1 of the flash memory control register 1 is the erase suspend request bit. By setting this bit to “1” when erase suspend enable bit is “1”, the erase processing is suspended.

Bit 6 of the flash memory control register 1 is the erase suspend flag. This bit is cleared to “0” at the flash erasing.

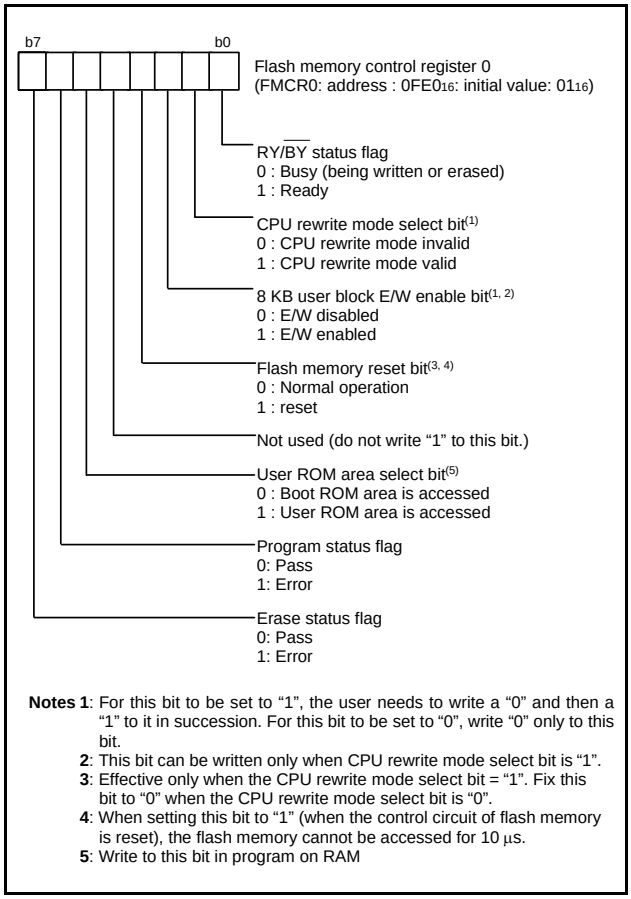


Fig. 82 Structure of flash memory control register 0

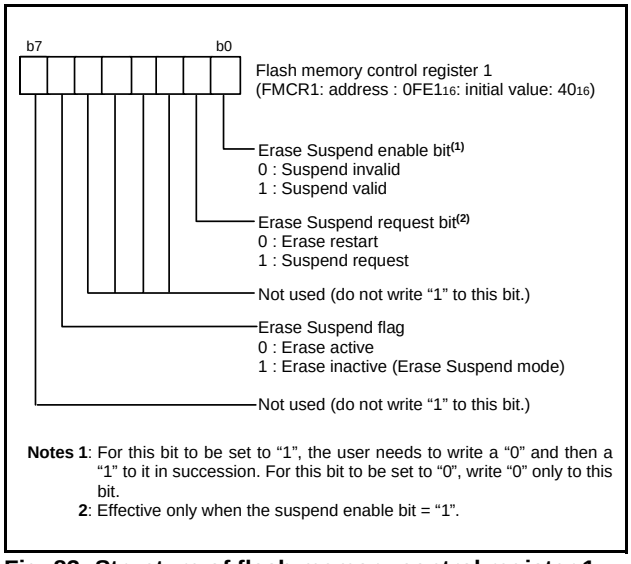
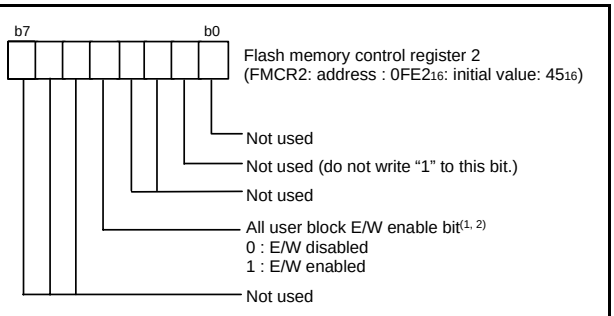


Fig. 83 Structure of flash memory control register 1



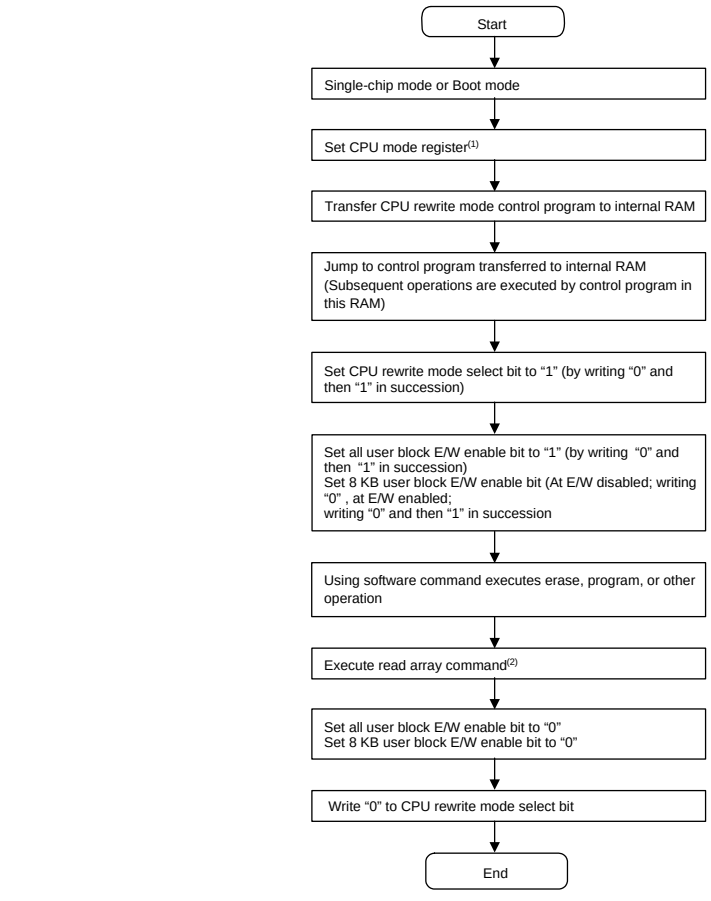
Notes 1: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. For this bit to be set to “0”, write “0” only to this bit.
2: Effective only when the CPU rewrite mode select bit = “1”.

Fig. 84 Structure of flash memory control register 2

Table 17 State of E/W inhibition function

All user block E/W enable bit	8 KB user block E/W enable bit	8 KB × 2 block Addresses C000₁₆ to FFFF₁₆	16 KB + 24 KB block Addresses 2000₁₆ to BFFF₁₆	Data block Addresses 1000₁₆ to 1FFF₁₆
0	0	E/W disabled	E/W disabled	E/W enabled
0	1	E/W disabled	E/W disabled	E/W enabled
1	0	E/W disabled	E/W enabled	E/W enabled
1	1	E/W enabled	E/W enabled	E/W enabled

Figure 85 shows a flowchart for setting/releasing CPU rewrite mode.



Notes 1: Set the main clock as follows depending on the clock division ratio selection bits of CPU mode register (bits 6, 7 of address 003B₁₆).
2: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute the read array command.

Fig. 85 CPU rewrite mode set/release flowchart be sure to execute

<Notes on CPU Rewrite Mode>

Take the notes described below when rewriting the flash memory in CPU rewrite mode.

(1) Operation speed

During CPU rewrite mode, set the system clock ϕ to 4.0 MHz or less using the clock division ratio selection bits (bits 6 and 7 of address $003B_{16}$).

(2) Instructions inhibited against use

The instructions which refer to the internal data of the flash memory cannot be used during CPU rewrite mode.

(3) Interrupts

The interrupts cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory.

(4) Watchdog timer

If the watchdog timer has been already activated, internal reset due to an underflow will not occur because the watchdog timer is surely cleared during program or erase.

(5) Reset

Reset is always valid. The MCU is activated using the boot mode at release of reset in the condition of CNV_{SS} = "H", so that the program will begin at the address which is stored in addresses $FFFC_{16}$ and $FFFD_{16}$ of the boot ROM area.

Software Commands

Table 18 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.

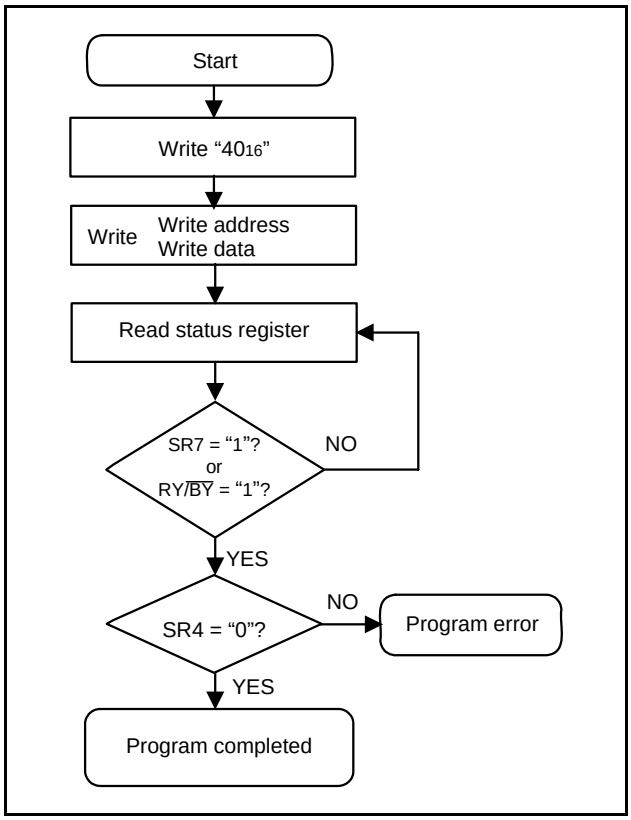


Fig. 86 Program flowchart

Table 18 List of software commands (CPU rewrite mode)

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

NOTES:
1. SRD = Status Register Data
2. WA = Write Address, WD = Write Data
3. BA = Block Address to be erased (Input the maximum address of each block.)
4. X denotes a given address in the User ROM area.

• Block Erase Command (20_{16}/$D0_{16}$)

By writing the command code "20_{16}" in the first bus cycle and the confirmation command code "$D0_{16}$" and the block address in the second bus cycle that follows, the block erase (erase and erase verify) operation starts for the block address of the flash memory to be specified.

Whether the block erase operation is completed can be confirmed by read status register or the RY/$\overline{BY}$ status flag of flash memory control register. At the same time the block erase operation starts, the read status register mode is automatically entered, so that the contents of the status register can be read out. The status register

bit 7 (SR7) is set to "0" at the same time the block erase operation starts and is returned to "1" upon completion of the block erase 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 is "0" during block erase operation and "1" when the block erase operation is completed as is the status register bit 7.

After the block erase ends, erase results can be checked by reading the status register. For details, refer to the section where the status register is detailed.

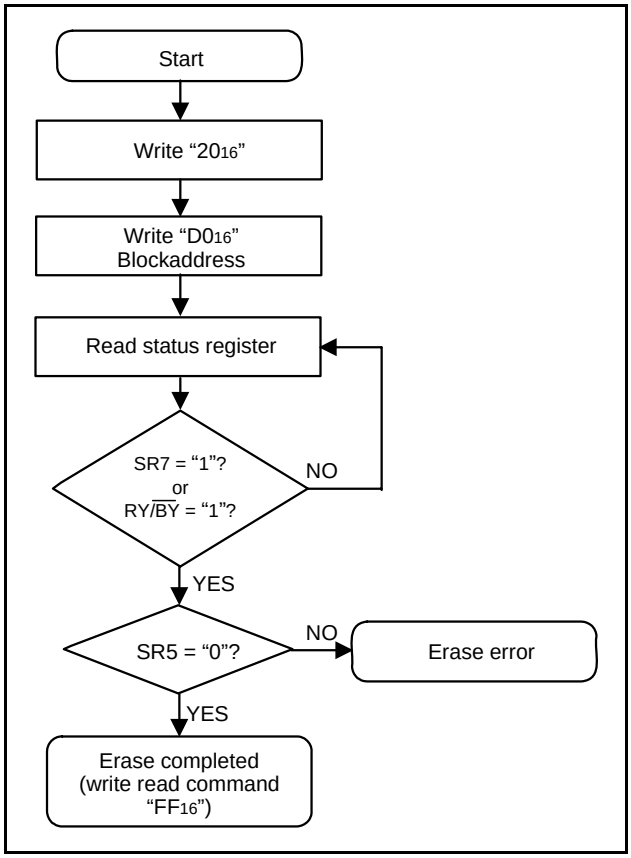


Fig. 87 Erase flowchart

• Status Register

The status register shows the operating status of the flash memory and whether erase operations and programs ended successfully or in error. It can be read in the following ways:

(1) By reading an arbitrary address from the User ROM area after writing the read status register command (70_{16})
(2) By reading an arbitrary address from the User ROM area in the period from when the program starts or erase operation starts to when the read array command (FF_{16}) is input.

Also, the status register can be cleared by writing the clear status register command (50_{16}).
After reset, the status register is set to "80_{16}".
Table 19 shows the status register. Each bit in this register is explained below.

• Sequencer status (SR7)

The sequencer status indicates the operating status of the flash memory. This bit is set to "0" (busy) during write or erase operation and is set to "1" when these operations ends.
After power-on, the sequencer status is set to "1" (ready).

• Erase status (SR5)

The erase status indicates the operating status of erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is reset to "0".

• Program status (SR4)

The program status indicates the operating status of write operation.
When a write error occurs, it is set to "1".
The program status is reset to "0" when it is cleared.

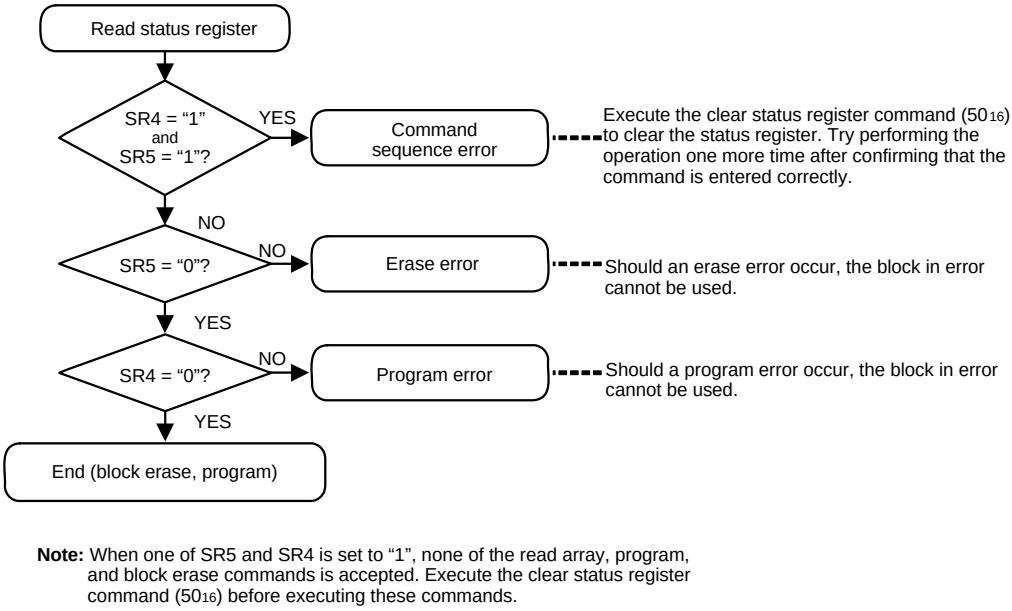
If "1" is written for any of the SR5 and SR4 bits, the read array, program, and block erase commands are not accepted. Before executing these commands, execute the clear status register command (50_{16}) and clear the status register.
Also, if any commands are not correct, both SR5 and SR4 are set to "1".

Table 19 Definition of each bit in status register

Each bit of SRD bits	Status name	Definition	
		"1"	"0"
SR7 (bit 7)	Sequencer status	Ready	Busy
SR6 (bit 6)	Reserved	–	–
SR5 (bit 5)	Erase status	Terminated in error	Terminated normally
SR4 (bit 4)	Program status	Terminated in error	Terminated normally
SR3 (bit 3)	Reserved	–	–
SR2 (bit 2)	Reserved	–	–
SR1 (bit 1)	Reserved	–	–
SR0 (bit 0)	Reserved	–	–

Full Status Check

By performing full status check, it is possible to know the execution results of erase and program operations. Figure 88 shows a full status check flowchart and the action to be taken when each error occurs.



Note: When one of SR5 and SR4 is set to "1", none of the read array, program, and block erase commands is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Fig. 88 Full status check flowchart and remedial procedure for errors

Functions To Inhibit Rewriting Flash Memory Version

To prevent the contents of internal flash memory from being read out or rewritten easily, this MCU incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

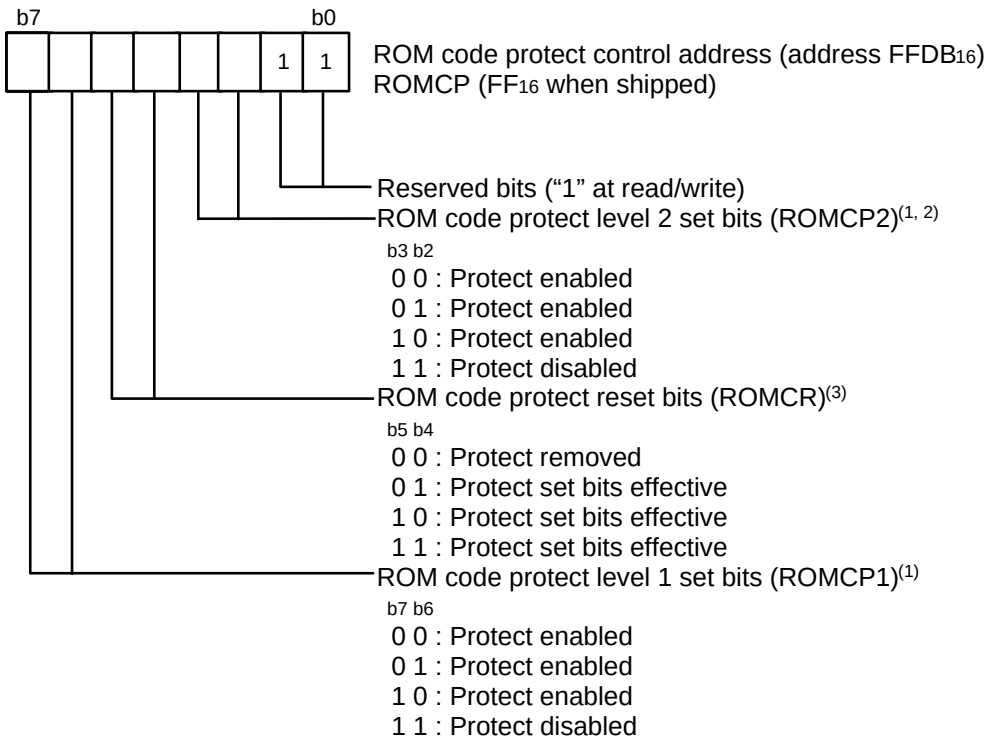
• ROM Code Protect Function

The ROM code protect function is the function to inhibit reading out or modifying the contents of internal flash memory by using the ROM code protect control address (address FFDB16) in parallel I/O mode. Figure 89 shows the ROM code protect control address (address FFDB16). (This address exists in the User ROM area.)

If one or both of the pair of ROM code protect bits is set to "0", the ROM code protect is turned on, so that the contents of internal flash memory are protected against readout and modification. The ROM code protect is implemented in two levels. If level 2 is selected, the flash memory is protected even against readout by a shipment inspection LSI tester, etc. When an attempt is made to select both level 1 and level 2, level 2 is selected by default.

If both of the two ROM code protect reset bits are set to "00", the ROM code protect is turned off, so that the contents of internal flash memory can be readout or modified. Once the ROM code protect is turned on, the contents of the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial I/O or CPU rewrite mode to rewrite the contents of the ROM code protect reset bits.

Rewriting of only the ROM code protect control address (address FFDB16) cannot be performed. When rewriting the ROM code protect reset bit, rewrite the whole user ROM area (block 0) containing the ROM code protect control address.



Notes 1: When ROM code protect is turned on, the internal flash memory is protected against readout or modification in parallel I/O mode.
2: When ROM code protect level 2 is turned on, ROM code readout by a shipment inspection LSI tester, etc. also is inhibited.
3: The ROM code protect reset bits can be used to turn off ROM code protect level 1 and ROM code protect level 2. However, since these bits cannot be modified in parallel I/O mode, they need to be rewritten in serial I/O mode or CPU rewrite mode.

Fig. 89 Structure of ROM code protect control address

• ID Code Check Function

Use this function in standard serial I/O mode. When the contents of the flash memory are not blank, the ID code sent from the programmer is compared with the ID code written in the flash memory to see if they match. If the ID codes do not match, the commands sent from the programmer are not accepted. The ID code consists of 8-bit data, and its areas are FFD4$_{16}$ to FFDA$_{16}$. Write a program which has had the ID code preset at these addresses to the flash memory.

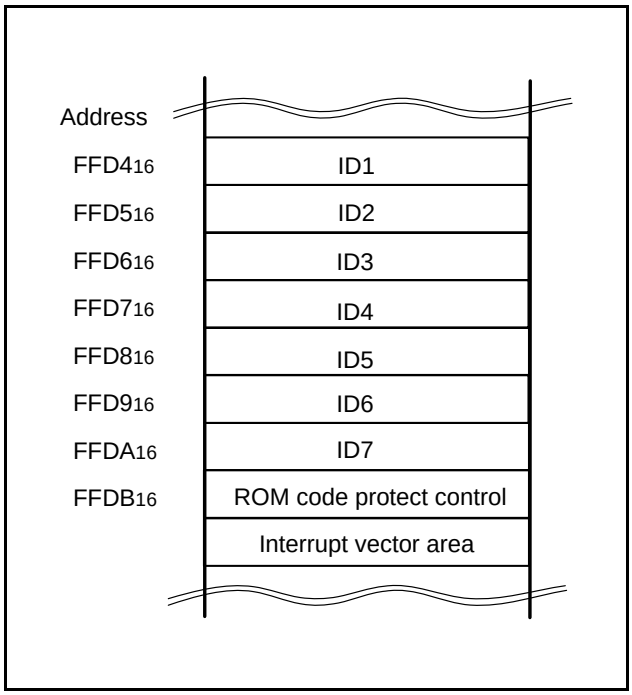


Fig. 90 ID code store addresses

Parallel I/O Mode

The parallel I/O mode is used to input/output software commands, address and data in parallel for operation (read, program and erase) to internal flash memory.
Use the external device (writer) only for 3804 group (Spec.L) flash memory version. For details, refer to the user's manual of each writer manufacturer.

• User ROM and Boot ROM Areas

In parallel I/O mode, the User ROM and Boot ROM areas shown in Figure 81 can be rewritten. Both areas of flash memory can be operated on in the same way.
The Boot ROM area is 4 Kbytes in size and located at addresses $F000_{16}$ through $FFFF_{16}$. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)
In the Boot ROM area, an erase block operation is applied to only one 4 Kbyte block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. Therefore, using the MCU in standard serial I/O mode, do not rewrite to the Boot ROM area.

Standard serial I/O Mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is clock synchronized serial. This mode requires a purpose-specific peripheral unit.

The standard serial I/O mode is different from the parallel I/O mode in that the CPU controls flash memory rewrite (uses the CPU rewrite mode), rewrite data input and so forth. The standard serial I/O mode is started by connecting "H" to the CNVSS pin and "H" to the P45 (BOOTENT) pin, and releasing the reset operation. (In the ordinary microcomputer mode, set CNVSS pin to "L" level.) This control program is written in the Boot ROM area when the product is shipped from Renesas. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the Boot ROM area is rewritten in parallel I/O mode. The standard serial I/ O mode has standard serial I/O mode 1 of the clock synchronous serial and standard serial I/O mode 2 of the clock asynchronous serial. Table 20 and 21 show description of pin function (standard serial I/O mode). Figures 91 to 96 show the pin connections for the standard serial I/O mode. Figures 97 and 98 show the operating waveform for standard serial I/O mode 1 and the operating waveform for standard serial I/O mode 1, respectively. Figures 99 and 100 show the connection examples in standard serial I/O mode.

In standard serial I/O mode, only the User ROM area shown in Figure 81 can be rewritten. The Boot ROM area cannot be written.

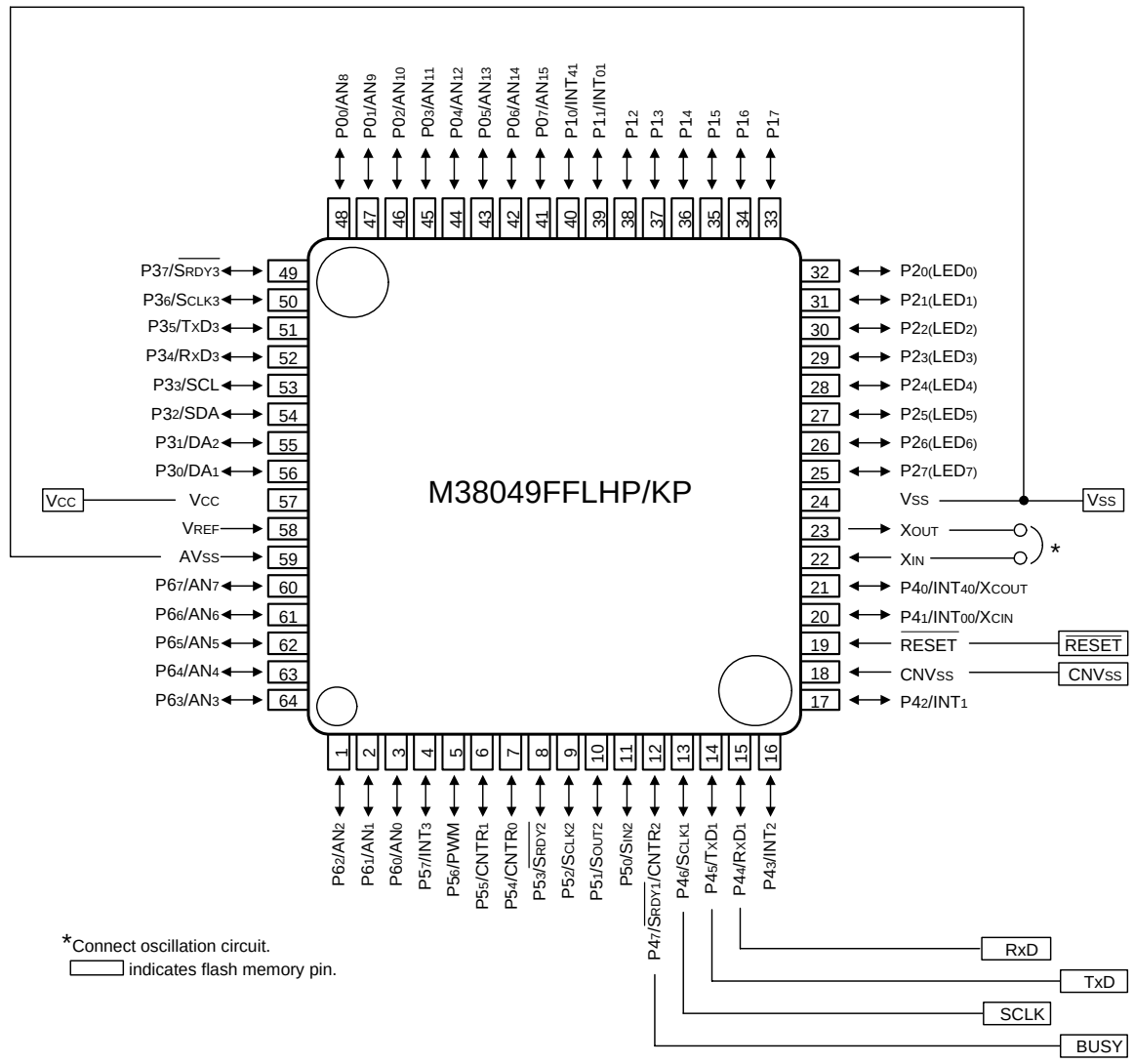
In standard serial I/O mode, a 7-byte ID code is used. When there is data in the flash memory, this function determines whether the ID code sent from the peripheral unit (programmer) and those written in the flash memory match. The commands sent from the peripheral unit (programmer) are not accepted unless the ID code matches.

Table 20 Description of pin function (Flash Memory Serial I/O Mode 1)

Pin name	Signal name	I/O	Function
V_{CC},V_{SS}	Power supply	I	Apply 2.7 to 5.5 V to the V_{CC} pin and 0 V to the V_{SS} pin.
CNV_{SS}	CNV_{SS}	I	After input of port is set, input “H” level.
$\overline{RESET}$	Reset input	I	Reset input pin. To reset the microcomputer, $\overline{RESET}$ pin should be held at an “L” level for 16 cycles or more of X_{IN}.
X_{IN}	Clock input	I	Connect an oscillation circuit between the X_{IN} and X_{OUT} pins. As for the connection method, refer to the “clock generating circuit”.
X_{OUT}	Clock output	O	
AV_{SS}	Analog power supply input		Connect AV_{SS} to V_{SS}.
V_{REF}	Reference voltage input	I	Apply reference voltage of A/D to this pin.
$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$, $P5_0$–$P5_7$, $P6_0$–$P6_7$	I/O port	I/O	Input “L” or “H” level, or keep open.
$P4_4$	RxD input	I	Serial data input pin.
$P4_5$	TxD output	O	Serial data output pin.
$P4_6$	S_{CLK} input	I	Serial clock input pin.
$P4_7$	BUSY output	O	BUSY signal output pin.

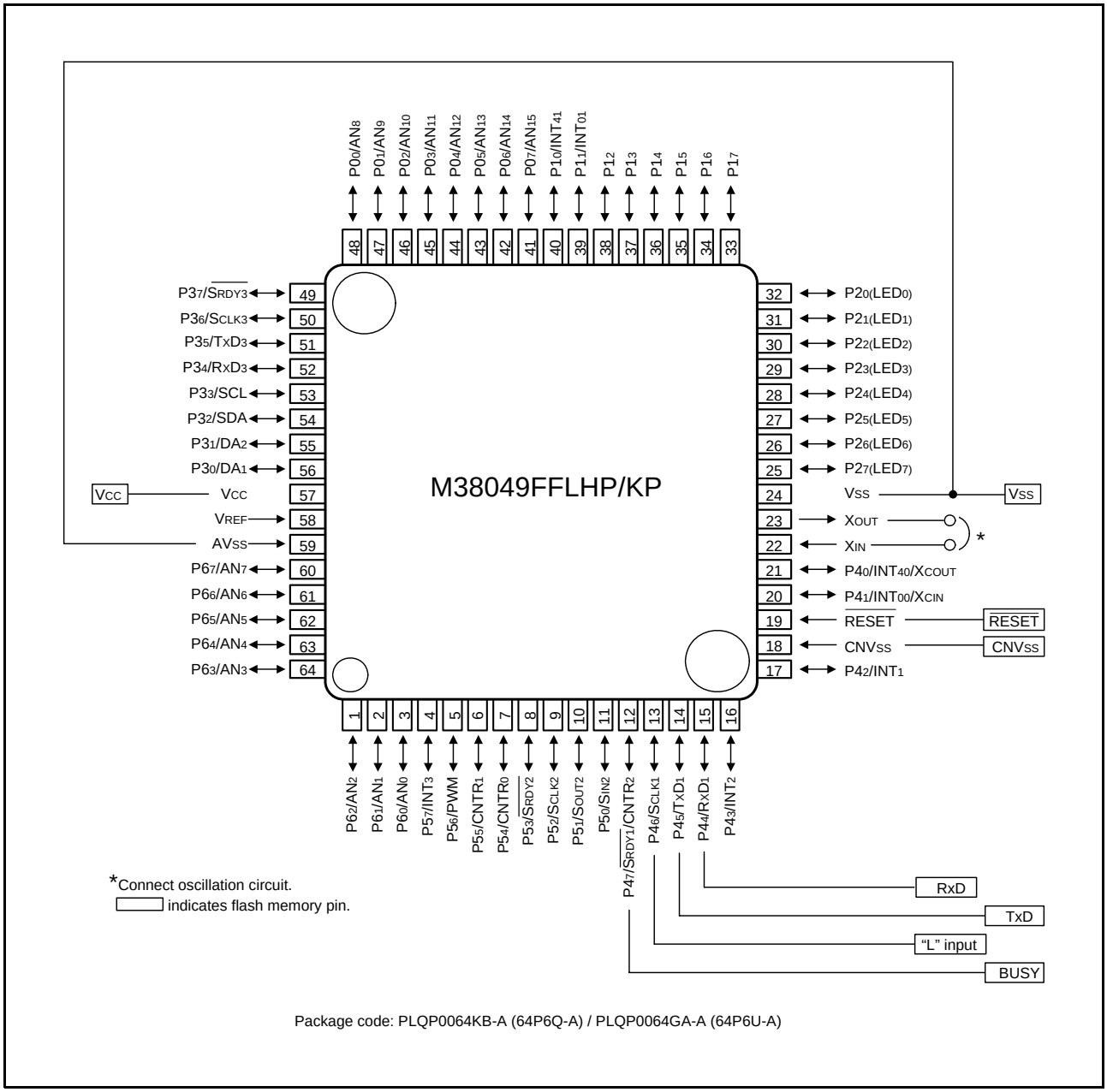
Table 21 Description of pin function (Flash Memory Serial I/O Mode 2)

Pin name	Signal name	I/O	Function
V_{CC},V_{SS}	Power supply	I	Apply 2.7 to 5.5 V to the V_{CC} pin and 0 V to the V_{SS} pin.
CNV_{SS}	CNV_{SS}	I	After input of port is set, input “H” level.
$\overline{RESET}$	Reset input	I	Reset input pin. To reset the microcomputer, $\overline{RESET}$ pin should be held at an “L” level for 16 cycles or more of X_{IN}.
X_{IN}	Clock input	I	Connect an oscillation circuit between the X_{IN} and X_{OUT} pins. As for the connection method, refer to the “clock generating circuit”.
X_{OUT}	Clock output	O	
AV_{SS}	Analog power supply input		Connect AV_{SS} to V_{SS}.
V_{REF}	Reference voltage input	I	Apply reference voltage of A/D to this pin.
$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$, $P5_0$–$P5_7$, $P6_0$–$P6_7$	I/O port	I/O	Input “L” or “H” level, or keep open.
$P4_4$	RxD input	I	Serial data input pin.
$P4_5$	TxD output	O	Serial data output pin.
$P4_6$	S_{CLK} input	I	Input “L” level.
$P4_7$	BUSY output	O	BUSY signal output pin.



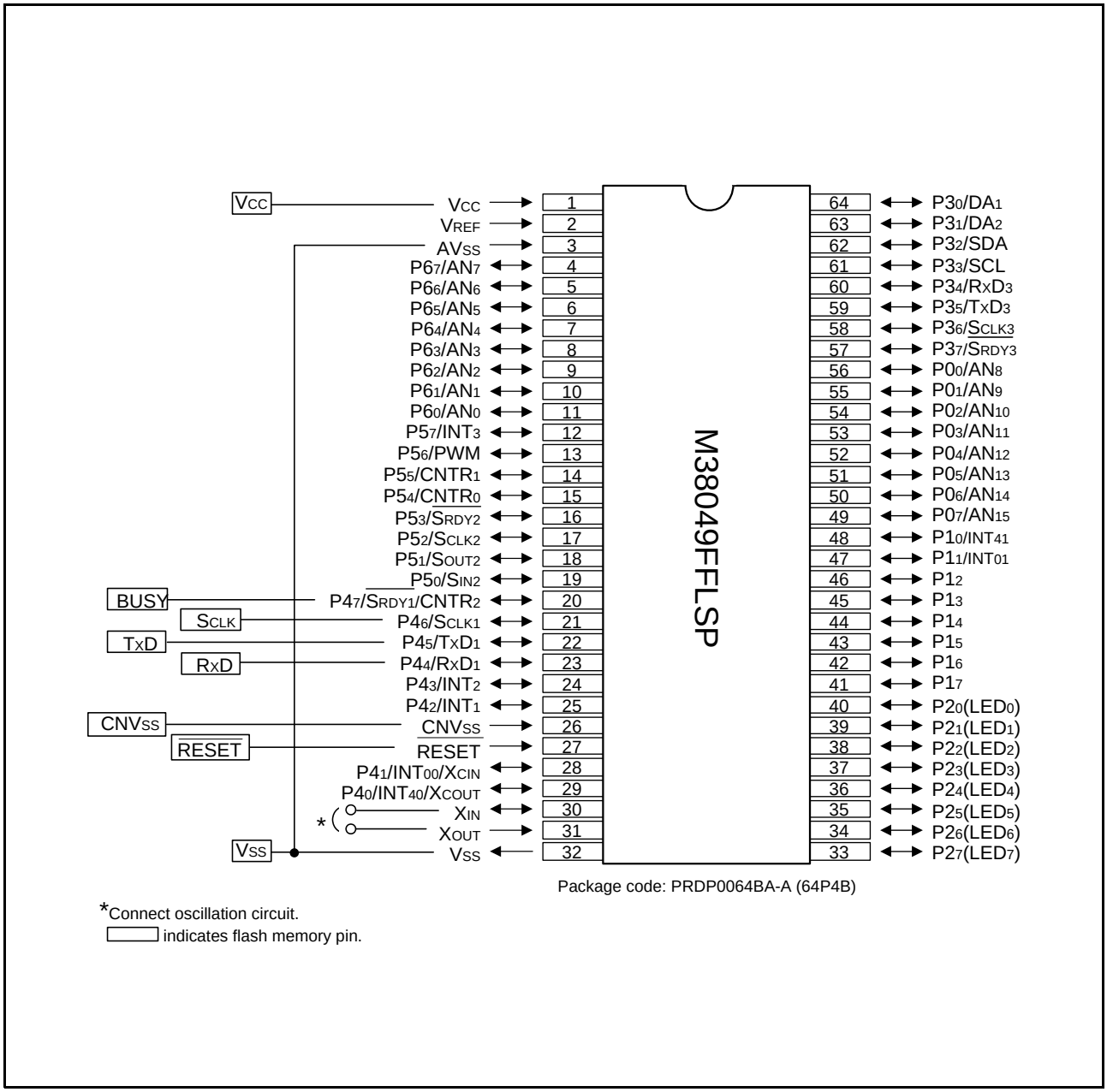
Package code: PLQP0064KB-A (64P6Q-A) / PLQP0064GA-A (64P6U-A)

Fig. 91 Connection for standard serial I/O mode 1 (M38049FFLHP/KP)



Package code: PLQP0064KB-A (64P6Q-A) / PLQP0064GA-A (64P6U-A)

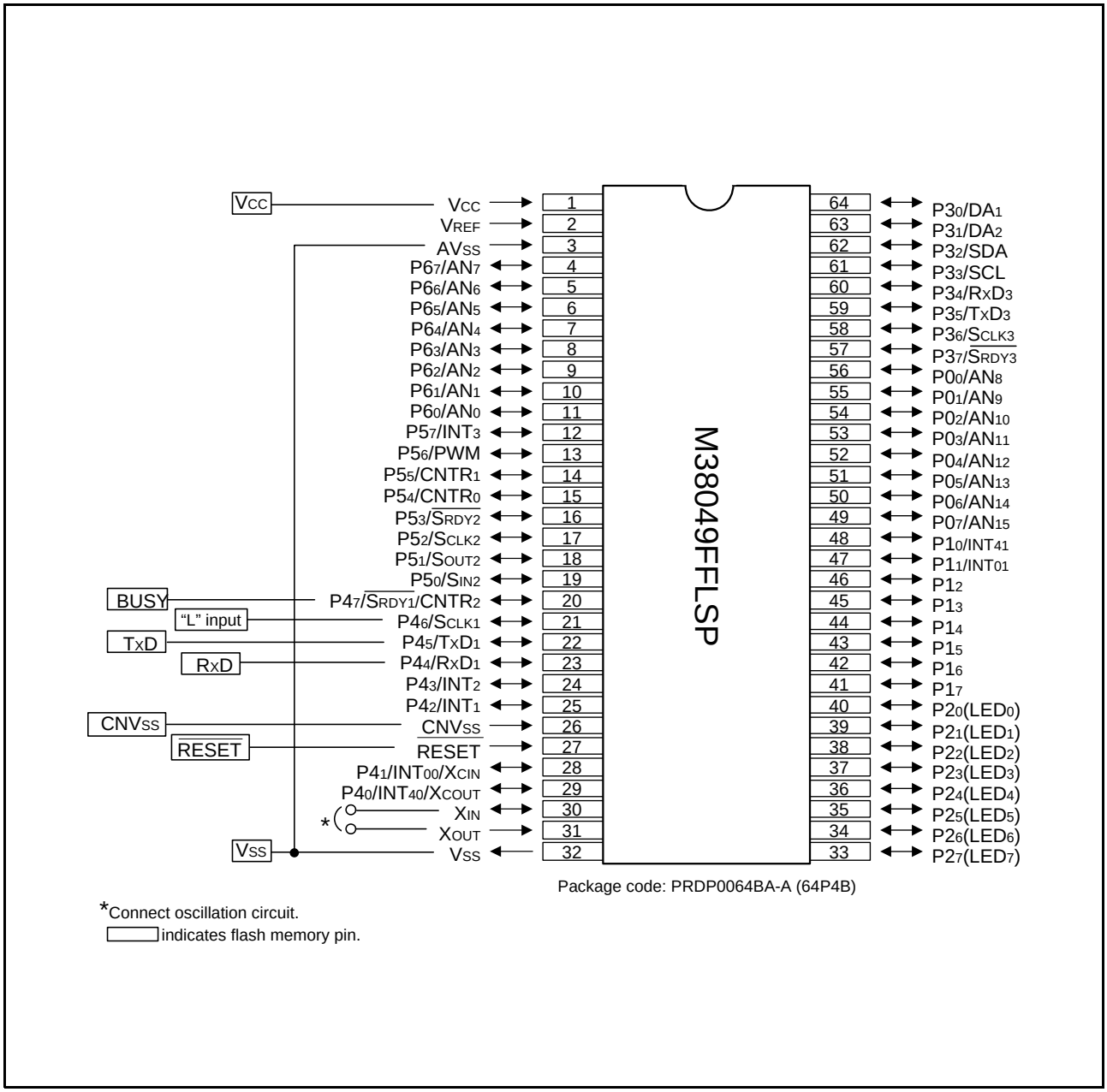
Fig. 92 Connection for standard serial I/O mode 2 (M38049FFLHP/KP)



Package code: PRDP0064BA-A (64P4B)

*Connect oscillation circuit.
indicates flash memory pin.

Fig. 93 Connection for standard serial I/O mode 1 (M38049FFLSP)



Package code: PRDP0064BA-A (64P4B)

*Connect oscillation circuit.

indicates flash memory pin.

Fig. 94 Connection for standard serial I/O mode 2 (M38049FFLSP)

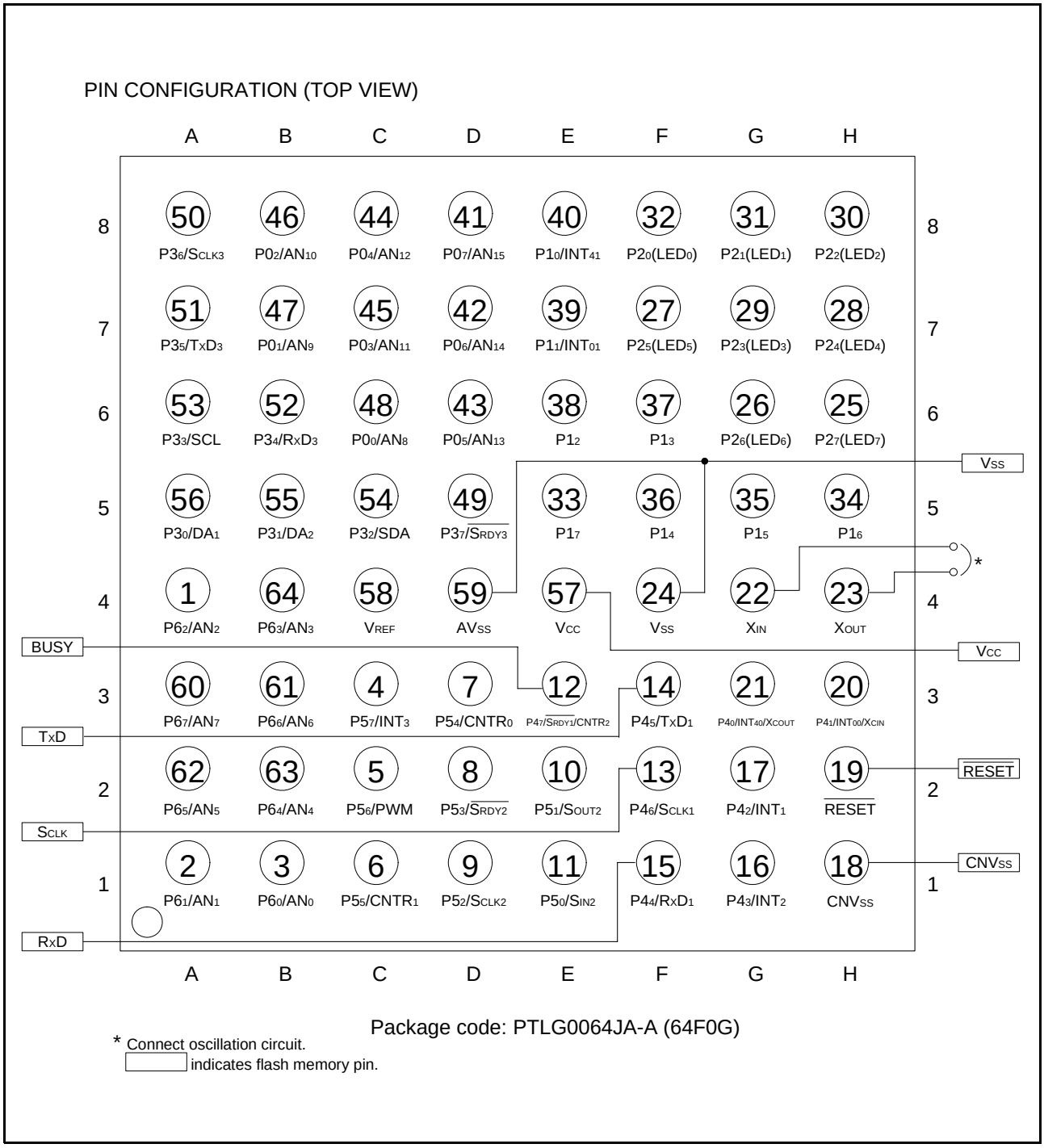
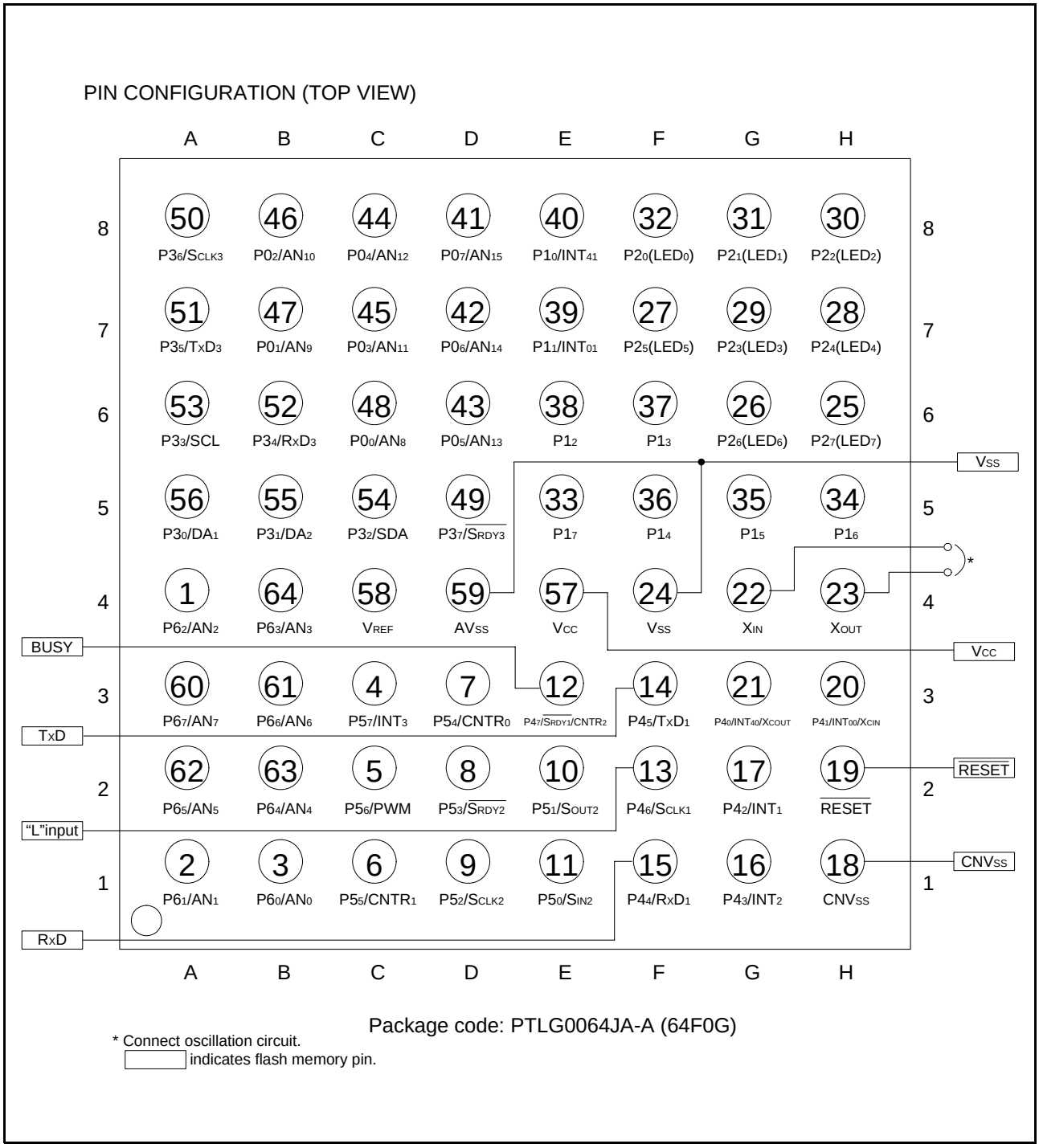


Fig. 95 Connection for standard serial I/O mode 1 (M38049FFLWG)

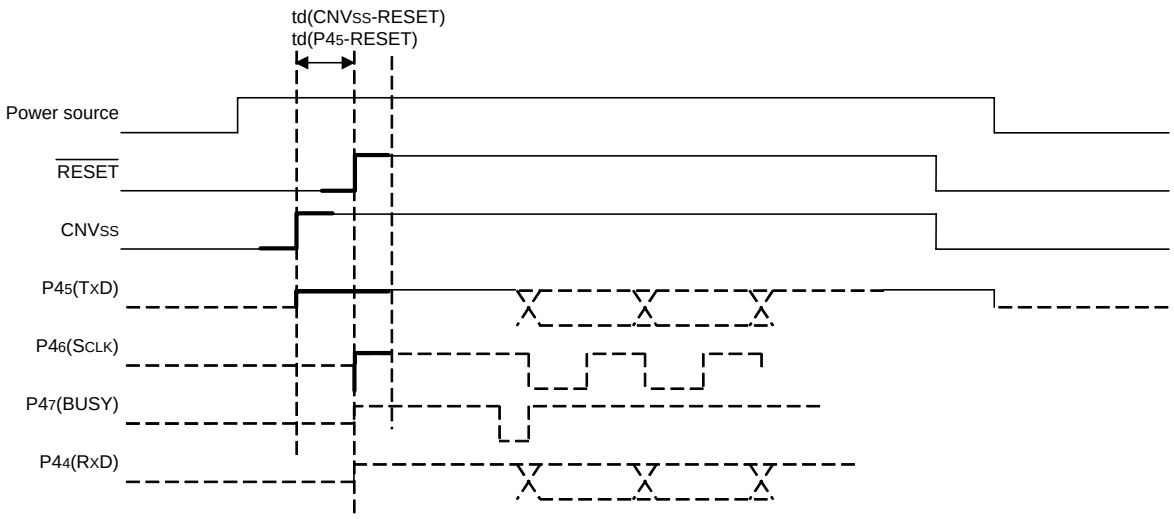


Package code: PTLG0064JA-A (64F0G)

* Connect oscillation circuit.

□ indicates flash memory pin.

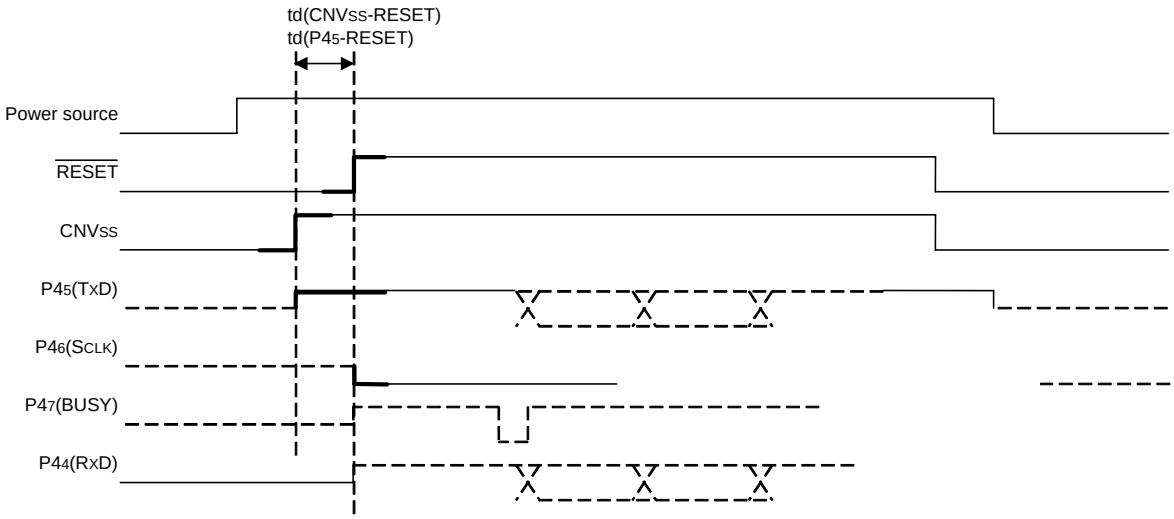
Fig. 96 Connection for standard serial I/O mode 2 (M38049FFLWG)



Symbol	Limits			Unit
	Min.	Typ.	Max.	
td(CNVSS-RESET)	0	–	–	ms
td(P45-RESET)	0			ms

Notes: In the standard serial I/O mode 1, input "H" to the P46 pin.
Be sure to set the CNVSS pin to "H" before rising RESET.
Be sure to set the P45 pin to "H" before rising RESET.

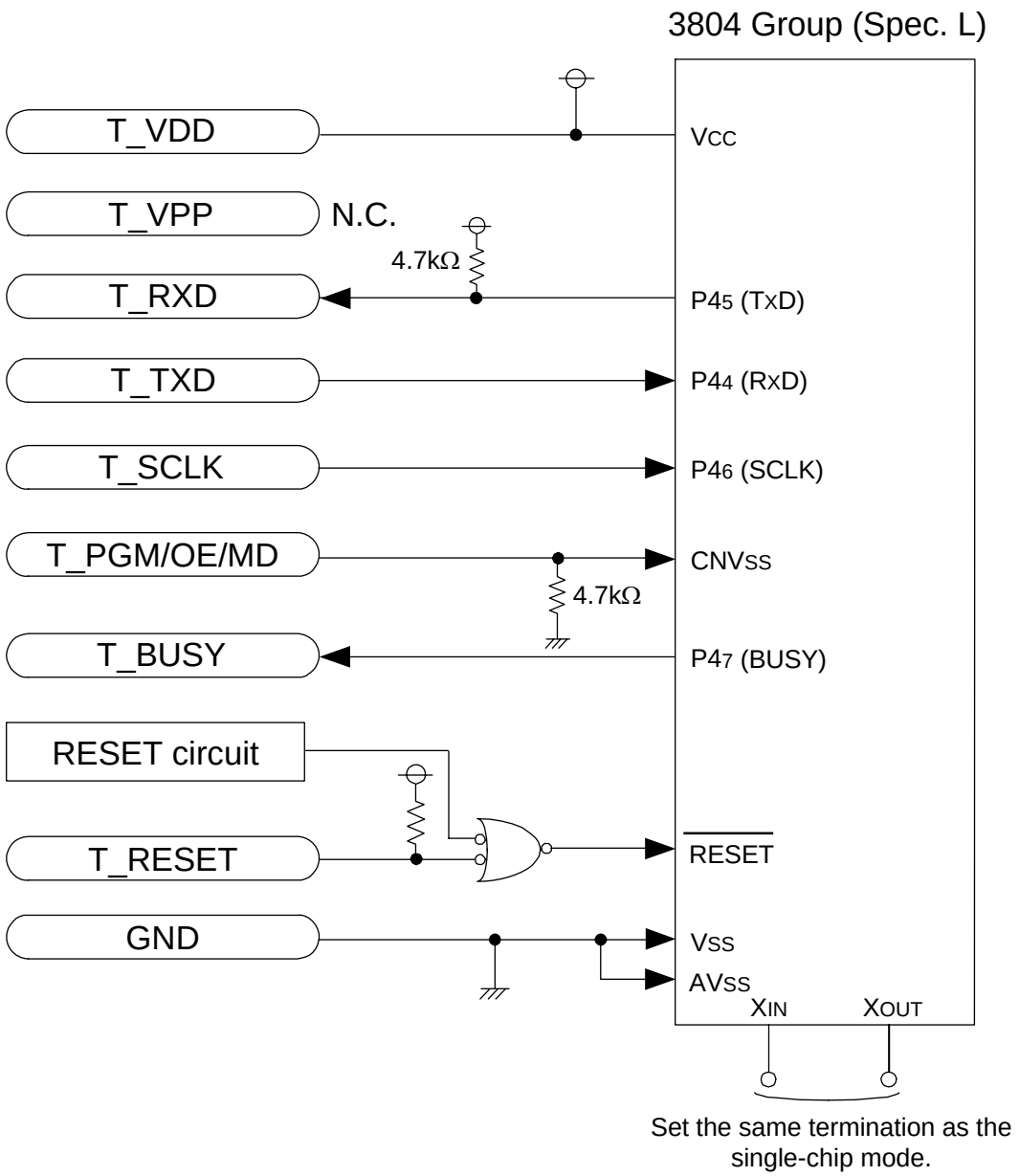
Fig. 97 Operating waveform for standard serial I/O mode 1



Symbol	Limits			Unit
	Min.	Typ.	Max.	
td(CNVSS-RESET)	0	–	–	ms
td(P45-RESET)	0			ms

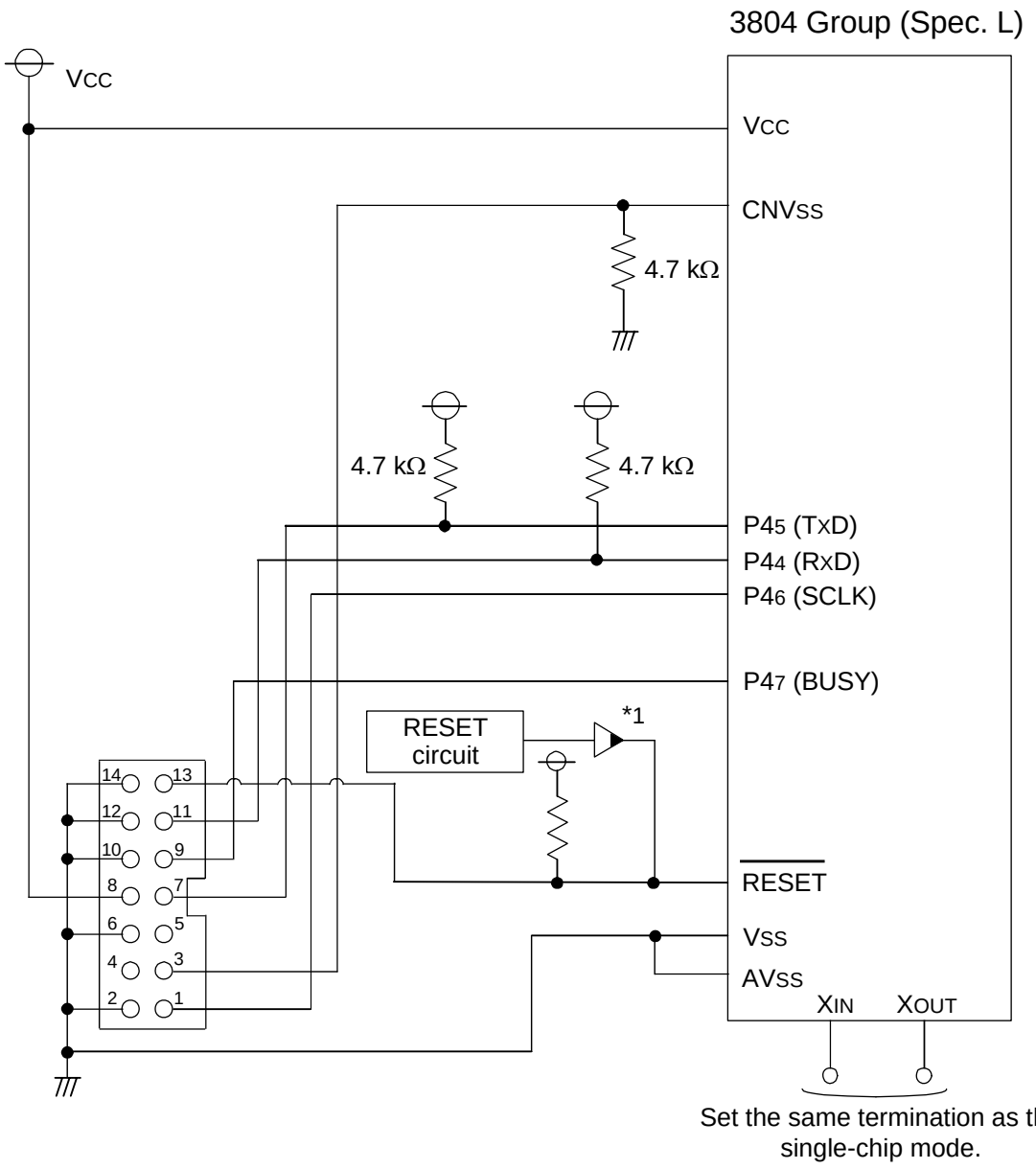
Notes: In the standard serial I/O mode 2, input "H" to the P46 pin.
Be sure to set the CNVSS pin to "H" before rising RESET.
Be sure to set the P45 pin to "H" before rising RESET.

Fig. 98 Operating waveform for standard serial I/O mode 2



Note: For the programming circuit, the wiring capacity of each signal pin must not exceed 47 pF.

Fig. 99 When using programmer (in standard serial I/O mode 1) of Suisei Electronics System Co., LTD, connection example



*1 : Open-collector buffer

Note : For the programming circuit, the wiring capacity of each signal pin must not exceed 47 pF.

Fig. 100 When using E8 programmer (in standard serial I/O mode 1), connection example

NOTES

NOTES ON PROGRAMMING

1. Processor Status Register

(1) Initializing of processor status register

Flags which affect program execution must be initialized after a reset. In particular, it is essential to initialize the T and D flags because they have an important effect on calculations.

<Reason>

After a reset, the contents of the processor status register (PS) are undefined except for the I flag which is "1".

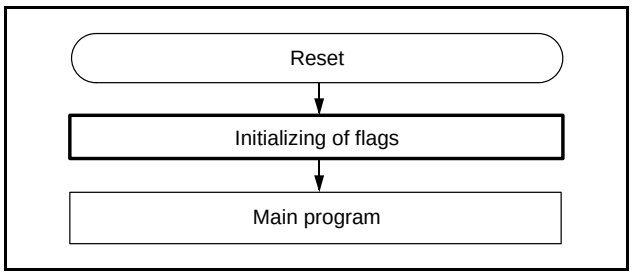


Fig. 101 Initialization of processor status register

(2) How to reference the processor status register

To reference the contents of the processor status register (PS), execute the PHP instruction once then read the contents of (S+1). If necessary, execute the PLP instruction to return the PS to its original status.

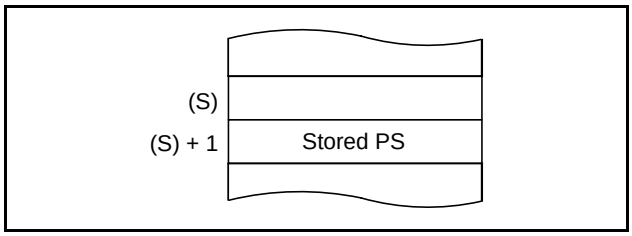


Fig. 102 Stack memory contents after PHP instruction execution

2. Decimal calculations

(1) Execution of decimal calculations

The ADC and SBC are the only instructions which will yield proper decimal notation, set the decimal mode flag (D) to "1" with the SED instruction. After executing the ADC or SBC instruction, execute another instruction before executing the SEC, CLC, or CLD instruction.

(2) Notes on status flag in decimal mode

When decimal mode is selected, the values of three of the flags in the status register (the N, V, and Z flags) are invalid after a ADC or SBC instruction is executed.

The carry flag (C) is set to "1" if a carry is generated as a result of the calculation, or is cleared to "0" if a borrow is generated. To determine whether a calculation has generated a carry, the C flag must be initialized to "0" before each calculation. To check for a borrow, the C flag must be initialized to "1" before each calculation.

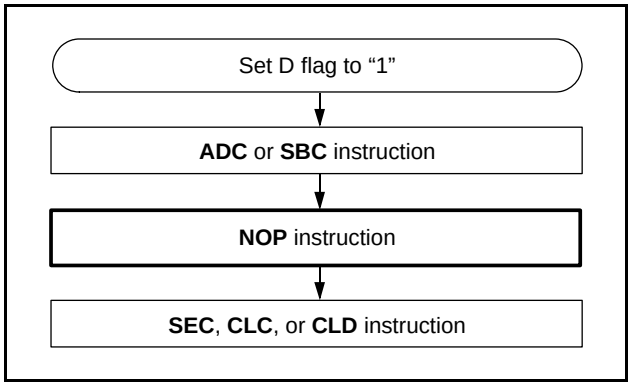


Fig. 103 Execution of decimal calculations

3. JMP instruction

When using the JMP instruction in indirect addressing mode, do not specify the last address on a page as an indirect address.

4. Multiplication and Division Instructions

- The index X mode (T) and the decimal mode (D) flags do not affect the MUL and DIV instruction.
- The execution of these instructions does not change the contents of the processor status register.

5. Ports

The contents of the port direction registers cannot be read. The following cannot be used:

- The data transfer instruction (LDA, etc.)
- The operation instruction when the index X mode flag (T) is "1"
- The instruction with the addressing mode which uses the value of a direction register as an index
- The bit-test instruction (BBC or BBS, etc.) to a direction register
- The read-modify-write instructions (ROR, CLB, or SEB, etc.) to a direction register.

Use instructions such as LDM and STA, etc., to set the port direction registers.

6. Instruction Execution Timing

The instruction execution time can be obtained by multiplying the frequency of the internal clock ϕ by the number of cycles mentioned in the 740 Family Software Manual.

The frequency of the internal clock ϕ is the twice the XIN cycle in high-speed mode, 8 times the XIN cycle in middle-speed mode, and the twice the XCIN in low-speed mode.

Countermeasures against noise

(1) Shortest wiring length

1. 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 20mm).

<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.

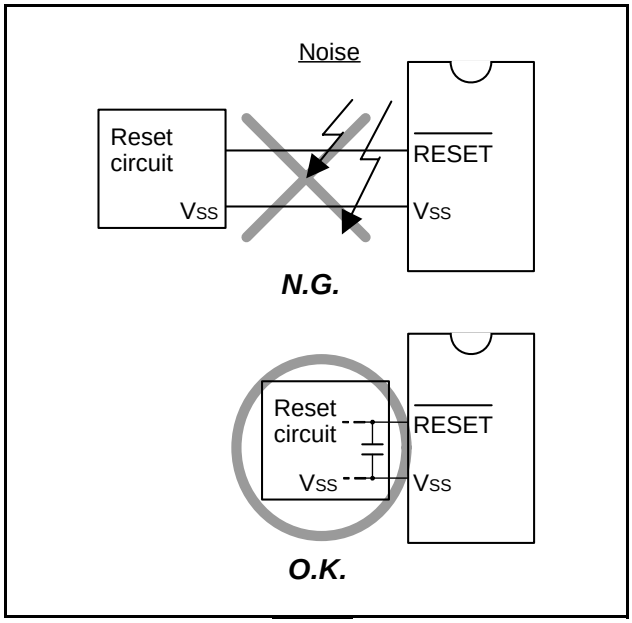


Fig. 104 Wiring for the $\overline{\text{RESET}}$ pin

2. 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.

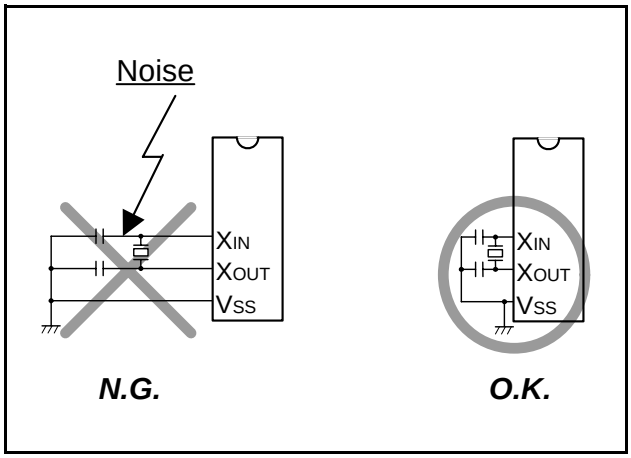


Fig. 105 Wiring for clock I/O pins

(2) Connection of bypass capacitor across Vss line and Vcc line
In order to stabilize the system operation and avoid the latch-up, 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.

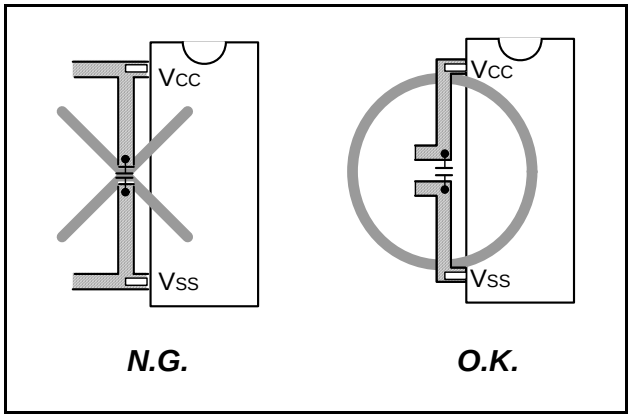


Fig. 106 Bypass capacitor across the Vss line and the Vcc line

(3) Oscillator concerns

In order to obtain the stabilized operation clock on the user system and its condition, contact the oscillator manufacturer and select the oscillator and oscillation circuit constants. Be careful especially when range of voltage and temperature is wide.
Also, 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.

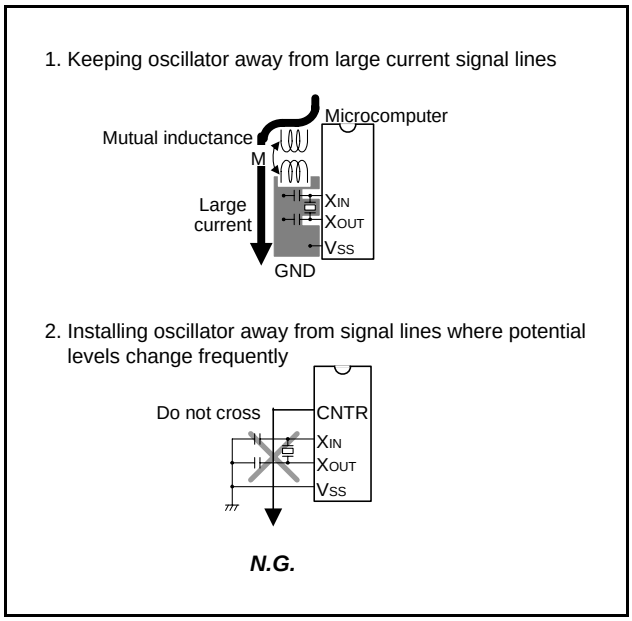


Fig. 107 Wiring for a large current signal line/Wiring of signal lines where potential levels change frequently

(4) Analog input

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.

(5) Difference of memory size

When memory size differ in one group, actual values such as an electrical characteristics, A/D conversion accuracy, and the amount of proof of noise incorrect operation may differ from the ideal values. When these products are used switching, perform system evaluation for each product of every after confirming product specification.

(6) Wiring to CNVSS pin

The CNVSS pin determines the flash memory mode.
Connect the CNVSS pin the shortest possible to the GND pattern which is supplied to the VSS pin of the microcomputer.
In addition connecting an approximately 5 kΩ resistor in series to the GND could improve noise immunity. In this case as well as the above mention, connect the pin the shortest possible to the GND pattern which is supplied to the VSS pin of the microcomputer.

Note. When the boot mode or the standard serial I/O mode is used, a switch of the input level to the CNVSS pin is required.

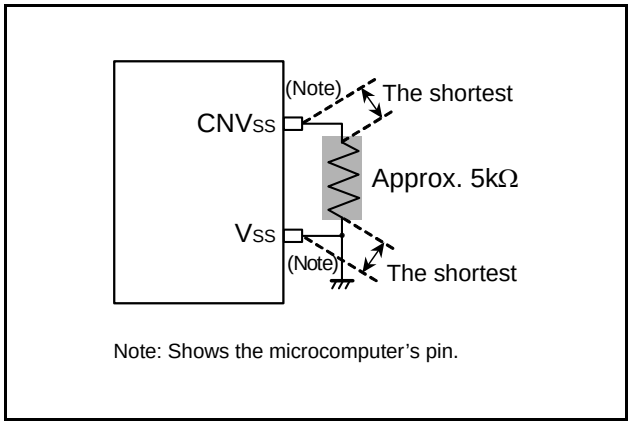


Fig. 108 Wiring for the CNVSS

NOTES ON PERIPHERAL FUNCTIONS

Notes on Input and Output Ports

1. Notes in standby state

In standby state*1 for low-power dissipation, do not make input levels of an I/O port "undefined". Even when an I/O port of N-channel open-drain is set as output mode, if output data is "1", the aforementioned notes are necessary.
Pull-up (connect the port to VCC) or pull-down (connect the port to VSS) these ports through a resistor.
When determining a resistance value, note the following points:
- External circuit
- Variation of output levels during the ordinary operation

When using built-in pull-up resistor, note on varied current values:
- When setting as an input port : Fix its input level
- When setting as an output port : Prevent current from flowing out to external

<Reason>
Exclusive input ports are always in a high-impedance state. An output transistor becomes an OFF state when an I/O port is set as input mode by the direction register, so that the port enter a high-impedance state. At this time, the potential which is input to the input buffer in a microcomputer is unstable in the state that input levels are "undefined". This may cause power source current.
Even when an I/O port of N-channel open-drain is set as output mode by the direction register, if the contents of the port latch is "1", the same phenomenon as that of an input port will occur.

*1 Standby state : stop mode by executing STP instruction
wait mode by executing WIT instruction

2. Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*1, the value of the unspecified bit may be changed.
<Reason>
I/O ports are set to input or output mode in bit units. Reading from a port register or writing to it involves the following operations.
- Port in input mode
  Read: Read the pin level.
  Write: Write to the port latch.
- Port in output mode
  Read: Read the port latch or read the output from the peripheral function (specifications differ depending on the port).
  Write: Write to the port latch. (The port latch value is output from the pin.)

Since bit managing instructions*1 are read-modify-write instructions,*2 using such an instruction on a port register causes a read and write to be performed simultaneously on the bits other than the one specified by the instruction.
When an unspecified bit is in input mode, its pin level is read and that value is written to the port latch. If the previous value of the port latch differs from the pin level, the port latch value is changed.
If an unspecified bit is in output mode, the port latch is generally read. However, for some ports the peripheral function output is read, and the value is written to the port latch. In this case, if the previous value of the port latch differs from the peripheral function output, the port latch value is changed.

*1 Bit managing instructions: SEB and CLB instructions
*2 Read-modify-write instructions: Instructions that read memory in byte units, modify the value, and then write the result to the same location in memory in byte units

Termination of Unused Pins

1. Terminate unused pins

(1) Output ports : Open

(2) I/O ports :
- Set the I/O ports for the input mode and connect them to VCC or VSS through each resistor of 1 kΩ to 10 kΩ.
  Ports that permit the selecting of a built-in pull-up resistor can also use this resistor. Set the I/O ports for the output mode and 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.

(3) The AVSS pin when not using the A/D converter :
- When not using the A/D converter, handle a power source pin for the A/D converter, AVSS pin as follows:
  AVSS: Connect to the VSS pin.

2. Termination remarks

(1) I/O ports :
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 (2) in 1 and shown on the above.

(2) I/O ports :
When setting for the input mode, do not connect to VCC or VSS directly.
<Reason>
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between a port and VCC (or VSS).

(3) I/O ports :
When setting for the input mode, do not connect multiple ports in a lump to VCC or VSS 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.

- At the termination of unused pins, perform wiring at the shortest possible distance (20 mm or less) from micro-computer pins.

Notes on Interrupts

1. Change of relevant register settings

When the setting of the following registers or bits is changed, the interrupt request bit may be set to “1”. When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

- Interrupt edge selection register (address $003A_{16}$)
- Timer XY mode register (address 0023_{16})
- Timer Z mode register (address $002A_{16}$)

Set the above listed registers or bits as the following sequence.

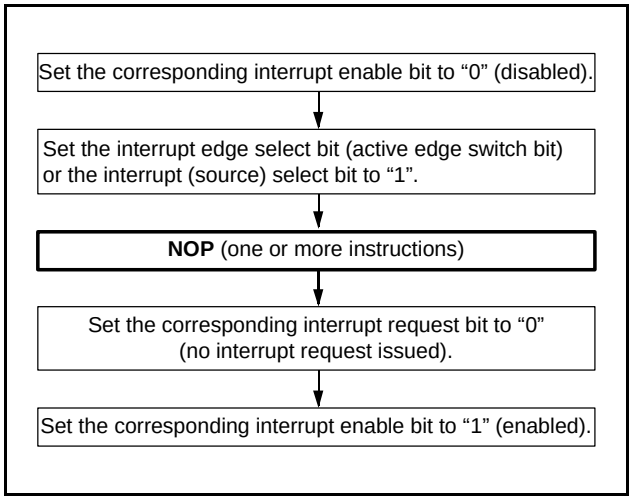


Fig. 109 Sequence of changing relevant register

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

- When setting external interrupt active edge
  Concerned register: Interrupt edge selection register (address $003A_{16}$)
  Timer XY mode register (address 0023_{16})
  Timer Z mode register (address $002A_{16}$)
- When switching interrupt sources of an interrupt vector address where two or more interrupt sources are allocated.
  Concerned register: Interrupt source selection register (address 0039_{16})

2. Check of interrupt request bit

When executing the BBC or BBS instruction to an interrupt request bit of an interrupt request register immediately after this bit is set to “0”, execute one or more instructions before executing the BBC or BBS instruction.

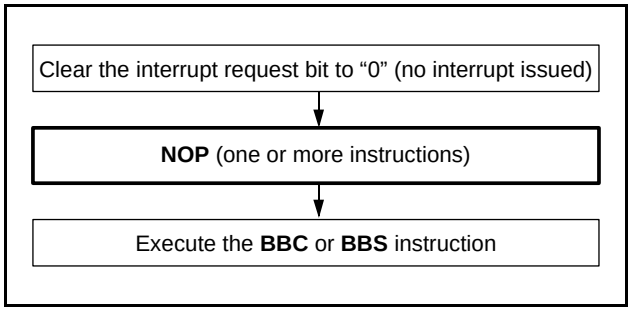


Fig. 110 Sequence of check of interrupt request bit

<Reason>
If the BBC or BBS instruction is executed immediately after an interrupt request bit of an interrupt request register is cleared to “0”, the value of the interrupt request bit before being cleared to “0” is read.

Notes on 8-bit Timer (timer 1, 2, X, Y)

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).
- When switching the count source by the timer 12, X and Y count source selection bits, the value of timer count is altered in unconsiderable amount owing to generating of thin pulses in the count input signals.
  Therefore, select the timer count source before set the value to the prescaler and the timer.
- Set the double-function port of the $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to output in the pulse output mode.
- Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to input in the event counter mode and the pulse width measurement mode.

Notes on 16-bit Timer (timer Z)

1. Pulse output mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to output.

2. Pulse period measurement mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to input.
- A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse period).
- Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.
- "$FFFF_{16}$" is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected.
  Consequently, the timer value at start of pulse period measurement depends on the timer value just before measurement start.

3. Pulse width measurement mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to input.
- A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse period).
- Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.
- "$FFFF_{16}$" is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected.
  Consequently, the timer value at start of pulse width measurement depends on the timer value just before measurement start.

4. Programmable waveform generating mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to output.

5. Programmable one-shot generating mode

- Set the double-function port of $CNTR_2$ pin and port $P4_7$ to output, and of INT_1 pin and port $P4_2$ to input in this mode.
- This mode cannot be used in low-speed mode.
- If the value of the $CNTR_2$ active edge switch bit is changed during one-shot generating enabled or generating one-shot pulse, then the output level from $CNTR_2$ pin changes.

6. All modes

- Timer Z write control

Which write control can be selected by the timer Z write control bit (bit 3) of the timer Z mode register (address $002A_{16}$), writing data to both the latch and the timer at the same time or writing data only to the latch.
When the operation "writing data only to the latch" is selected, the value is set to the timer latch by writing data to the address of timer Z and the timer is updated at next underflow. After reset release, the operation "writing data to both the latch and the timer at the same time" is selected, and the value is set to both the latch and the timer at the same time by writing data to the address of timer Z.
In the case of writing data only to the latch, if writing data to the latch and an underflow are performed almost at the same time, the timer value may become undefined.

- Timer Z read control

A read-out of timer value is impossible in pulse period measurement mode and pulse width measurement mode. In the other modes, a read-out of timer value is possible regardless of count operating or stopped.
However, a read-out of timer latch value is impossible.

- Switch of interrupt active edge of $CNTR_2$ and INT_1

Each interrupt active edge depends on setting of the $CNTR_2$ active edge switch bit and the INT_1 active edge selection bit.

- Switch of count source

When switching the count source by the timer Z count source selection bits, the value of timer count is altered in inconsiderable amount owing to generating of thin pulses on the count input signals.
Therefore, select the timer count source before setting the value to the prescaler and the timer.

Notes on Serial Interface

1. Notes when selecting clock synchronous serial I/O

(1) Stop of transmission operation

As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the serial I/Oi enable bit and the transmit enable bit to "0" (serial I/Oi and transmit disabled).

<Reason>

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/Oi enable bit is cleared to "0" (serial I/Oi disabled), the internal transmission is running (in this case, since pins TxDi, RxDi, SCLKi, and $\overline{\text{SRDYi}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/Oi enable bit is set to "1" at this time, the data during internally shifting is output to the TxDi pin and an operation failure occurs.

(2) Stop of receive operation

As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the receive enable bit to "0" (receive disabled), or clear the serial I/Oi enable bit to "0" (serial I/Oi disabled).

(3) Stop of transmit/receive operation

As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear both the transmit enable bit and receive enable bit to "0" (transmit and receive disabled).

(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

<Reason>

In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.

In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to "0" (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/Oi enable bit to "0" (serial I/Oi disabled) (refer to (1) in 1.).

2. Notes when selecting clock asynchronous serial I/O

(1) Stop of transmission operation

Clear the transmit enable bit to "0" (transmit disabled). The transmission operation does not stop by clearing the serial I/Oi enable bit (i = 1, 3) to "0".

<Reason>

This is the same as (1) in 1.

(2) Stop of receive operation

Clear the receive enable bit to "0" (receive disabled).

(3) Stop of transmit/receive operation

Only transmission operation is stopped.

Clear the transmit enable bit to "0" (transmit disabled). The transmission operation does not stop by clearing the serial I/Oi enable bit (i = 1, 3) to "0".

<Reason>

This is the same as (1) in 1.

Only receive operation is stopped.

Clear the receive enable bit to "0" (receive disabled).

3. $\overline{\text{SRDYi}}$ (i = 1, 3) output of reception side

When signals are output from the $\overline{\text{SRDYi}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDYi}}$ output enable bit, and the transmit enable bit to "1" (transmit enabled).

4. Setting serial I/Oi (i = 1, 3) control register again

Set the serial I/Oi control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to "0".

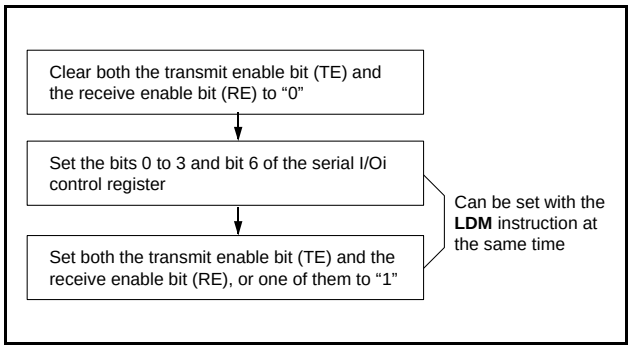


Fig. 111 Sequence of setting serial I/Oi (i = 1, 3) control register again

5. Data transmission control with referring to transmit shift register completion flag

After the transmit data is written to the transmit buffer register, the transmit shift register completion flag changes from "1" to "0" with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

6. Transmission control when external clock is selected

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to "1" at "H" of the SCLKi (i = 1, 3) input level. Also, write the transmit data to the transmit buffer register at "H" of the SCLKi input level.

7. Transmit interrupt request when transmit enable bit is set

When using the transmit interrupt, take the following sequence.

(1) Set the serial I/Oi transmit interrupt enable bit (i = 1, 3) to "0" (disabled).
(2) Set the transmit enable bit to "1".
(3) Set the serial I/Oi transmit interrupt request bit (i = 1, 3) to "0" after 1 or more instruction has executed.
(4) Set the serial I/Oi transmit interrupt enable bit (i = 1, 3) to "1" (enabled).

<Reason>

When the transmission enable bit is set to "1", the transmit buffer empty flag and transmit shift register shift completion flag are also set to "1".

Therefore, regardless of selecting which timing for the generating of transmit interrupts, the interrupt request is generated and the transmit interrupt request bit is set at this point.

8. Writing to baud rate generator i (BRGi) (i = 1, 3)

Write data to the baud rate generator i (BRGi) (i = 1, 3) while the transmission/reception operation is stopped.

Notes on PWM

The PWM starts from "H" level after the PWM enable bit is set to enable and "L" level is temporarily output from the PWM pin. The length of this "L" level output is as follows:

$\frac{n + 1}{2 \times f(X_{IN})}$ (s) (Count source selection bit = "0", where n is the value set in the prescaler)

$\frac{n + 1}{f(X_{IN})}$ (s) (Count source selection bit = "1", where n is the value set in the prescaler)

Notes on A/D Converter

1. Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01 μF to 1 μF. Further, be sure to verify the operation of application products on the user side.

<Reason>

An analog input pin includes the capacitor for analog voltage comparison. Accordingly, when signals from signal source with high impedance are input to an analog input pin, charge and discharge noise generates. This may cause the A/D conversion precision to be worse.

2. A/D converter power source pin

The AV_{SS} pin is A/D converter power source pins. Regardless of using the A/D conversion function or not, connect it as following :

- AV_{SS} : Connect to the V_{SS} line

<Reason>

If the AV_{SS} pin is opened, the microcomputer may have a failure because of noise or others.

3. Clock frequency during A/D conversion

The comparator consists of a capacity coupling, and a charge of the capacity will be lost if the clock frequency is too low. Thus, make sure the following during an A/D conversion.

- $f(X_{IN})$ is 500 kHz or more
- Do not execute the STP instruction

4. Difference between at 8-bit reading in 10-bit A/D mode and at 8-bit A/D mode

At 8-bit reading in the 10-bit A/D mode, "–1/2 LSB" correction is not performed to the A/D conversion result.

In the 8-bit A/D mode, the A/D conversion characteristics is the same as 3802 group's characteristics because "–1/2 LSB" correction is performed.

Notes on D/A Converter

1. Vcc when using D/A converter

The D/A converter accuracy when V_{CC} is 4.0 V or less differs from that of when V_{CC} is 4.0 V or more. When using the D/A converter, we recommend using a V_{CC} of 4.0 V or more.

2. DAi conversion register when not using D/A converter

When a D/A converter is not used, set all values of the DAi conversion registers (i = 1, 2) to "00_{16}". The initial value after reset is "00_{16}".

Notes on Watchdog Timer

- Make sure that the watchdog timer H does not underflow while waiting Stop release, because the watchdog timer keeps counting during that term.
- When the STP instruction disable bit has been set to "1", it is impossible to switch it to "0" by a program.

Notes on $\overline{\text{RESET}}$ Pin

Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the V_{SS} pin.

Use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to verify the operation of application products on the user side.

<Reason>

If the several nanosecond or several ten nanosecond impulse noise enters the $\overline{\text{RESET}}$ pin, it may cause a microcomputer failure.

Notes on Low-speed Operation Mode

1. Using sub-clock

To use a sub-clock, fix bit 3 of the CPU mode register to "1" or control the Rd (refer to Figure 112) resistance value to a certain level to stabilize an oscillation. For resistance value of Rd, consult the oscillator manufacturer.

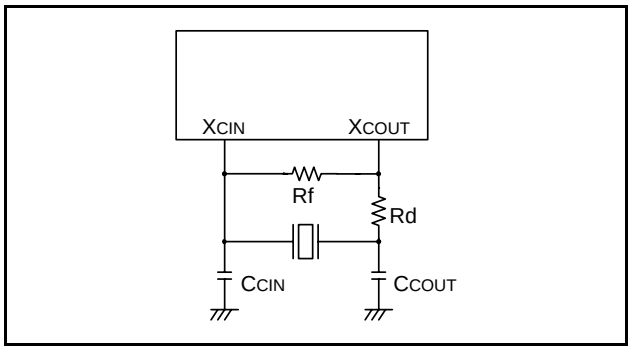


Fig. 112 Ceramic resonator circuit

<Reason>

When bit 3 of the CPU mode register is set to "0", the sub-clock oscillation may stop.

2. Switch between middle/high-speed mode and low-speed mode

If you switch the mode between middle/high-speed and low-speed, stabilize both X_{IN} and X_{CIN} oscillations. The sufficient time is required for the sub clock to stabilize, especially immediately after power on and at returning from stop mode. When switching the mode between middle/high-speed and low-speed, set the frequency on condition that $f(X_{IN}) > 3 \times f(X_{CIN})$.

Quartz-Crystal Oscillator

When using the quartz-crystal oscillator of high frequency, such as 16 MHz etc., it may be necessary to select a specific oscillator with the specification demanded.

Notes on Restarting Oscillation

• Restarting oscillation

Usually, when the MCU stops the clock oscillation by STP instruction and the STP instruction has been released by an external interrupt source, the fixed values of Timer 1 and Prescaler 12 (Timer 1 = "01$_{16}$", Prescaler 12 = "FF$_{16}$") are automatically reloaded in order for the oscillation to stabilize. The user can inhibit the automatic setting by writing "1" to bit 0 of MISRG (address 0010$_{16}$).

However, by setting this bit to "1", the previous values, set just before the STP instruction was executed, will remain in Timer 1 and Prescaler 12. Therefore, you will need to set an appropriate value to each register, in accordance with the oscillation stabilizing time, before executing the STP instruction.

<Reason>

Oscillation will restart when an external interrupt is received. However, internal clock ϕ is supplied to the CPU only when Timer 1 starts to underflow. This ensures time for the clock oscillation using the ceramic resonators to be stabilized.

Notes on Using Stop Mode

• Register setting

Since values of the prescaler 12 and Timer 1 are automatically reloaded when returning from the stop mode, set them again, respectively. (When the oscillation stabilizing time set after STP instruction released bit is "0")

• Clock restoration

After restoration from the stop mode to the normal mode by an interrupt request, the contents of the CPU mode register previous to the STP instruction execution are retained. Accordingly, if both main clock and sub clock were oscillating before execution of the STP instruction, the oscillation of both clocks is resumed at restoration.

In the above case, when the main clock side is set as a system clock, the oscillation stabilizing time for approximately 8,000 cycles of the XIN input is reserved at restoration from the stop mode. At this time, note that the oscillation on the sub clock side may not be stabilized even after the lapse of the oscillation stabilizing time of the main clock side.

Notes on Wait Mode

• Clock restoration

If the wait mode is released by a reset when XCIN is set as the system clock and XIN oscillation is stopped during execution of the WIT instruction, XCIN oscillation stops, XIN oscillations starts, and XIN is set as the system clock.

In the above case, the $\overline{\text{RESET}}$ pin should be held at "L" until the oscillation is stabilized.

Notes on CPU rewrite mode of flash memory version

1. Operation speed

During CPU rewrite mode, set the system clock ϕ 4.0 MHz or less using the main clock division ratio selection bits (bits 6 and 7 of address 003B$_{16}$).

2. Instructions inhibited against use

The instructions which refer to the internal data of the flash memory cannot be used during the CPU rewrite mode.

3. Interrupts inhibited against use

The interrupts cannot be used during the CPU rewrite mode because they refer to the internal data of the flash memory.

4. Watchdog timer

In case of the watchdog timer has been running already, the internal reset generated by watchdog timer underflow does not happen, because of watchdog timer is always clearing during program or erase operation.

5. Reset

Reset is always valid. In case of CNVSS = "H" when reset is released, boot mode is active. So the program starts from the address contained in address FFFC$_{16}$ and FFFD$_{16}$ in boot ROM area.

Notes on flash memory version

The CNVSS pin determines the flash memory mode.

Connect the CNVSS pin the shortest possible to the GND pattern which is supplied to the VSS pin of the microcomputer.

In addition connecting an approximately 5 kΩ resistor in series to the GND could improve noise immunity. In this case as well as the above mention, connect the pin the shortest possible to the GND pattern which is supplied to the VSS pin of the microcomputer.

Note. When the boot mode or the standard serial I/O mode is used, a switch of the input level to the CNVSS pin is required.

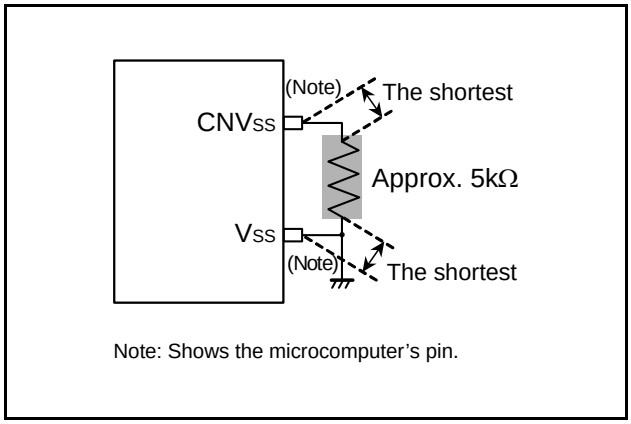


Fig. 113 Wiring for the CNVSS

Notes on Handling of Power Source Pins

In order to avoid a latch-up occurrence, connect a capacitor suitable for high frequencies as bypass capacitor between power source pin (VCC pin) and GND pin (VSS pin), and between power source pin (VCC pin) and analog power source input pin (AVSS pin). Besides, connect the capacitor to as close as possible. For bypass capacitor which should not be located too far from the pins to be connected, a ceramic capacitor of 0.01 μF–0.1 μF is recommended.

Power Source Voltage

When the power source voltage value of a microcomputer is less than the value which is indicated as the recommended operating conditions, the microcomputer does not operate normally and may perform unstable operation.

In a system where the power source voltage drops slowly when the power source voltage drops or the power supply is turned off, reset a microcomputer when the power source voltage is less than the recommended operating conditions and design a system not to cause errors to the system by this unstable operation.

ELECTRICAL CHARACTERISTICS

Absolute maximum ratings

Table 22 Absolute maximum ratings

Symbol	Parameter		Conditions	Ratings	Unit
V_{CC}	Power source voltages		All voltages are based on V_{SS}. When an input voltage is measured, output transistors are cut off.	–0.3 to 6.5	V
V_I	Input voltage	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, V_{REF}		–0.3 to V_{CC} + 0.3	V
V_I	Input voltage	$P3_2$, $P3_3$		–0.3 to 5.8	V
V_I	Input voltage	$\overline{RESET}$, X_{IN}		–0.3 to V_{CC} + 0.3	V
V_I	Input voltage	CNV_{SS}		–0.3 to V_{CC} + 0.3	V
V_O	Output voltage	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, X_{OUT}		–0.3 to V_{CC} + 0.3	V
V_O	Output voltage	$P3_2$, $P3_3$		–0.3 to 5.8	V
P_d	Power dissipation		Ta=25 °C	1000(1)	mW
T_{opr}	Operating temperature		—	–20 to 85	°C
T_{stg}	Storage temperature		—	–65 to 125	°C

NOTE:
1. This value is 300 mW except SP package.

Recommended operating conditions

Table 23 Recommended operating conditions (1)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{CC}	Power source voltage(1)	When start oscillating(2)		2.7	5.0	5.5	V
		High-speed mode $f(\phi) = f(X_{IN})/2$	$f(X_{IN}) \le 8.4$ MHz	2.7	5.0	5.5	V
			$f(X_{IN}) \le 12.5$ MHz	4.0	5.0	5.5	
			$f(X_{IN}) \le 16.8$ MHz	4.5	5.0	5.5	
		Middle-speed mode $f(\phi) = f(X_{IN})/8$	$f(X_{IN}) \le 12.5$ MHz	2.7	5.0	5.5	V
			$f(X_{IN}) \le 16.8$ MHz	4.5	5.0	5.5	
V_{SS}	Power source voltage				0		V
V_{IH}	“H” input voltage $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			0.8 V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage $P3_2$, $P3_3$			0.8 V_{CC}		5.5	V
V_{IH}	“H” input voltage (when I^2C-BUS input level is selected) SDA, SCL			0.7 V_{CC}		5.5	V
V_{IH}	“H” input voltage (when SMBUS input level is selected) SDA, SCL			1.4		5.5	V
V_{IH}	“H” input voltage $\overline{RESET}$, X_{IN}, CNV_{SS}			0.8 V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage X_{CIN}			2		V_{CC}	V
V_{IL}	“L” input voltage $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			0		0.2 V_{CC}	V
V_{IL}	“L” input voltage (when I^2C-BUS input level is selected) SDA, SCL			0		0.3 V_{CC}	V
V_{IL}	“L” input voltage (when SMBUS input level is selected) SDA, SCL			0		0.6	V
V_{IL}	“L” input voltage $\overline{RESET}$, CNV_{SS}			0		0.2 V_{CC}	V
V_{IL}	“L” input voltage X_{IN}					0.16 V_{CC}	V
V_{IL}	“L” input voltage X_{CIN}					0.4	V
$f(X_{IN})$	Main clock input oscillation frequency(3)	High-speed mode $f(\phi) = f(X_{IN})/2$	2.7 ≤ V_{CC} < 4.0 V			$\frac{(9 \times V_{CC} - 0.3) \times 1.05}{3}$	MHz
			4.0 ≤ V_{CC} < 4.5 V			$\frac{(24 \times V_{CC} - 60) \times 1.05}{3}$	MHz
			4.5 ≤ V_{CC} ≤ 5.5 V			16.8	MHz
		Middle-speed mode $f(\phi) = f(X_{IN})/8$	2.7 ≤ V_{CC} < 4.5 V			$\frac{(15 \times V_{CC} + 39) \times 1.1}{7}$	MHz
			4.5 ≤ V_{CC} ≤ 5.5 V			16.8	MHz
$f(X_{CIN})$	Sub-clock input oscillation frequency(3, 4)				32.768	50	kHz

NOTES:
1. When using A/D converter, see A/D converter recommended operating conditions.
2. The start voltage and the start time for oscillation depend on the using oscillator, oscillation circuit constant value and operating temperature range, etc.. Particularly a high-frequency oscillator might require some notes in the low voltage operation.
3. When the oscillation frequency has a duty cycle of 50%.
4. When using the microcomputer in low-speed mode, set the sub-clock input oscillation frequency on condition that $f(X_{CIN}) < f(X_{IN})/3$.

Table 24 Recommended operating conditions (2)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$\Sigma I_{OH(peak)}$	"H" total peak output current(1)	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$			–80	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current(1)	$P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			–80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current(1)	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P3_0$-$P3_7$			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current(1)	$P2_0$-$P2_7$			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current(1)	$P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			80	mA
$\Sigma I_{OH(avg)}$	"H" total average output current(1)	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$			–40	mA
$\Sigma I_{OH(avg)}$	"H" total average output current(1)	$P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			–40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current(1)	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P3_0$-$P3_7$			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current(1)	$P2_0$-$P2_7$			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current(1)	$P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			40	mA
$I_{OH(peak)}$	"H" peak output current(2)	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			–10	mA
$I_{OL(peak)}$	"L" peak output current(2)	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			10	mA
$I_{OL(peak)}$	"L" peak output current(2)	$P2_0$-$P2_7$			20	mA
$I_{OH(avg)}$	"H" average output current(3)	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			–5	mA
$I_{OL(avg)}$	"L" average output current(3)	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$			5	mA
$I_{OL(avg)}$	"L" average output current(3)	$P2_0$-$P2_7$			10	mA

NOTES:
1. The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100 ms. The total peak current is the peak value of all the currents.
2. The peak output current is the peak current flowing in each port.
3. The average output current I_{OL}(avg), I_{OH}(avg) are average value measured over 100 ms.

Electrical characteristics

Table 25 Electrical characteristics (1)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	"H" output voltage(1) $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$	I_{OH} = –10 mA V_{CC} = 4.0 to 5.5 V	V_{CC} – 2.0			V
		I_{OH} = –1.0 mA V_{CC} = 2.7 to 5.5 V	V_{CC} – 1.0			
V_{OL}	"L" output voltage $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$	I_{OL} = 10 mA V_{CC} = 4.0 to 5.5 V			2.0	V
		I_{OL} = 1.6 mA V_{CC} = 2.7 to 5.5 V			1.0	
V_{OL}	"L" output voltage $P2_0$-$P2_7$	I_{OL} = 20 mA V_{CC} = 4.0 to 5.5 V			2.0	V
		I_{OL} = 1.6 mA V_{CC} = 2.7 to 5.5 V			0.4	
$V_{T+} – V_{T–}$	Hysteresis $CNTR_0$, $CNTR_1$, $CNTR_2$, INT_0-INT_4			0.4		V
$V_{T+} – V_{T–}$	Hysteresis RxD_1, $SCLK_1$, SIN_2, $SCLK_2$, RxD_3, $SCLK_3$			0.5		V
$V_{T+} – V_{T–}$	Hysteresis $\overline{RESET}$			0.5		V
I_{IH}	"H" input current $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$	V_I = V_{CC} (Pin floating, Pull-up transistor "off")			5.0	µA
I_{IH}	"H" input current $\overline{RESET}$, CNV_{SS}	V_I = V_{CC}			5.0	µA
I_{IH}	"H" input current X_{IN}	V_I = V_{CC}		4.0		µA
I_{IL}	"L" input current $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$	V_I = V_{SS} (Pin floating, Pull-up transistor "off")			–5.0	µA
I_{IL}	"L" input current $\overline{RESET}$, CNV_{SS}	V_I = V_{SS}			–5.0	µA
I_{IL}	"L" input current X_{IN}	V_I = V_{SS}		–4.0		µA
I_{IL}	"L" input current (at Pull-up) $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$, $P3_1$, $P3_4$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$	V_I = V_{SS} V_{CC} = 5.0 V	–80	–210	–420	µA
		V_I = V_{SS} V_{CC} = 3.0 V	–30	–70	–140	
V_{RAM}	RAM hold voltage	When clock stopped	1.8		V_{CC}	V

NOTE:
1. $P3_5$ is measured when the $P3_5/TxD_3$ P-channel output disable bit of the UART3 control register (bit 4 of address 0033_{16}) is "0".
$P4_5$ is measured when the $P4_5/TxD_1$ P-channel output disable bit of the UART1 control register (bit 4 of address $001B_{16}$) is "0".

Table 26 Electrical characteristics (2)
(V_{CC} = 2.7 to 5.5 V, Ta = –20 to 85 °C, f(X_{CIN})=32.768 kHz (Stopped in middle-speed mode), Output transistors “off”, AD converter not operated)

Symbol	Parameter	Test conditions			Limits			Unit
					Min.	Typ.	Max.	
I_{CC}	Power source current	High-speed mode	V_{CC} = 5.0 V	f(X_{IN}) = 16.8 MHz		5.5	8.3	mA
				f(X_{IN}) = 12.5 MHz		4.5	6.8	
				f(X_{IN}) = 8.4 MHz		3.5	5.3	
				f(X_{IN}) = 4.2 MHz		2.2	3.3	
				f(X_{IN}) = 16.8 MHz (in WIT state)		2.2	3.3	
			V_{CC} = 3.0 V	f(X_{IN}) = 8.4 MHz		2.7	4.1	mA
				f(X_{IN}) = 4.2 MHz		1.8	2.7	
				f(X_{IN}) = 2.1 MHz		1.1	1.7	
		Middle-speed mode	V_{CC} = 5.0 V	f(X_{IN}) = 16.8 MHz		3.0	4.5	mA
				f(X_{IN}) = 12.5 MHz		2.4	3.6	
				f(X_{IN}) = 8.4 MHz		2.0	3.0	
				f(X_{IN}) = 16.8 MHz (in WIT state)		2.1	3.2	
			V_{CC} = 3.0 V	f(X_{IN}) = 12.5 MHz		1.7	2.6	mA
				f(X_{IN}) = 8.4 MHz		1.5	2.3	
				f(X_{IN}) = 6.3 MHz		1.3	2.0	
		Low-speed mode	V_{CC} = 5.0 V	f(X_{IN}) = stopped		410	630	µA
				In WIT state		4.5	6.8	
			V_{CC} = 3.0 V	f(X_{IN}) = stopped		400	600	µA
				In WIT state		3.7	5.6	
		In STP state (All oscillation stopped)		Ta = 25 °C		0.55	3.0	µA
				Ta = 85 °C		0.75		
		Increment when A/D conversion is executed		f(X_{IN}) = 16.8 MHz, V_{CC} = 5.0 V In Middle-, high-speed mode		1000		µA

A/D converter characteristics

Table 27 A/D converter recommended operating conditions
(V_{CC} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage (When A/D converter is used)	8-bit A/D mode(1)	2.7	5.0	5.5	V
		10-bit A/D mode(2)	2.7	5.0	5.5	
V_{REF}	Analog convert reference voltage		2.0		V_{CC}	V
AV_{SS}	Analog power source voltage			0		V
V_{IA}	Analog input voltage AN_0-AN_{15}		0		V_{CC}	V
f(X_{IN})	Main clock input oscillation frequency (When A/D converter is used)	2.7 ≤ V_{CC} = V_{REF} < 4.0 V	0.5		$\frac{(9 \times V_{CC} - 0.3) \times 1.05}{3}$	MHz
		4.0 ≤ V_{CC} = V_{REF} < 4.5 V	0.5		$\frac{(24.6 \times V_{CC} - 62.7) \times 1.05}{3}$	
		4.5 ≤ V_{CC} = V_{REF} ≤ 5.5 V	0.5		16.8	

NOTES:
1. 8-bit A/D mode: When the conversion mode selection bit (bit 7 of address 0038_{16}) is “1”.
2. 10-bit A/D mode: When the conversion mode selection bit (bit 7 of address 0038_{16}) is “0”.

Table 28 A/D converter characteristics
(V_{CC} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
–	Resolution	8-bit A/D mode(1)				8	bit
		10-bit A/D mode(2)				10	
–	Absolute accuracy (excluding quantization error)	8-bit A/D mode(1)	2.7 ≤ V_{REF} ≤ 5.5 V			±2	LSB
		10-bit A/D mode(2)	2.7 ≤ V_{REF} ≤ 5.5 V			±4	LSB
t_{CONV}	Conversion time	8-bit A/D mode(1)				50	2tc(X_{IN})
		10-bit A/D mode(2)				61	
R_{LADDER}	Ladder resistor			12	35	100	kΩ
I_{VREF}	Reference power source input current	at A/D converter operated	V_{REF} = 5.0 V	50	150	200	μA
		at A/D converter stopped	V_{REF} = 5.0 V			5.0	μA
$I_{I(AD)}$	A/D port input current					5.0	μA

NOTES:
1. 8-bit A/D mode: When the conversion mode selection bit (bit 7 of address 0038_{16}) is “1”.
2. 10-bit A/D mode: When the conversion mode selection bit (bit 7 of address 0038_{16}) is “0”.

D/A converter characteristics

Table 29 D/A converter characteristics
(V_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 V to V_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
–	Resolution				8	bit
–	Absolute accuracy	4.0 ≤ V_{REF} ≤ 5.5 V			1.0	%
		2.7 ≤ V_{REF} < 4.0 V			2.5	
tsu	Setting time				3	μs
RO	Output resistor		2	3.5	5	kΩ
I_{VREF}	Reference power source input current(1)				3.2	mA

NOTE:
1. Using one D/A converter, with the value in the DA conversion register of the other D/A converter being “00_{16}”.

Power source circuit timing characteristics

Table 30 Power source circuit timing characteristics
(V_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 V to V_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
td(P–R)	Internal power source stable time at power-on	2.7 ≤ V_{CC} < 5.5 V			2	ms

Timing requirements and switching characteristics

Table 31 Timing requirements (1)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width		td(P-R)ms + 16			X_{IN} cycle
$t_{C(XIN)}$	Main clock X_{IN} input cycle time	4.5 ≤ V_{CC} ≤ 5.5 V	59.5			ns
		4.0 ≤ V_{CC} < 4.5 V	10000/(86 V_{CC} – 219)			
		2.7 ≤ V_{CC} < 4.0 V	26 × 10^3/(82 V_{CC} – 3)			
$t_{WH(XIN)}$	Main clock X_{IN} input “H” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V	25			ns
		4.0 ≤ V_{CC} < 4.5 V	4000/(86 V_{CC} – 219)			
		2.7 ≤ V_{CC} < 4.0 V	10000/(82 V_{CC} – 3)			
$t_{WL(XIN)}$	Main clock X_{IN} input “L” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V	25			ns
		4.0 ≤ V_{CC} < 4.5 V	4000/(86 V_{CC} – 219)			
		2.7 ≤ V_{CC} < 4.0 V	10000/(82 V_{CC} – 3)			
$t_{C(XCIN)}$	Sub-clock X_{CIN} input cycle time		20			µs
$t_{WH(XCIN)}$	Sub-clock X_{CIN} input “H” pulse width		5			µs
$t_{WL(XCIN)}$	Sub-clock X_{CIN} input “L” pulse width		5			µs
$t_{C(CNTR)}$	$CNTR_0$–$CNTR_2$ input cycle time	4.5 ≤ V_{CC} ≤ 5.5 V	120			ns
		4.0 ≤ V_{CC} < 4.5 V	160			
		2.7 ≤ V_{CC} < 4.0 V	250			
$t_{WH(CNTR)}$	$CNTR_0$–$CNTR_2$ input “H” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V	48			ns
		4.0 ≤ V_{CC} < 4.5 V	64			
		2.7 ≤ V_{CC} < 4.0 V	115			
$t_{WL(CNTR)}$	$CNTR_0$–$CNTR_2$ input “L” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V	48			ns
		4.0 ≤ V_{CC} < 4.5 V	64			
		2.7 ≤ V_{CC} < 4.0 V	115			
$t_{WH(INT)}$	INT_{00}, INT_{01}, INT_1, INT_2, INT_3, INT_{40}, INT_{41} input “H” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V	48			ns
		4.0 ≤ V_{CC} < 4.5 V	64			
		2.7 ≤ V_{CC} < 4.0 V	115			
$t_{WL(INT)}$	INT_{00}, INT_{01}, INT_1, INT_2, INT_3, INT_{40}, INT_{41} input “L” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V	48			ns
		4.0 ≤ V_{CC} < 4.5 V	64			
		2.7 ≤ V_{CC} < 4.0 V	115			

Table 32 Timing requirements (2)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, Ta = −20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$t_{C(SCLK1)}$ $t_{C(SCLK3)}$	Serial I/O1, serial I/O3 clock input cycle time(1)	4.5 ≤ V_{CC} ≤ 5.5 V	250			ns
		4.0 ≤ V_{CC} < 4.5 V	320			
		2.7 ≤ V_{CC} < 4.0 V	500			
$t_{WH(SCLK1)}$ $t_{WH(SCLK3)}$	Serial I/O1, serial I/O3 clock input “H” pulse width(1)	4.5 ≤ V_{CC} ≤ 5.5 V	120			ns
		4.0 ≤ V_{CC} < 4.5 V	150			
		2.7 ≤ V_{CC} < 4.0 V	240			
$t_{WL(SCLK1)}$ $t_{WL(SCLK3)}$	Serial I/O1, serial I/O3 clock input “L” pulse width(1)	4.5 ≤ V_{CC} ≤ 5.5 V	120			ns
		4.0 ≤ V_{CC} < 4.5 V	150			
		2.7 ≤ V_{CC} < 4.0 V	240			
$t_{su(RxD1-SCLK1)}$ $t_{su(RxD3-SCLK3)}$	Serial I/O1, serial I/O3 clock input setup time	4.5 ≤ V_{CC} ≤ 5.5 V	70			ns
		4.0 ≤ V_{CC} < 4.5 V	90			
		2.7 ≤ V_{CC} < 4.0 V	100			
$t_{h(SCLK1-RxD1)}$ $t_{h(SCLK3-RxD3)}$	Serial I/O1, serial I/O3 clock input hold time	4.5 ≤ V_{CC} ≤ 5.5 V	32			ns
		4.0 ≤ V_{CC} < 4.5 V	40			
		2.7 ≤ V_{CC} < 4.0 V	50			
$t_{C(SCLK2)}$	Serial I/O2 clock input cycle time	4.5 ≤ V_{CC} ≤ 5.5 V	500			ns
		4.0 ≤ V_{CC} < 4.5 V	650			
		2.7 ≤ V_{CC} < 4.0 V	1000			
$t_{WH(SCLK2)}$	Serial I/O2 clock input “H” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V	200			ns
		4.0 ≤ V_{CC} < 4.5 V	260			
		2.7 ≤ V_{CC} < 4.0 V	400			
$t_{WL(SCLK2)}$	Serial I/O2 clock input “L” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V	200			ns
		4.0 ≤ V_{CC} < 4.5 V	260			
		2.7 ≤ V_{CC} < 4.0 V	400			
$t_{su(SIN2-SCLK2)}$	Serial I/O2 clock input setup time	4.5 ≤ V_{CC} ≤ 5.5 V	100			ns
		4.0 ≤ V_{CC} < 4.5 V	130			
		2.7 ≤ V_{CC} < 4.0 V	200			
$t_{h(SCLK2-SIN2)}$	Serial I/O2 clock input hold time	4.5 ≤ V_{CC} ≤ 5.5 V	100			ns
		4.0 ≤ V_{CC} < 4.5 V	130			
		2.7 ≤ V_{CC} < 4.0 V	150			

NOTE:
1. When bit 6 of address $001A_{16}$ and bit 6 of address 0032_{16} are “1” (clock synchronous).
Divide this value by four when bit 6 of address $001A_{16}$ and bit 6 of address 0032_{16} are “0” (UART).

Table 33 Switching characteristics
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, Ta = −20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
$t_{WH}(S_{CLK1})$ $t_{WH}(S_{CLK3})$	Serial I/O1, serial I/O3 clock output “H” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V		$t_C(S_{CLK1})/2-30$, $t_C(S_{CLK3})/2-30$			ns
		4.0 ≤ V_{CC} < 4.5 V		$t_C(S_{CLK1})/2-35$, $t_C(S_{CLK3})/2-35$			
		2.7 ≤ V_{CC} < 4.0 V		$t_C(S_{CLK1})/2-40$, $t_C(S_{CLK3})/2-40$			
$t_{WL}(S_{CLK1})$ $t_{WL}(S_{CLK3})$	Serial I/O1, serial I/O3 clock output “L” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V		$t_C(S_{CLK1})/2-30$, $t_C(S_{CLK3})/2-30$			ns
		4.0 ≤ V_{CC} < 4.5 V		$t_C(S_{CLK1})/2-35$, $t_C(S_{CLK3})/2-35$			
		2.7 ≤ V_{CC} < 4.0 V		$t_C(S_{CLK1})/2-40$, $t_C(S_{CLK3})/2-40$			
$t_d(S_{CLK1}\text{-}T_XD_1)$ $t_d(S_{CLK3}\text{-}T_XD_3)$	Serial I/O1, serial I/O3 output delay time(1)	4.5 ≤ V_{CC} ≤ 5.5 V				140	ns
		4.0 ≤ V_{CC} < 4.5 V				200	
		2.7 ≤ V_{CC} < 4.0 V				350	
$t_V(S_{CLK1}\text{-}T_XD_1)$ $t_V(S_{CLK3}\text{-}T_XD_3)$	Serial I/O1, serial I/O3 output valid time(1)	4.5 ≤ V_{CC} ≤ 5.5 V		−30			ns
		4.0 ≤ V_{CC} < 4.5 V		−30			
		2.7 ≤ V_{CC} < 4.0 V		−30			
$t_r(S_{CLK1})$ $t_r(S_{CLK3})$	Serial I/O1, serial I/O3 rise time of clock output	4.5 ≤ V_{CC} ≤ 5.5 V				30	ns
		4.0 ≤ V_{CC} < 4.5 V				35	
		2.7 ≤ V_{CC} < 4.0 V				40	
$t_f(S_{CLK1})$ $t_f(S_{CLK3})$	Serial I/O1, serial I/O3 fall time of clock output	4.5 ≤ V_{CC} ≤ 5.5 V				30	ns
		4.0 ≤ V_{CC} < 4.5 V				35	
		2.7 ≤ V_{CC} < 4.0 V				40	
$t_{WH}(S_{CLK2})$	Serial I/O2 clock output “H” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V		$t_C(S_{CLK2})/2-160$			ns
		4.0 ≤ V_{CC} < 4.5 V	Fig.114	$t_C(S_{CLK2})/2-200$			
		2.7 ≤ V_{CC} < 4.0 V		$t_C(S_{CLK2})/2-240$			
$t_{WL}(S_{CLK2})$	Serial I/O2 clock output “L” pulse width	4.5 ≤ V_{CC} ≤ 5.5 V		$t_C(S_{CLK2})/2-160$			ns
		4.0 ≤ V_{CC} < 4.5 V		$t_C(S_{CLK2})/2-200$			
		2.7 ≤ V_{CC} < 4.0 V		$t_C(S_{CLK2})/2-240$			
$t_d(S_{CLK2}\text{-}S_{OUT2})$	Serial I/O2 output delay time	4.5 ≤ V_{CC} ≤ 5.5 V				200	ns
		4.0 ≤ V_{CC} < 4.5 V				250	
		2.7 ≤ V_{CC} < 4.0 V				300	
$t_V(S_{CLK2}\text{-}S_{OUT2})$	Serial I/O2 output valid time	4.5 ≤ V_{CC} ≤ 5.5 V			0		ns
		4.0 ≤ V_{CC} < 4.5 V			0		
		2.7 ≤ V_{CC} < 4.0 V			0		
$t_f(S_{CLK2})$	Serial I/O2 fall time of clock output	4.5 ≤ V_{CC} ≤ 5.5 V				30	ns
		4.0 ≤ V_{CC} < 4.5 V				35	
		2.7 ≤ V_{CC} < 4.0 V				40	
t_r(CMOS)	CMOS rise time of output(2)	4.5 ≤ V_{CC} ≤ 5.5 V			10	30	ns
		4.0 ≤ V_{CC} < 4.5 V			12	35	
		2.7 ≤ V_{CC} < 4.0 V			15	40	
t_f(CMOS)	CMOS fall time of output(2)	4.5 ≤ V_{CC} ≤ 5.5 V			10	30	ns
		4.0 ≤ V_{CC} < 4.5 V			12	35	
		2.7 ≤ V_{CC} < 4.0 V			15	40	

NOTES:
1. When the $P4_5/T_XD_1$ P-channel output disable bit of the UART1 control register (bit 4 of address $001B_{16}$) is “0”.
2. When the $P3_5/T_XD_3$ P4-channel output disable bit of the UART3 control register (bit 4 of address 0033_{16}) is “0”.

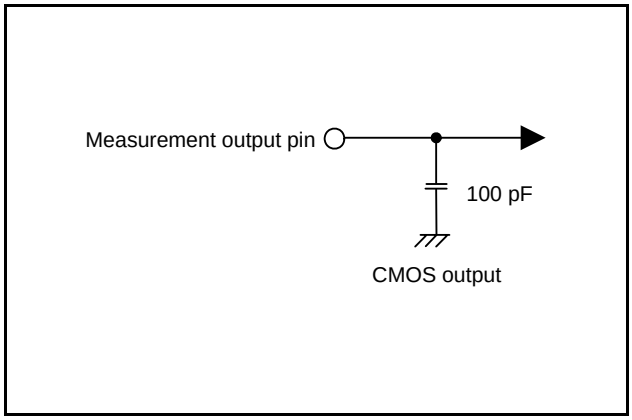


Fig. 114 Circuit for measuring output switching characteristics (1)

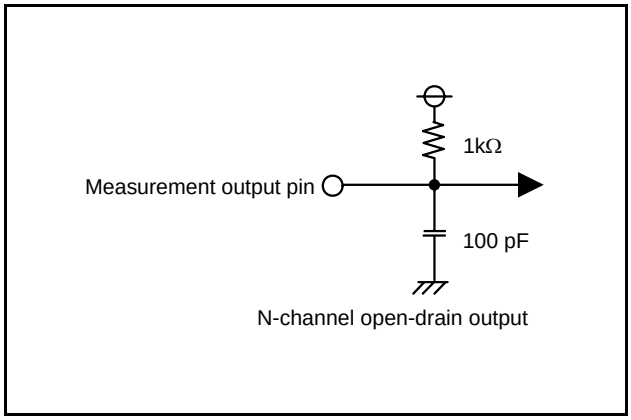


Fig. 115 Circuit for measuring output switching characteristics (2)

Single-chip mode timing diagram

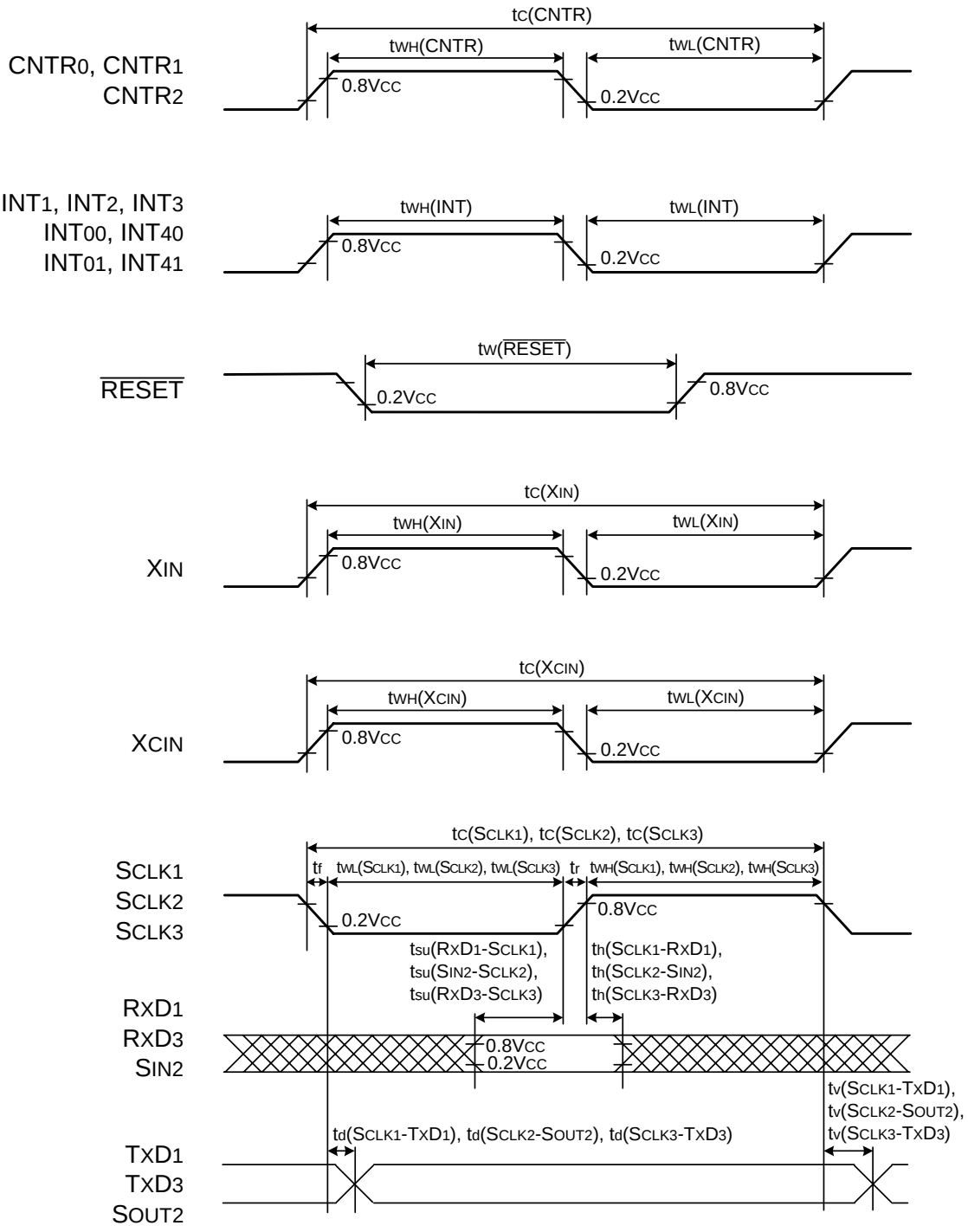


Fig. 116 Timing diagram (in single-chip mode)

Table 34 Multi-master I²C-BUS bus line characteristics

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
t_{BUF}	Bus free time	4.7		1.3		µs
$t_{HD;STA}$	Hold time for START condition	4.0		0.6		µs
t_{LOW}	Hold time for SCL clock = ”0”	4.7		1.3		µs
t_R	Rising time of both SCL and SDA signals		1000	20+0.1Cb(1)	300	ns
$t_{HD;DAT}$	Data hold time	0		0	0.9	µs
t_{HIGH}	Hold time for SCL clock = “1”	4.0		0.6		µs
t_F	Falling time of both SCL and SDA signals		300	20+0.1Cb(1)	300	ns
$t_{SU;DAT}$	Data setup time	250		100		ns
$t_{SU;STA}$	Setup time for repeated START condition	4.7		0.6		µs
$t_{SU;STO}$	Setup time for STOP condition	4.0		0.6		µs

NOTE:
1. Cb = total capacitance of 1 bus line

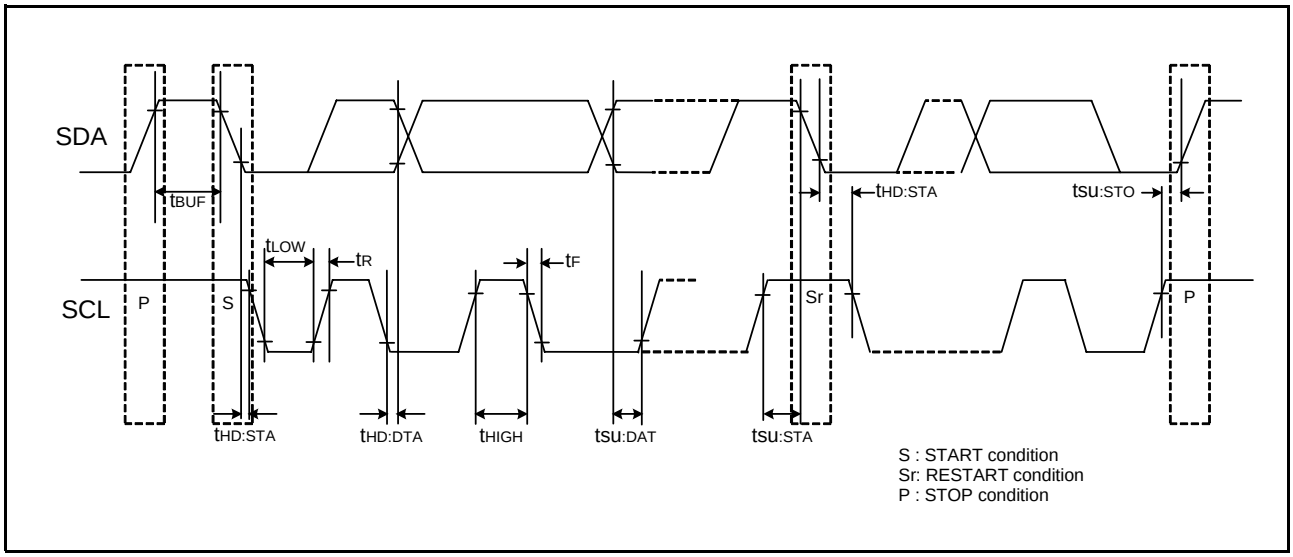
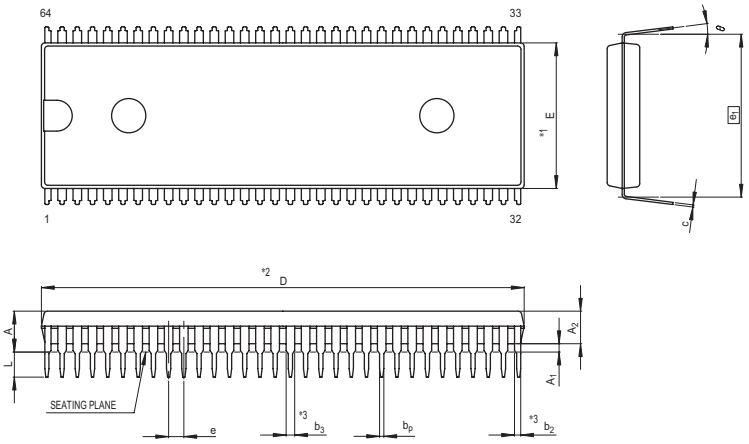


Fig. 117 Timing diagram of multi-master I²C-BUS

PACKAGE OUTLINE

Diagrams showing the latest package dimensions and mounting information are available in the “Packages” section of the Renesas Technology website.

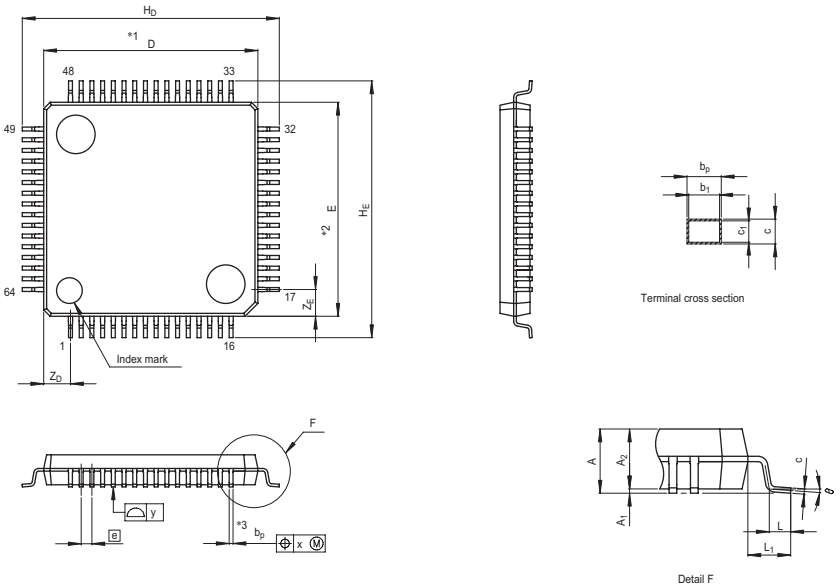
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SDIP64-17x56.4-1.78	PRDP0064BA-A	64P4B	7.9g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
e1	18.75	19.05	19.35
D	56.2	56.4	56.6
E	16.85	17.0	17.15
A	—	—	5.08
A_1	0.38	—	—
A_2	—	3.8	—
b_p	0.4	0.5	0.6
b_2	0.65	0.75	1.05
b_3	0.9	1.0	1.3
c	0.2	0.25	0.32
θ	0°	—	15°
e	1.528	1.778	2.028
L	2.8	—	—

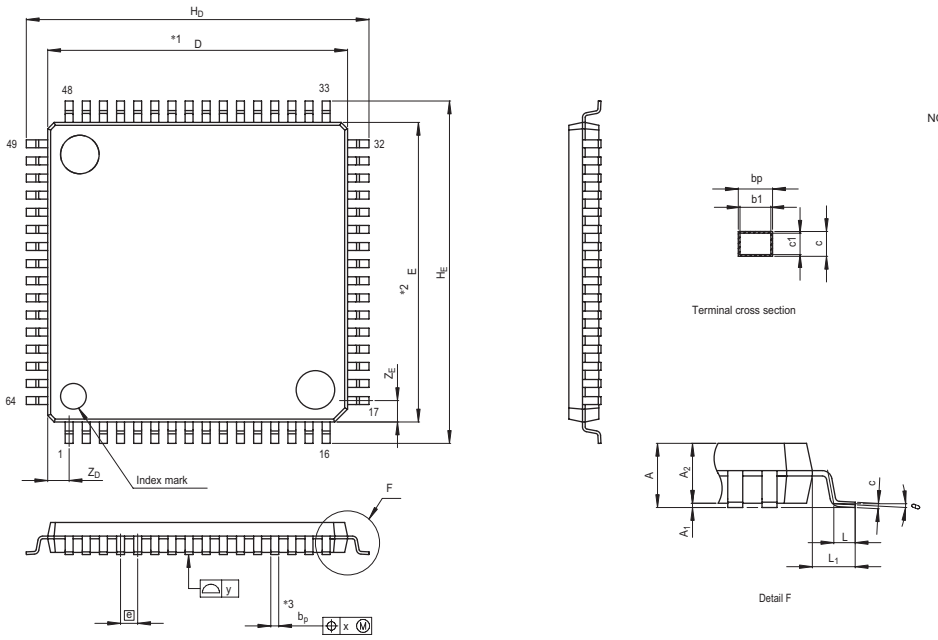
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP64-10x10-0.50	PLQP0064KB-A	64P6Q-A / FP-64K / FP-64KV	0.3g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

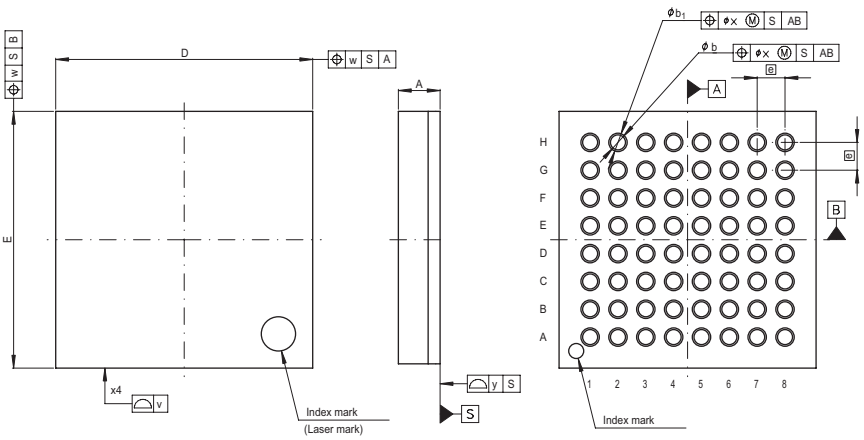
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP64-14x14-0.80	PLQP0064GA-A	64P6U-A	0.7g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.32	0.37	0.42
b_1	—	0.35	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.8	—
x	—	—	0.20
y	—	—	0.10
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.3	0.5	0.7
L_1	—	1.0	—

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA64-6x6-0.65	PTLG0064JA-A	64F0G	0.07g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	6.0	—
E	—	6.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.65	—
b	0.31	0.35	0.39
b_1	0.39	0.43	0.47
x	—	—	0.08
y	—	—	0.10

REVISION HISTORY	3804 Group (Spec.L) Data Sheet

Rev.	Date	Description	
		Page	Summary
1.00	Oct. 27, 2008	-	First edition issued