MITSUBISHI 8-BIT SINGLE-CHIP MICROCOMPUTER
740 FAMILY / 38000 SERIES

3874 Group

User's Manual

MITSUBISHI
ELECTRIC

keep safety first in your circuit designs !

● Mitsubishi Electric Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

● These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.

● Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts or circuit application examples contained in these materials.

● All information contained in these materials, including product data, diagrams and charts, represent information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.

● Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.

● The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.

● If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of JAPAN and/or the country of destination is prohibited.

● Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

BEFORE USING THIS USER'S MANUAL

This user's manual consists of the following three chapters. Refer to the chapter appropriate to your conditions, such as hardware design or software development.

1. Organization

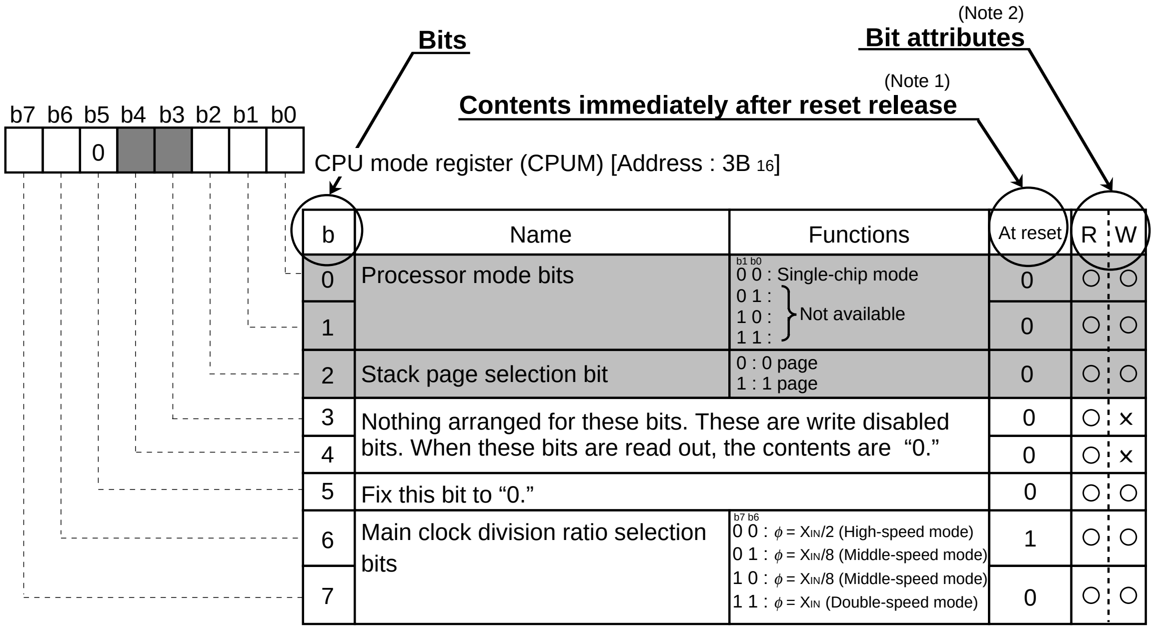
- **CHAPTER 1 HARDWARE**
  This chapter describes features of the microcomputer and operation of each peripheral function.

- **CHAPTER 2 APPLICATION**
  This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of relevant registers.

- **CHAPTER 3 APPENDIX**
  This chapter includes a list of registers, and necessary information for systems development using the microcomputer, the mask ROM confirmation (for mask ROM version), ROM programming confirmation, and the mark specifications which are to be submitted when ordering.

2. Structure of Register

The figure of each register structure describes its functions, contents at reset, and attributes as follows:



CPU mode register (CPUM) [Address : 3B$_{16}$]

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : 1 0 : } Not available 1 1 :	0	○	○
1			0	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Nothing arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0."		0	○	×
4			0	○	×
5	Fix this bit to "0."		0	○	○
6	Main clock division ratio selection bits	b7 b6 0 0 : ϕ = X_{IN}/2 (High-speed mode) 0 1 : ϕ = X_{IN}/8 (Middle-speed mode) 1 0 : ϕ = X_{IN}/8 (Middle-speed mode) 1 1 : ϕ = X_{IN} (Double-speed mode)	1	○	○
7			0	○	○

▇ : Bit in which nothing is arranged   ▒ : Bit that is not used for control of the corresponding function

Notes 1: Contents immediately after reset release
0••••••"0" at reset release
1••••••"1" at reset release
Undefined••••••Undefined or reset release
✻••••••Contents determined by option at reset release

2: Bit attributes••••••The attributes of control register bits are classified into 3 bytes : read-only, write-only and read and write. In the figure, these attributes are represented as follows :

R••••••Read	W••••••Write
○••••••Read enabled	○ ••••••Write enabled
×••••••Read disabled	× ••••••Write disabled

Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 3874 Group.

After reading this manual, the user should have a through knowledge of the functions and features of the 3874 Group, and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.

For details of software, refer to the "740 Family Software Manual."

For details of development support tools, refer to the "DEVELOPMENT SUPPORT TOOLS FOR MICROCOMPUTERS" data book.

Table of contents

CHAPTER 1. HARDWARE

CHAPTER 2. APPLICATION

CHAPTER 3. APPENDIX

List of figures

CHAPTER 1. HARDWARE

List of figures

CHAPTER 2. APPLICATION

CHAPTER 3. APPENDIX

List of tables

CHAPTER 1. HARDWARE

CHAPTER 2. APPLICATION

CHAPTER 3. APPENDIX

CHAPTER 1
HARDWARE

DESCRIPTION
FEATURES
APPLICATION
PIN CONFIGURATION
FUNCTIONAL BLOCK
PIN DESCRIPTION
PART NUMBERING
GROUP EXPANSION
FUNCTIONAL DESCRIPTION
NOTES ON PROGRAMMING
DATA REQUIRED FOR MASK ORDERS
DATA REQUIRED FOR ROM WRITING ORDERS
ROM PROGRAMMING METHOD
FUNCTIONAL DESCRIPTION SUPPLEMENT

DESCRIPTION/FEATURES/APPLICATION

DESCRIPTION

The 3874 group is the 8-bit microcomputer based on the 740 family core technology.

The 3874 group includes data link layer communication control circuit, A-D converters, D-A converter, automatic data transfer serial I/O, UART, and watchdog timer etc.

The various microcomputers in the 3874 group include variations of internal memory size and packaging. For details, refer to the section on part numbering.

For details on availability of microcomputers in the 3874 group, refer to the section on group expansion.

FEATURES

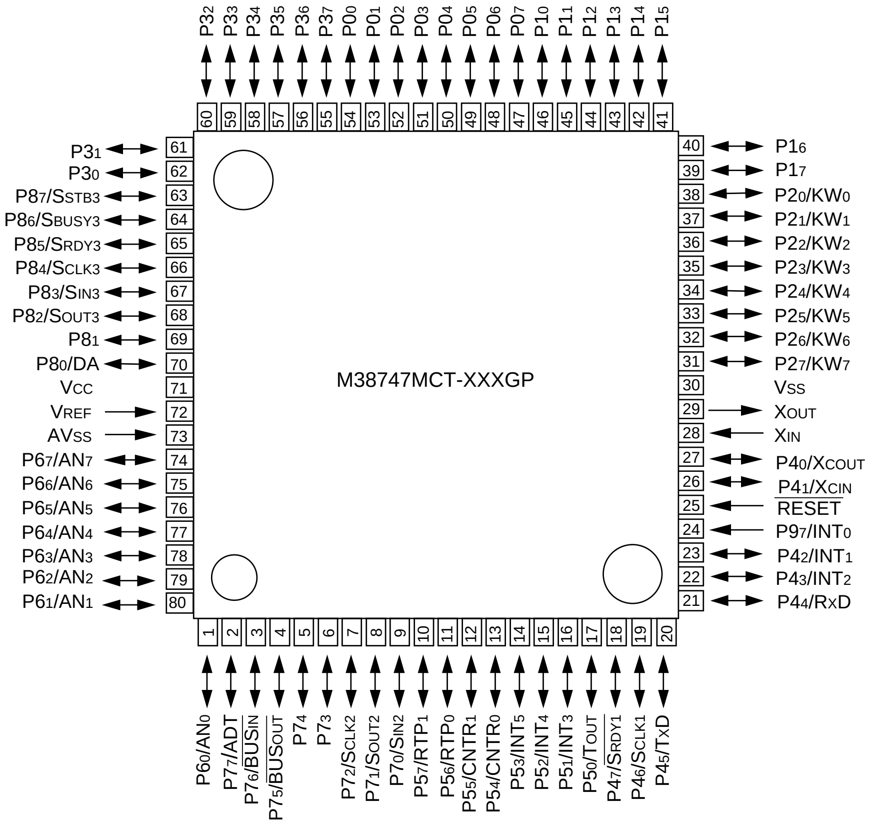
- Basic machine-language instructions 71
- Minimum instruction execution time .... 0.32 µs
  (at 6.4 MHz oscillation frequency, in double-speed mode)
- Memory size
  ROM .... 16 K to 60 K bytes
  RAM .... 1024 to 2048 bytes
- Programmable input/output ports 72
- Input port 1
- Interrupts .... 27 sources, 16 vectors
  (Interrupt source discrimination register exists, included key input interrupt)
- Timer 1, timer 2, timer 3 .... 8-bit ✕ 3
- Timer X, timer Y .... 16-bit ✕ 2
- Serial I/O1 .... 8-bit ✕ 1 (UART or Clock-synchronized)
- Serial I/O2 .... 8-bit ✕ 1(Clock-synchronized)
- Serial I/O3 .... 8-bit ✕ 1
  (Clock-synchronized automatic data transfer/arbitrary bit transfer function available)
- A-D converter .... 8-bit ✕ 8 channels
- D-A converter .... 8-bit ✕ 1 channel
- Data link layer communication control circuit 1
- Clock generating circuit .... Built-in 2 circuits
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Watchdog timer .... 20-bit ✕ 1
- Power source voltage .... 3.0 to 5.5 V
- Power dissipation
  In double-speed mode ....90 mW
  In high-speed mode ....60 mW
  (at 32 kHz oscillation frequency, at 5 V power source voltage)
  In low-speed mode .... 180 µW
  (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Operating temperature range.................................. –40 to 85 °C
  (Extended operating temperature version and automotive version)

APPLICATION

Automotive comfort control for audio system, air conditioning etc., automotive body electronics control, household appliances, and other consumer applications, etc.

PIN CONFIGURATION

PIN CONFIGURATION (TOP VIEW)



Package type : 80P6S-A

Fig. 1 M38747MCT-XXXGP pin configuration

PIN CONFIGURATION

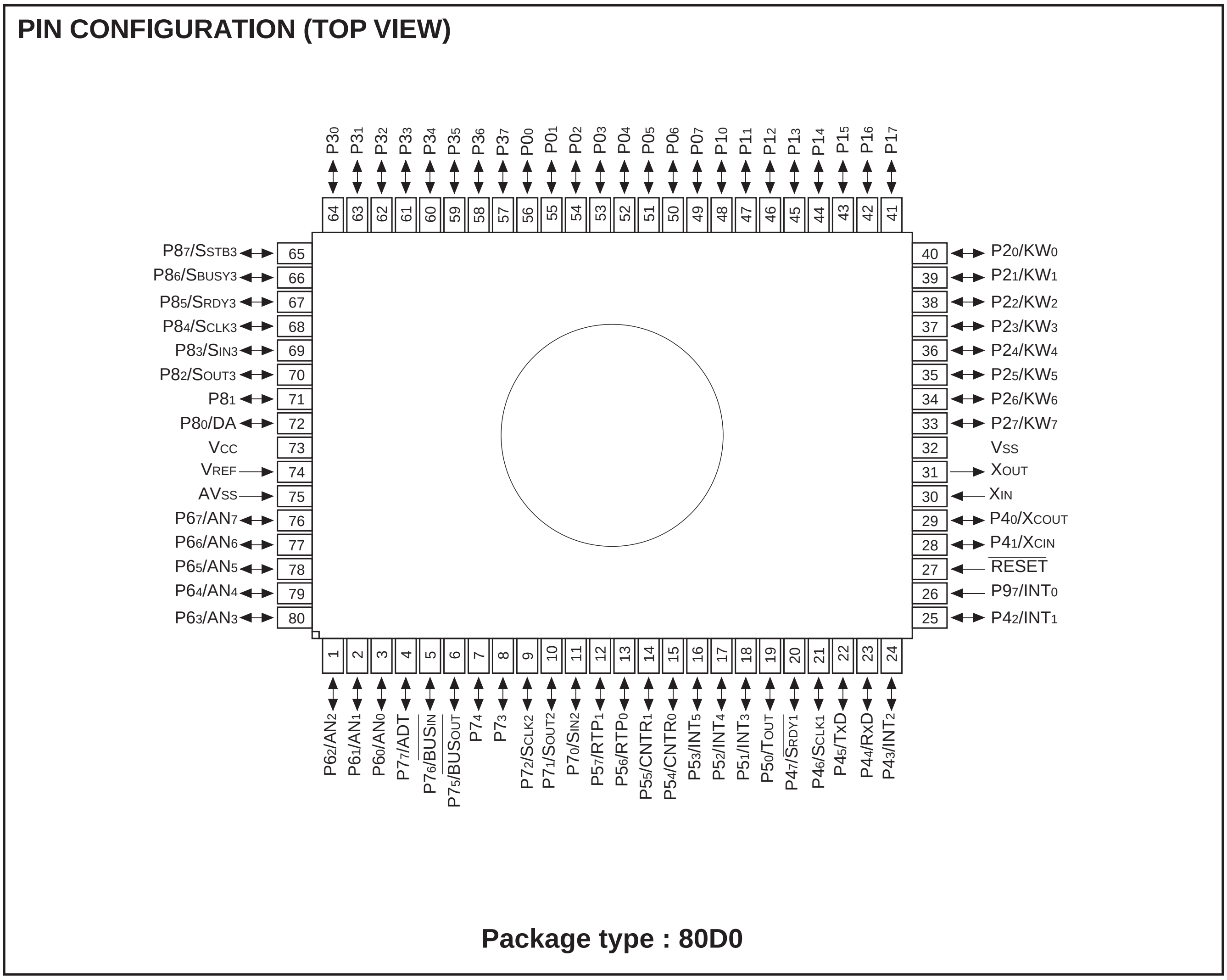


Fig. 2 M38749EFFS pin configuration

FUNCTIONAL BLOCK

FUNCTIONAL BLOCK DIAGRAM (Package : 80P6S-A)

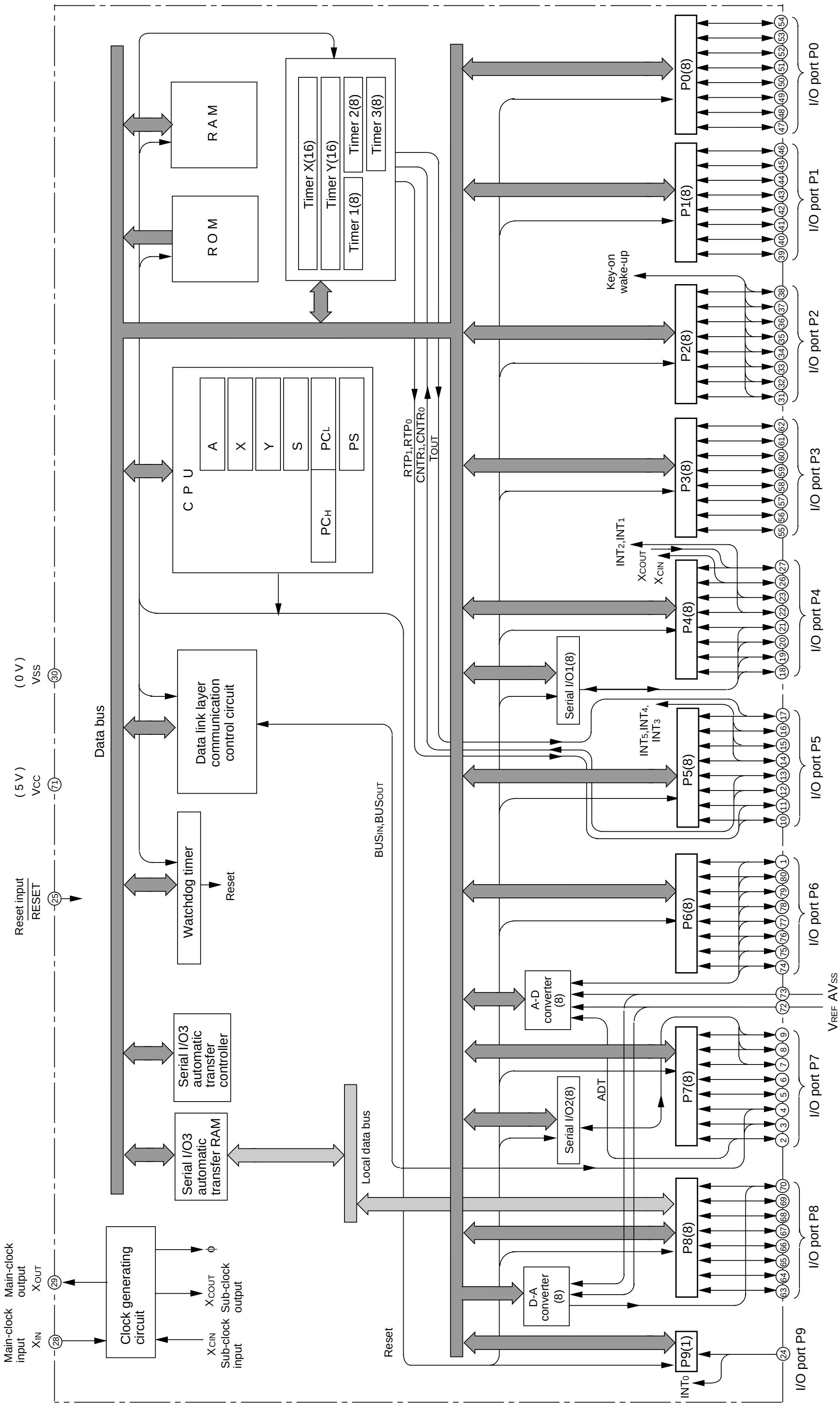


Fig. 3 Functional block diagram

PIN DESCRIPTION

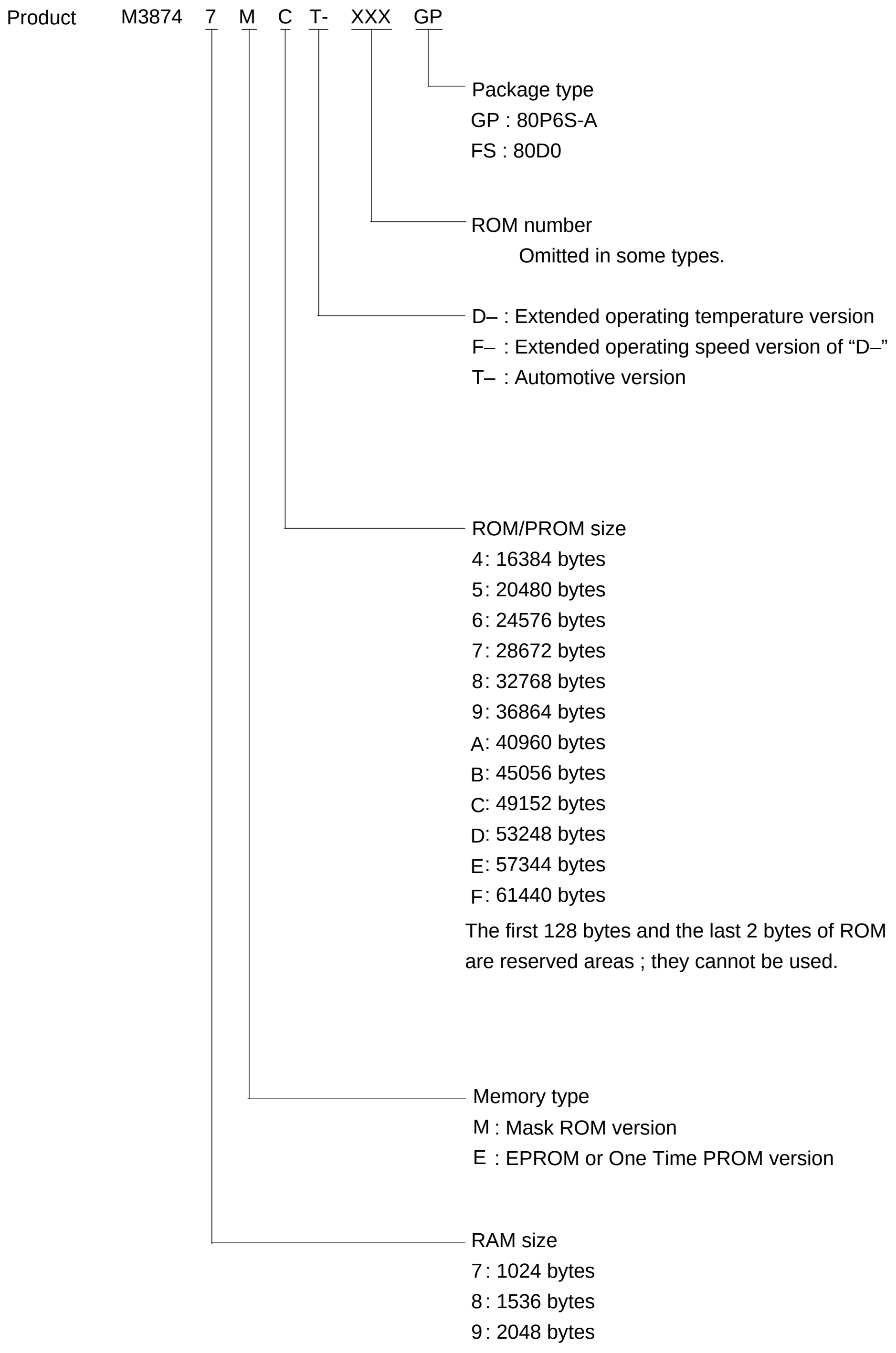
Table 1 Pin description (1)

Pin	Name	Functions	
			Function except a port function
V_{CC}, V_{SS}	Power source input	•Apply voltage of 3.0 V – 5.5 V to Vcc, and 0 V to Vss.	
V_{REF}	Reference voltage input	•Reference voltage input pin for A-D and D-A converters.	
AV_{SS}	Analog power source input	•Analog power source input pin for A-D and D-A converters. •Connect to V_{SS}.	
$\overline{RESET}$	Reset input	•Reset input pin for active "L."	
X_{IN}	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. •Feedback resistor is built in between X_{IN} pin and X_{OUT} pin.	
X_{OUT}	Clock output		
$P0_0$–$P0_7$	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.	
$P1_0$–$P1_7$	I/O port P1		
$P2_0$–$P2_7$	I/O port P2		•Key-on wake-up interrupt input pins
$P3_0$–$P3_7$	I/O port P3		
$P4_0/X_{COUT}$, $P4_1/X_{CIN}$	I/O port P4	•8-bit I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure.	•Sub-clock generating circuit I/O pins connect a resonator. (This circuit cannot be operated by an external clock.)
$P4_2/INT_1$, $P4_3/INT_2$			•Interrupt input pins
$P4_4/R_XD$, $P4_5/T_XD$, $P4_6/S_{CLK1}$, $P4_7/\overline{S_{RDY1}}$			•Serial I/O1 function pins
$P5_0/T_{OUT}$	I/O port P5	•8-bit I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure.	•Timer 2 output pin
$P5_1/INT_3$– $P5_3/INT_5$			•Interrupt input pins
$P5_4/CNTR_0$, $P5_5/CNTR_1$			•Timer X, timer Y function pins •Interrupt input pins
$P5_6/RTP_0$, $P5_7/RTP_1$			•Real time port function pins

Table 2 Pin description (2)

Pin	Name	Functions	
			Function except a port function
P6$_0$/AN$_0$– P6$_7$/AN$_7$	I/O port P6	•8-bit I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure.	•A-D converter input pins
P7$_0$/S$_{IN2}$, P7$_1$/S$_{OUT2}$, P7$_2$/S$_{CLK2}$	I/O port P7	•8-bit I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure.	•Serial I/O2 function pins
P7$_3$, P7$_4$			
P7$_5$/$\overline{BUS_{OUT}}$, P7$_6$/$\overline{BUS}_{IN}$			•Data link layer communication control pins
P7$_7$/ADT			•A-D trigger input pin
P8$_0$/DA	I/O port P8	•8-bit I/O port with the same function as port P0.	•D-A converter output pin
P8$_1$			
P8$_2$/S$_{OUT3}$, P8$_3$/S$_{IN3}$, P8$_4$/S$_{CLK3}$, P8$_5$/S$_{RDY3}$			•Serial I/O3 function pins
P8$_6$/S$_{BUSY3}$, P8$_7$/S$_{STB3}$			
P9$_7$/INT$_0$	Input port P9	•1-bit input port. •CMOS compatible input level.	•Interrupt input pin

PART NUMBERING



Product M3874 7 M C T- XXX GP

Package type
GP : 80P6S-A
FS : 80D0

ROM number
Omitted in some types.

D– : Extended operating temperature version
F– : Extended operating speed version of “D–”
T– : Automotive version

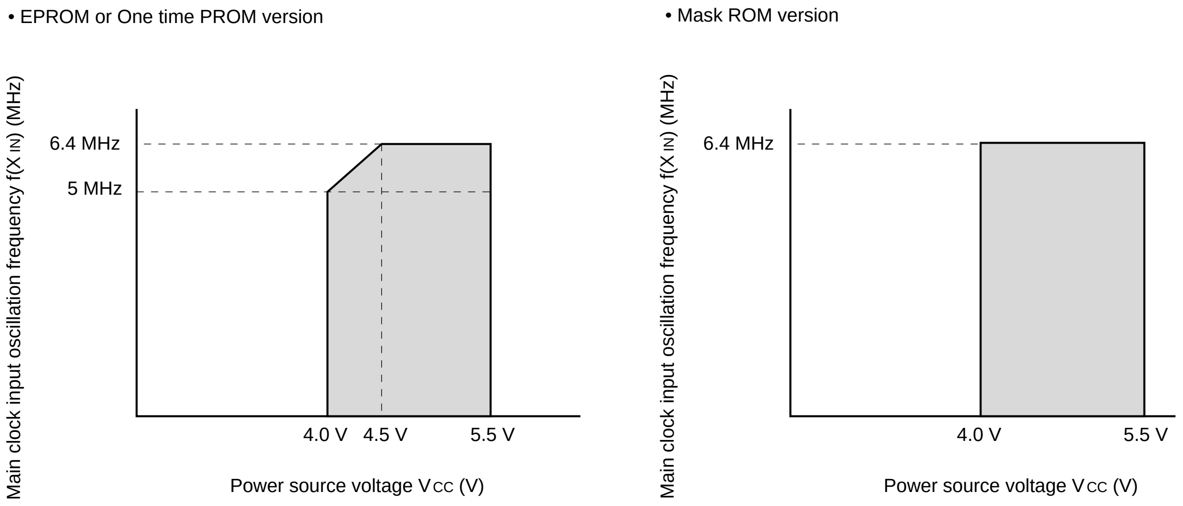
ROM/PROM size
4 : 16384 bytes
5 : 20480 bytes
6 : 24576 bytes
7 : 28672 bytes
8 : 32768 bytes
9 : 36864 bytes
A : 40960 bytes
B : 45056 bytes
C : 49152 bytes
D : 53248 bytes
E : 57344 bytes
F : 61440 bytes
The first 128 bytes and the last 2 bytes of ROM are reserved areas ; they cannot be used.

Memory type
M : Mask ROM version
E : EPROM or One Time PROM version

RAM size
7 : 1024 bytes
8 : 1536 bytes
9 : 2048 bytes

Fig. 4 Part numbering

3874 group main clock input oscillation frequency in double-speed mode



In low-speed mode, middle-speed mode, and high-speed mode, characteristic of main clock input oscillation frequency guarantee limit is not different.

Fig. 5 Main clock input oscillation frequency in double-speed mode

GROUP EXPANSION (Extended operating temperature version)

The 3874 group (extended operating temperature version) is designed for automotive comfort and amusement control such as audio, air-conditioner etc., household appliances, and other consumer applications.

Mitsubishi plans to expand the 3874 group (extended operating temperature version) as follows:

Memory Type

Support for mask ROM, One Time PROM, and EPROM versions

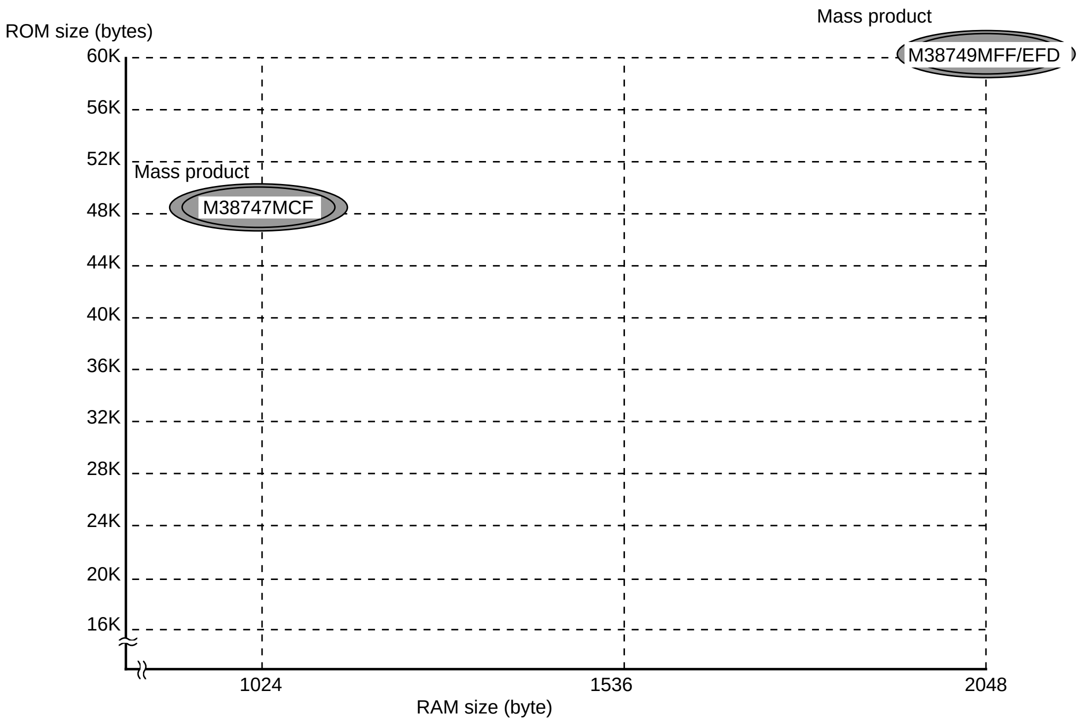
Memory Size

ROM/PROM size .... 48 K to 60 K bytes
RAM size .... 1024 to 2048 bytes

Packages

80P6S-A .... 0.65 mm-pitch plastic molded QFP
80D0 .... 0.8 mm-pitch ceramic LCC (EPROM version)

Memory Expansion Plan of 3874 group (Extended operating temperature version)



Products under development or planning : the development schedule and specification may be revised without notice.
Planning products may be stopped during the development.

Fig. 6 Memory expansion plan (Extended operating temperature version)

Currently planning products are listed below.

Table 3 Support products

As of August 1998

Product name	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38749EFDGP	61440 (61310)	2048	80P6S-A	One Time PROM version (blank)
M38749EFFS			80D0	EPROM version (for software development, operating temperature = –20 to 85 °C)
M38749MFF			80P6S-A	Mask ROM version
M38747MCF	49152 (49022)	1024		

GROUP EXPANSION (Automotive version)

The 3874 group (automotive version) is designed for automotive body electronics control.

Mitsubishi plans to expand the 3874 group (automotive version) as follows:

Memory Type

Support for mask ROM and One Time PROM versions

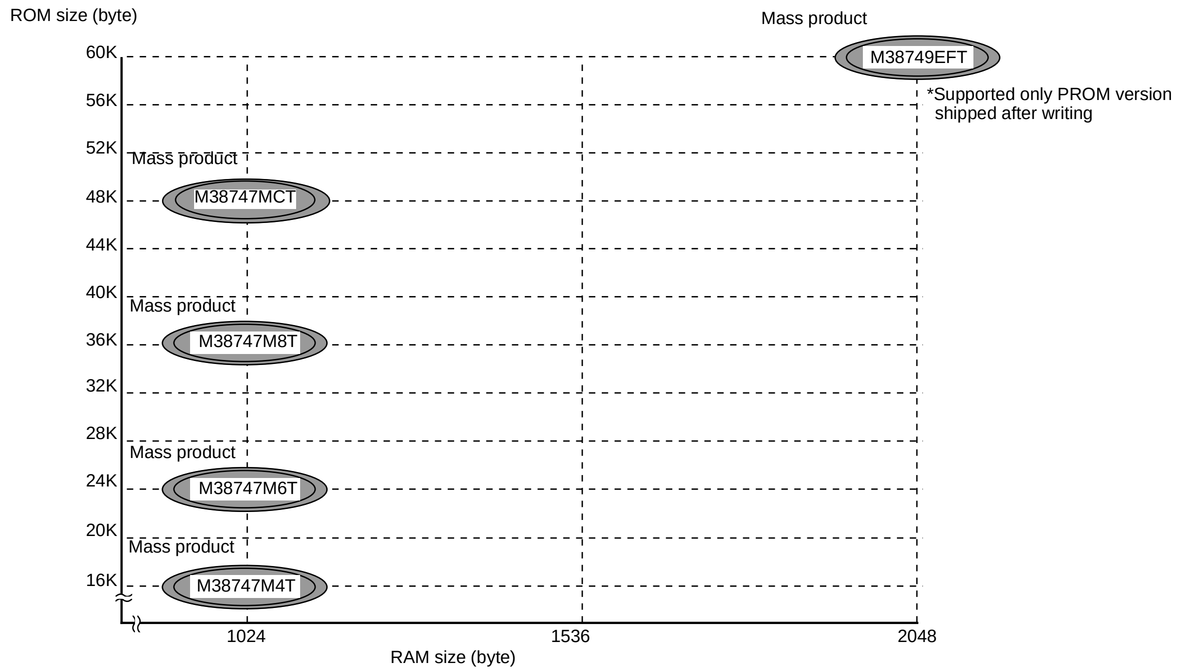
Memory Size

ROM/PROM size .... 16 K to 60 K bytes
RAM size .... 1024 to 2048 bytes

Packages

80P6S-A .... 0.65 mm-pitch plastic molded QFP

Memory Expansion Plan of 3874 group (Automotive version)



Products under development or planning : the development schedule and specification may be revised without notice.
Planning products may be stopped during the development.

Fig. 7 Memory expansion plan (Automotive correspondence version)

Currently planning products are listed below.

Table 4 Support products

As of August 1998

Product name	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38749EFT	61440 (61310)	2048	80P6S-A	One Time PROM version
M38747MCT	49152 (49022)	1024		Mask ROM version
M38747M8T	32768 (32638)			
M38747M6T	24576 (24446)			
M38747M4T	16384 (16254)			

FUNCTIONAL DESCRIPTION
CENTRAL PROCESSING UNIT (CPU)

The 3874 group uses the standard 740 Family instruction set. Refer to the table of 740 Family addressing modes and machine instructions or 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 9.

Store registers other than those described in Figure 9 with program when the user needs them during interrupts or subroutine calls.

[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.

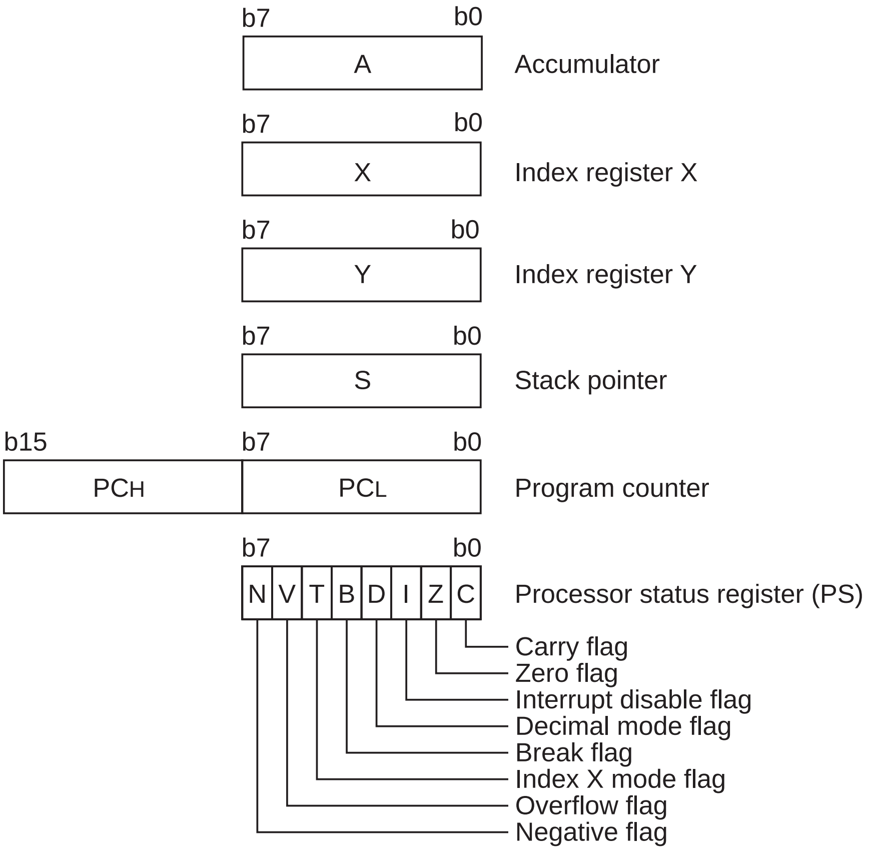
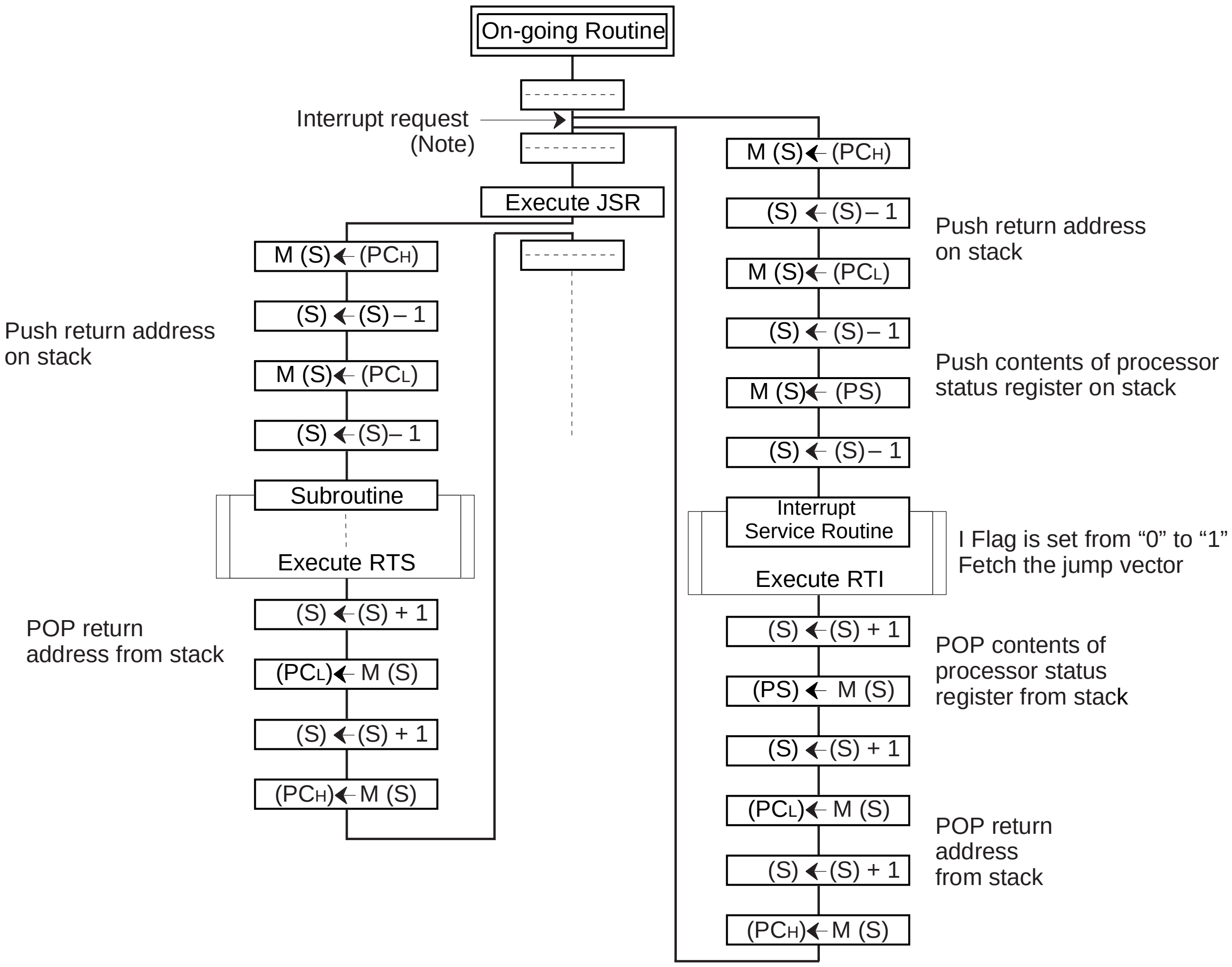


Fig. 8 740 Family CPU register structure



Note : Condition for acceptance of an interrupt → Interrupt enable flag is “1”
Interrupt disable flag is “0”

Fig. 9 Register push and pop at interrupt generation and subroutine call

Table 5 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 be used for 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.

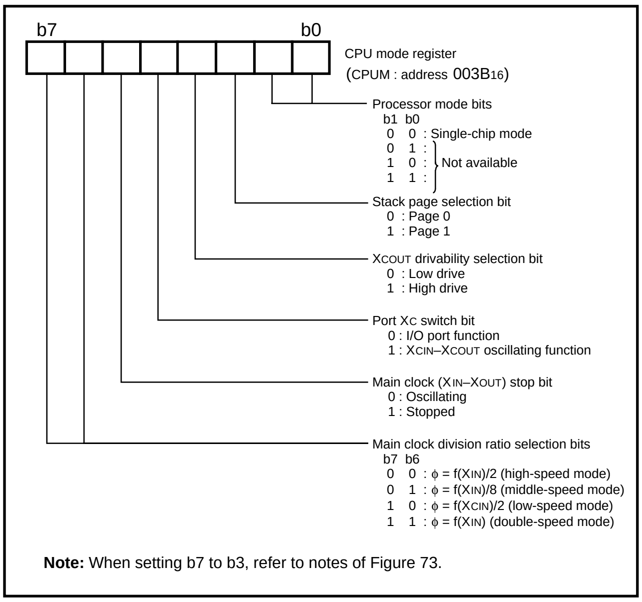
Table 6 Set and clear instructions of each bit of processor status register

	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	–

The central processing unit (CPU) has the six registers. Figure 10 shows the structure of the CPU mode register.

[CPU Mode Register (CPUM)] 003B$_{16}$

The CPU mode register contains the stack page selection bit and the internal system clock selection bit etc.
The CPU mode register is allocated at address 003B$_{16}$.


b7
b0
CPU mode register
(CPUM : address 003B$_{16}$)
Processor mode bits
b1 b0
0 0 : Single-chip mode
0 1 :
1 0 : Not available
1 1 :
Stack page selection bit
0 : Page 0
1 : Page 1
X_{COUT} drivability selection bit
0 : Low drive
1 : High drive
Port X_C switch bit
0 : I/O port function
1 : X_{CIN}–X_{COUT} oscillating function
Main clock (X_{IN}–X_{OUT}) stop bit
0 : Oscillating
1 : Stopped
Main clock division ratio selection bits
b7 b6
0 0 : ϕ = f(X_{IN})/2 (high-speed mode)
0 1 : ϕ = f(X_{IN})/8 (middle-speed mode)
1 0 : ϕ = f(X_{CIN})/2 (low-speed mode)
1 1 : ϕ = f(X_{IN}) (double-speed mode)

Note: When setting b7 to b3, refer to notes of Figure 73.

Fig. 10 Structure of CPU mode register

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

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

ROM

The first 128 bytes and the last 2 bytes of ROM are reserved for device testing and the rest is user area for storing programs.

Interrupt Vector Area

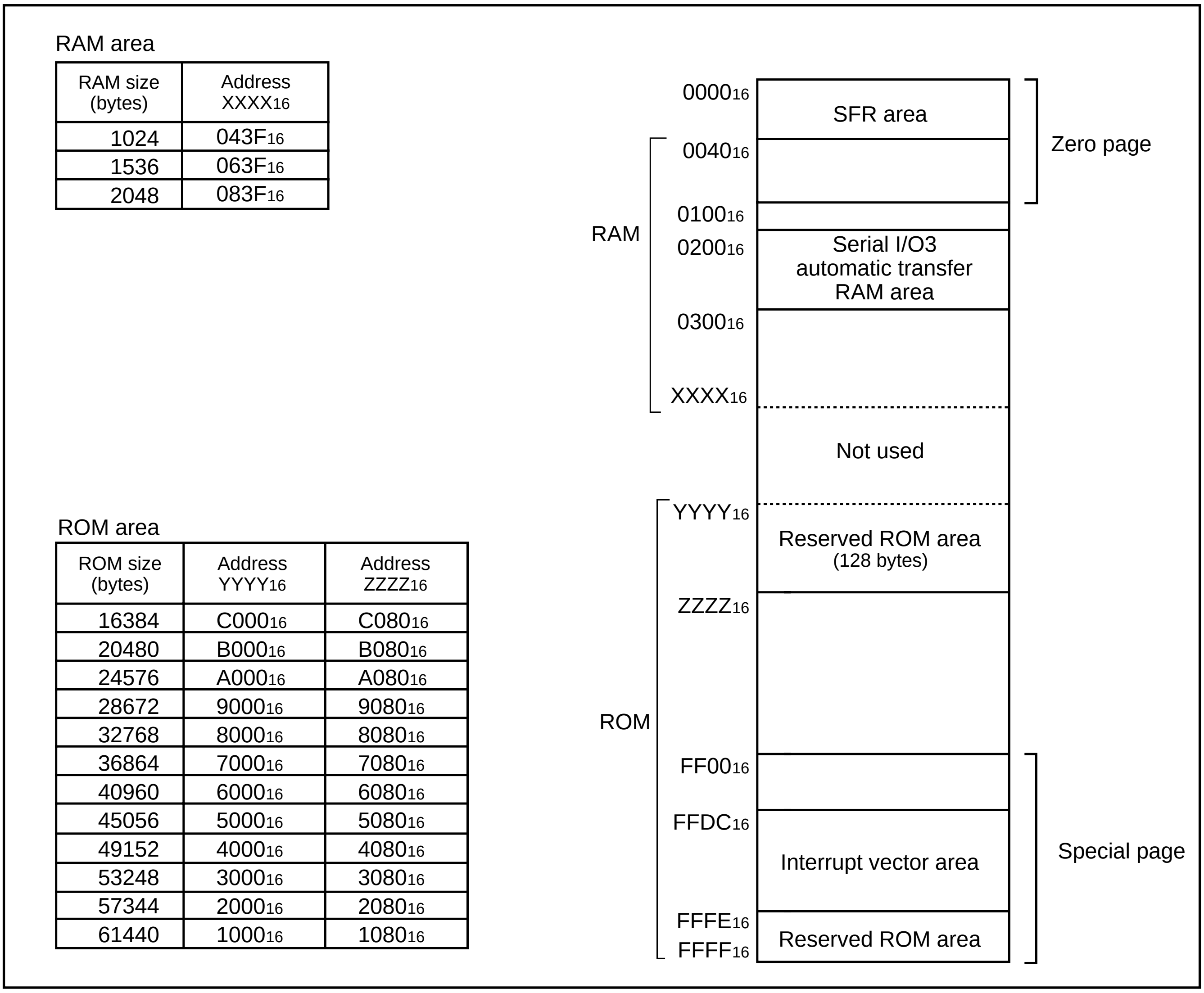
The interrupt vector area contains reset and interrupt vectors.

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.



RAM area

RAM size (bytes)	Address $XXXX_{16}$
1024	$043F_{16}$
1536	$063F_{16}$
2048	$083F_{16}$

ROM area

ROM size (bytes)	Address $YYYY_{16}$	Address $ZZZZ_{16}$
16384	$C000_{16}$	$C080_{16}$
20480	$B000_{16}$	$B080_{16}$
24576	$A000_{16}$	$A080_{16}$
28672	9000_{16}	9080_{16}
32768	8000_{16}	8080_{16}
36864	7000_{16}	7080_{16}
40960	6000_{16}	6080_{16}
45056	5000_{16}	5080_{16}
49152	4000_{16}	4080_{16}
53248	3000_{16}	3080_{16}
57344	2000_{16}	2080_{16}
61440	1000_{16}	1080_{16}

Fig. 11 Memory map diagram

0000_{16}	Port P0 (P0)	0020_{16}	Timer X (low-order) (TXL)
0001_{16}	Port P0 direction register (P0D)	0021_{16}	Timer X (high-order) (TXH)
0002_{16}	Port P1 (P1)	0022_{16}	Timer Y (low-order) (TYL)
0003_{16}	Port P1 direction register (P1D)	0023_{16}	Timer Y (high-order) (TYH)
0004_{16}	Port P2 (P2)	0024_{16}	Timer 1 (T1)
0005_{16}	Port P2 direction register (P2D)	0025_{16}	Timer 2 (T2)
0006_{16}	Port P3 (P3)	0026_{16}	Timer 3 (T3)
0007_{16}	Port P3 direction register (P3D)	0027_{16}	Timer X mode register (TXM)
0008_{16}	Port P4 (P4)	0028_{16}	Timer Y mode register (TYM)
0009_{16}	Port P4 direction register (P4D)	0029_{16}	Timer 123 mode register (T123M)
$000A_{16}$	Port P5 (P5)	$002A_{16}$	Communication mode register (BUSM)
$000B_{16}$	Port P5 direction register (P5D)	$002B_{16}$	Transmit control register (TXDCON)
$000C_{16}$	Port P6 (P6)	$002C_{16}$	Transmit status register (TXDSTS)
$000D_{16}$	Port P6 direction register (P6D)	$002D_{16}$	Receive control register (RXDCON)
$000E_{16}$	Port P7 (P7)	$002E_{16}$	Receive status register (RXDSTS)
$000F_{16}$	Port P7 direction register (P7D)	$002F_{16}$	Bus interrupt source discrimination control register (BICOND)
0010_{16}	Port P8 (P8)	0030_{16}	Control field selection register (CFSEL)
0011_{16}	Port P8 direction register (P8D)	0031_{16}	Control field register (CF)
0012_{16}	Port P9 (P9)	0032_{16}	Transmit/Receive FIFO (TRFIFO)
0013_{16}	Serial I/O3 register/Transfer counter (SIO3)	0033_{16}	PULL UP register (PULLU)
0014_{16}	Serial I/O3 control register 1 (SIO3CON1)	0034_{16}	A-D control register (ADCON)
0015_{16}	Serial I/O3 control register 2 (SIO3CON2)	0035_{16}	A-D/D-A conversion register (AD)
0016_{16}	Serial I/O3 control register 3 (SIO3CON3)	0036_{16}	Interrupt source discrimination register 2 (IREQD2)
0017_{16}	Serial I/O3 automatic transfer data pointer (SIO3DP)	0037_{16}	Interrupt source discrimination control register 2 (ICOND2)
0018_{16}	Transmit/Receive buffer register (TB/RB)	0038_{16}	Interrupt source discrimination register 1 (IREQD1)
0019_{16}	Serial I/O1 status register (SIO1STS)	0039_{16}	Interrupt source discrimination control register 1 (ICOND1)
$001A_{16}$	Serial I/O1 control register (SIO1CON)	$003A_{16}$	Interrupt edge selection register (INTEDGE)
$001B_{16}$	UART control register (UARTCON)	$003B_{16}$	CPU mode register (CPUM)
$001C_{16}$	Baud rate generator (BRG)	$003C_{16}$	Interrupt request register 1 (IREQ1)
$001D_{16}$	Serial I/O2 control register (SIO2CON)	$003D_{16}$	Interrupt request register 2 (IREQ2)
$001E_{16}$	Watchdog timer control register (WDTCON)	$003E_{16}$	Interrupt control register 1 (ICON1)
$001F_{16}$	Serial I/O2 register (SIO2)	$003F_{16}$	Interrupt control register 2 (ICON2)

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

I/O PORTS

The I/O ports P0–P8 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.

Table 7 I/O port function (1)

Pin	Name	Input/Output	I/O Structure	Non-Port Function	Related SFRs	Ref.No.
P0$_0$–P0$_7$	Port P0	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output			(1)
P1$_0$–P1$_7$	Port P1	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output			
P2$_0$–P2$_7$	Port P2	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output	•Key input (key-on wake-up) interrupt input	•PULL UP register	(2)
P3$_0$–P3$_7$	Port P3	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output		•CPU mode register	(1)
P4$_0$/X$_{COUT}$	Port P4	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output	•Sub-clock generating circuit I/O	•CPU mode register	(3)
P4$_1$/X$_{CIN}$						(4)
P4$_2$/INT$_1$, P4$_3$/INT$_2$				•External interrupt input	•Interrupt edge selection register	(5)
P4$_4$/R$_X$D				•Serial I/O1 function I/O	•Serial I/O1 control register •Serial I/O1 status register •UART control register •PULL UP register	(6)
P4$_5$/T$_X$D						(7)
P4$_6$/S$_{CLK1}$						(8)
P4$_7$/$\overline{S_{RDY1}}$						(9)
P5$_0$/T$_{OUT}$	Port P5	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output	•Timer 2 output	•Timer 123 mode register	(10)
P5$_1$/INT$_3$, P5$_2$/INT$_4$, P5$_3$/INT$_5$				•External interrupt input	•Interrupt edge selection register	(5)
P5$_4$/CNTR$_0$				•Timer X function I/O	•Timer X mode register	(11)
P5$_5$/CNTR$_1$				•Timer Y function I/O	•Timer Y mode register	(12)
P5$_6$/RTP$_0$				•Real time port function output	•Timer X mode register	(13)
P5$_7$/RTP$_1$				•Real time port function output	•Timer Y mode register	
P6$_0$/AN$_0$– P6$_7$/AN$_7$	Port P6	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output	•A-D converter input	•A-D control register	(14)

Table 8 I/O port function (2)

Pin	Name	Input/Output	I/O Function	Non-Port Function	Related SFRs	Ref.No.
P7$_0$/S$_{IN2}$	Port P7	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output	•Serial I/O2 function I/O	•Serial I/O2 control register •PULL UP register	(15)
P7$_1$/S$_{OUT2}$						(16)
P7$_2$/S$_{CLK2}$						(17)
P7$_3$,P7$_4$						(1)
P7$_5$/$\overline{\mathrm{BUS}_{\mathrm{OUT}}}$				•Data link layer communication control I/O	•Communication mode register •Transmit control register •Transmit status register •Receive control register •Receive status register •Bus interrupt source discrimination control register •Control field selection register •Control field register	(18)
P7$_6$/$\overline{\mathrm{BUS}_{\mathrm{IN}}}$						(19)
P7$_7$/ADT				•A-D trigger input	•Transmit/Receive FIFO	(20)
P8$_0$/DA	Port P8	Input/output, individual bits	•CMOS compatible input level •CMOS 3-state output	•D-A function output	•A-D control register	(21)
P8$_1$						(1)
P8$_2$/S$_{OUT3}$				•Serial I/O3 function I/O	•A-D control register •Serial I/O3 register/ Transfer counter •Serial I/O3 control register 1 •Serial I/O3 control register 2 •Serial I/O3 control register 3 •Serial I/O3 automatic	(22)
P8$_3$/S$_{IN3}$						(23)
P8$_4$/S$_{CLK3}$						(24)
P8$_5$/S$_{RDY3}$						(25)
P8$_6$/S$_{BUSY3}$						(26)
P8$_7$/S$_{STB3}$						(27)
P9$_7$/INT$_0$	Port P9	Input	•CMOS compatible input level	•External interrupt input	transfer data pointer •Interrupt edge selection register	(28)

Note: Make sure that the input level at each pin is either 0 V or Vcc during execution of the STP instruction.
When an input level is at an intermediate potential, a current will flow from Vcc to Vss through the input-stage gate.

Pull-up Control

P2₀–P2₇, TxD, SCLK1, SOUT2, and SCLK2 can perform pull-up control by setting “1” to the PULL UP register (address 0033_{16}).
P2₀–P2₇’s pull-up is valid in the input mode, and TxD, SCLK1, SOUT2, and SCLK2S’ pull-up is valid in the output mode.

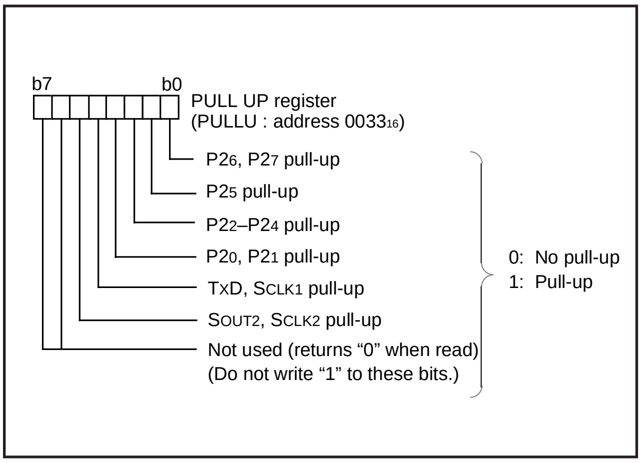
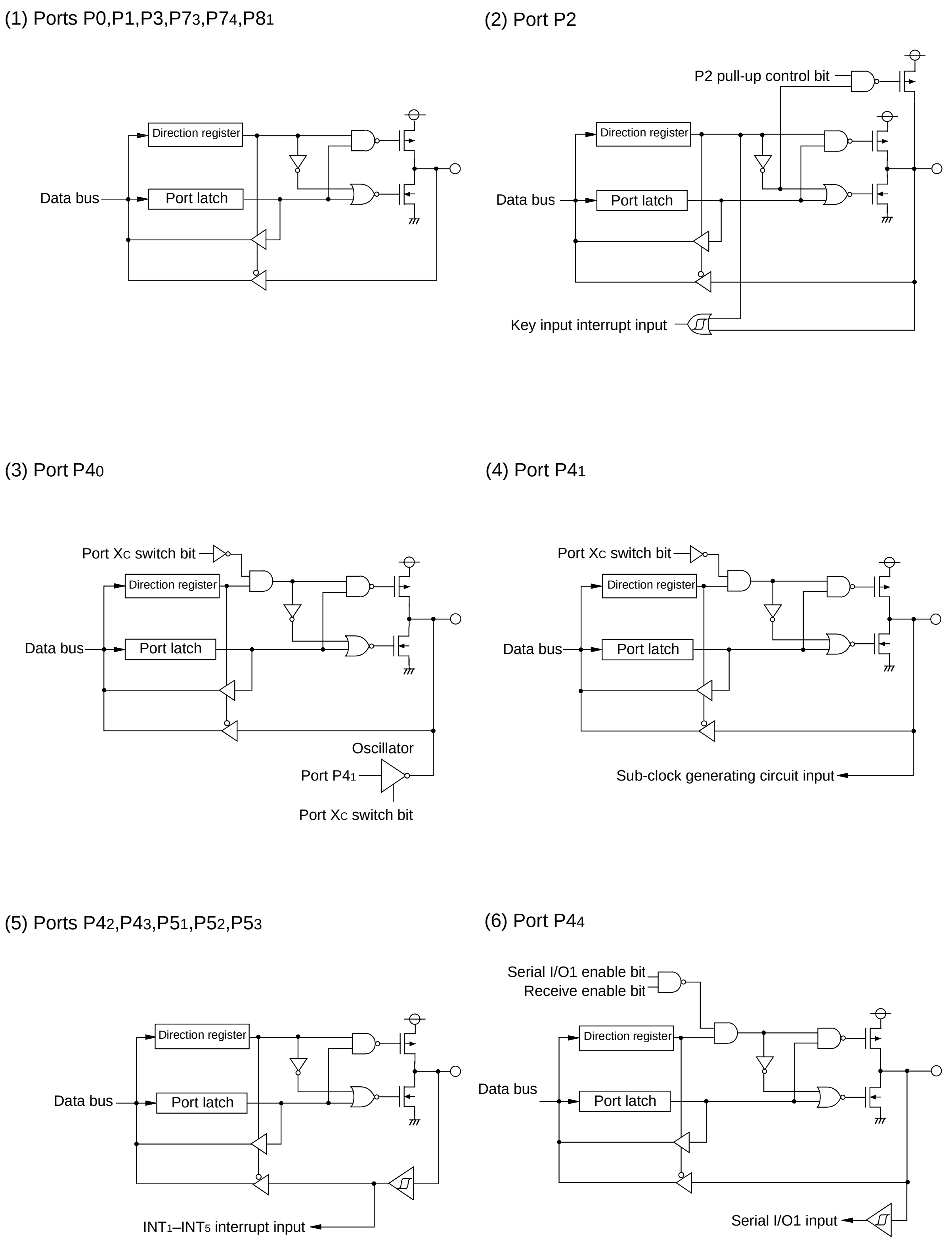


Fig. 13 Structure of PULL UP register



(1) Ports P0,P1,P3,P7$_3$,P7$_4$,P8$_1$

(2) Port P2

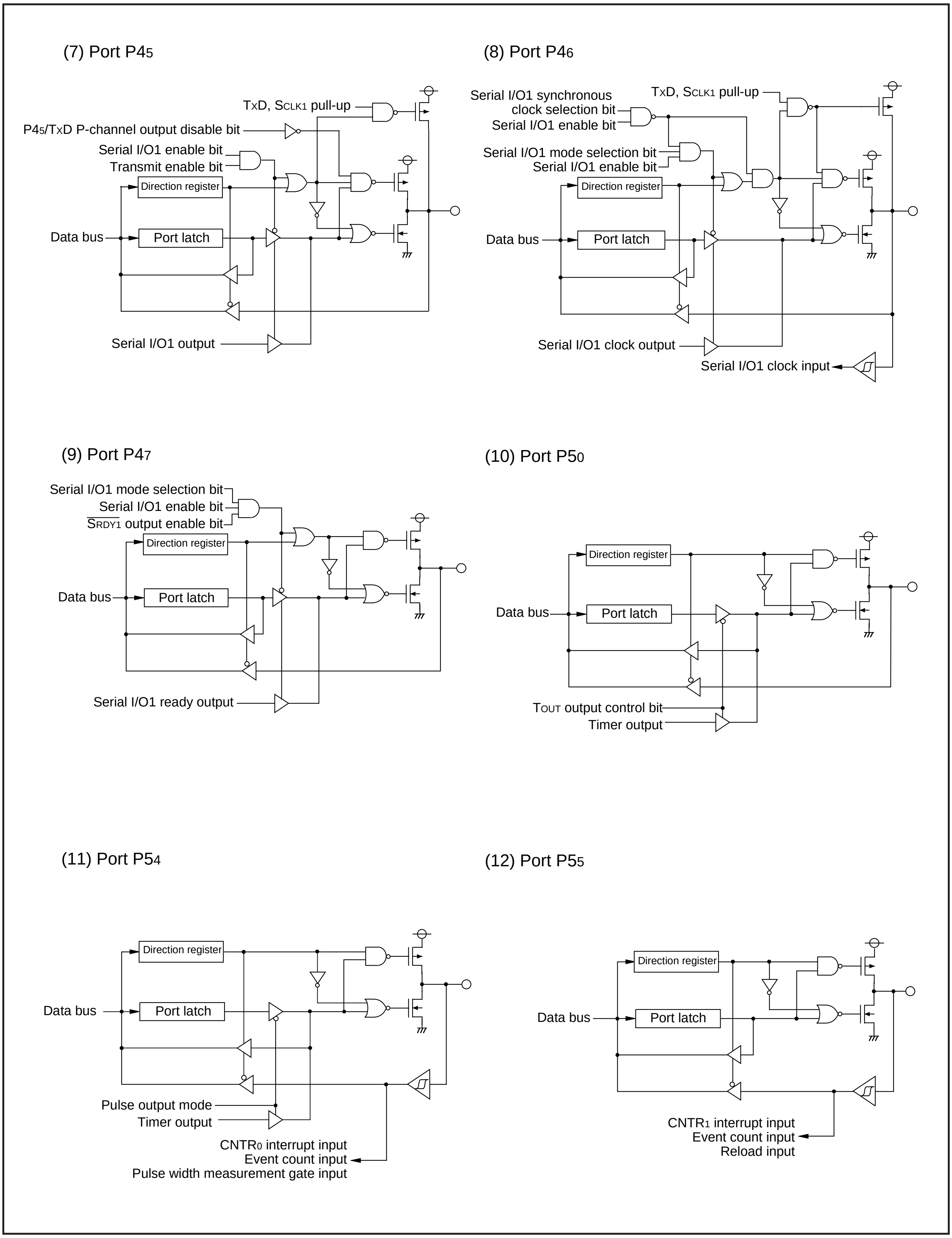
(3) Port P4$_0$

(4) Port P4$_1$

(5) Ports P4$_2$,P4$_3$,P5$_1$,P5$_2$,P5$_3$

(6) Port P4$_4$

Fig. 14 Port block diagram (1)


(7) Port P45

(8) Port P46

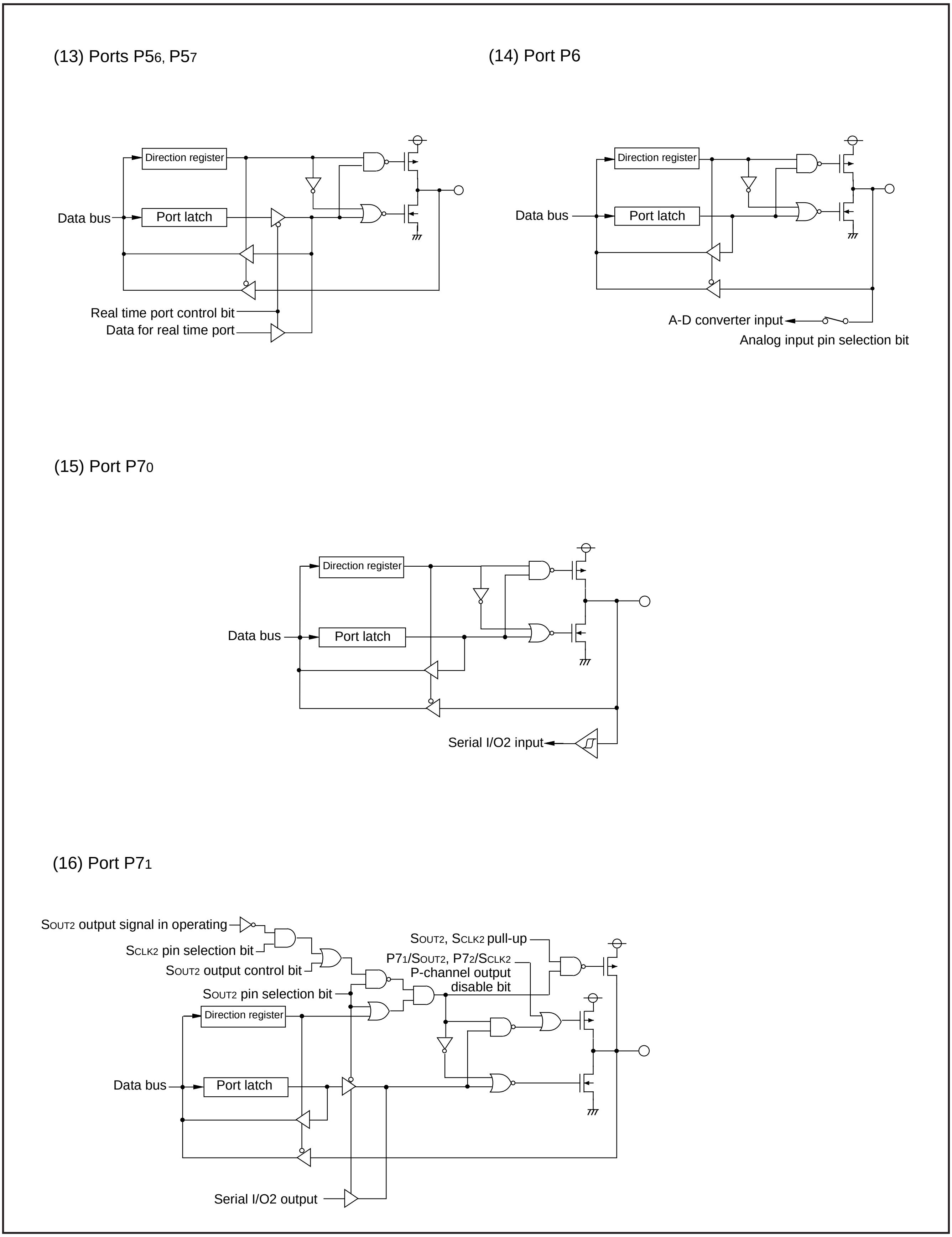
(9) Port P47

(10) Port P50

(11) Port P54

(12) Port P55

Fig. 15 Port block diagram (2)



(13) Ports $P5_6$, $P5_7$

(14) Port P6

(15) Port $P7_0$

(16) Port $P7_1$

Fig. 16 Port block diagram (3)

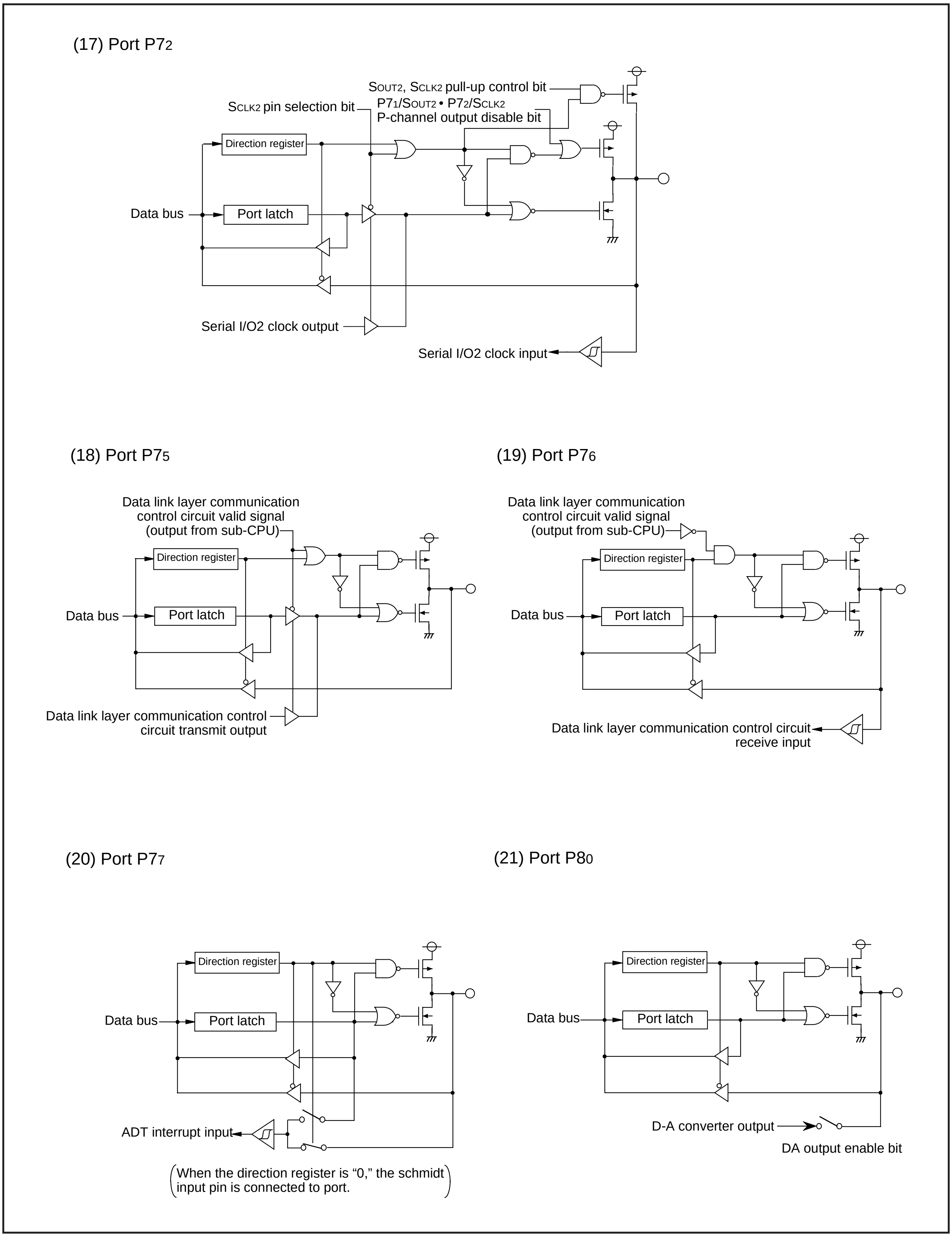
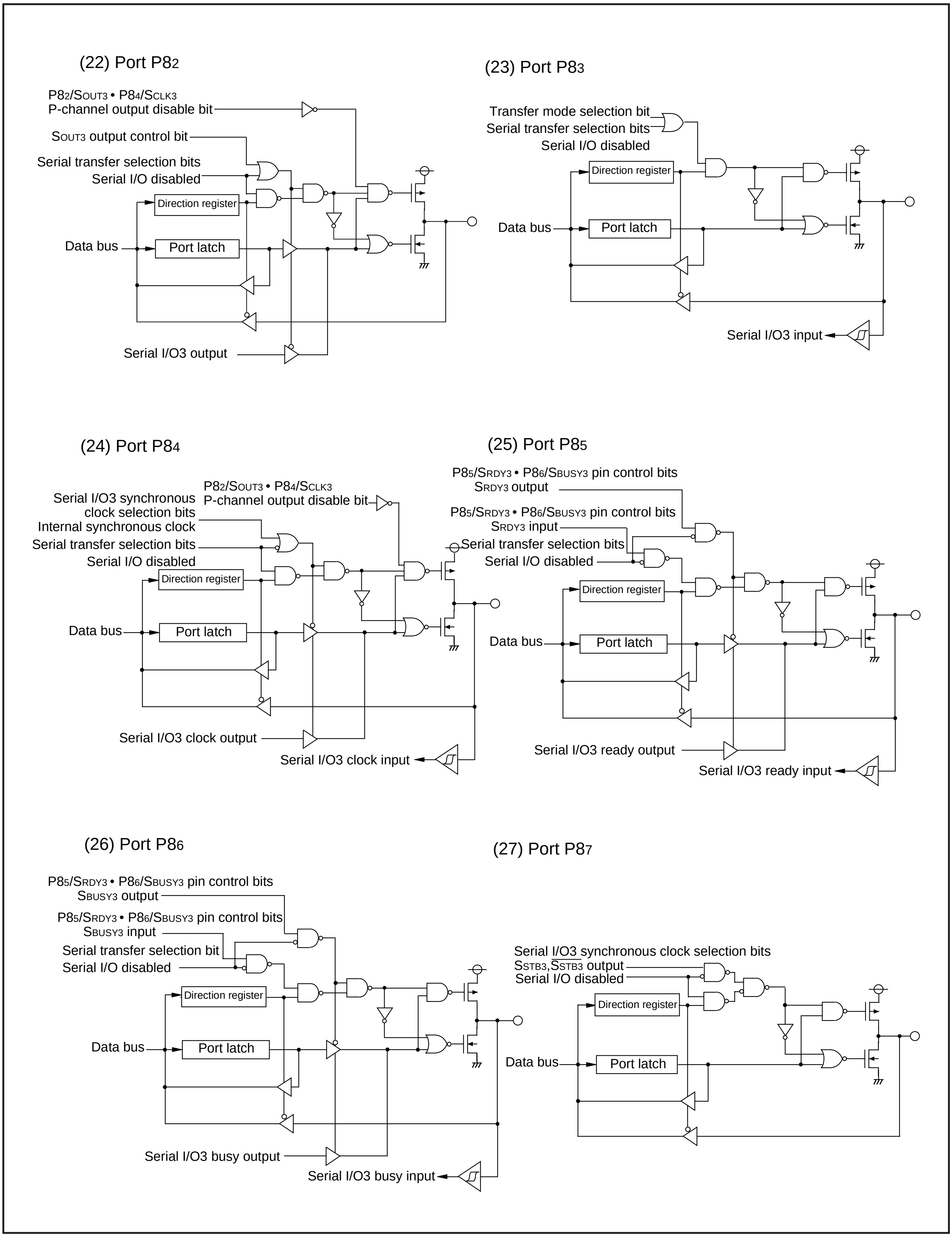

(17) Port $P7_2$
S_{OUT2}, S_{CLK2} pull-up control bit
S_{CLK2} pin selection bit
$P7_1/S_{OUT2}$ • $P7_2/S_{CLK2}$
P-channel output disable bit
Direction register
Data bus
Port latch
Serial I/O2 clock output
Serial I/O2 clock input
(18) Port $P7_5$
Data link layer communication control circuit valid signal (output from sub-CPU)
Direction register
Data bus
Port latch
Data link layer communication control circuit transmit output
(19) Port $P7_6$
Data link layer communication control circuit valid signal (output from sub-CPU)
Direction register
Data bus
Port latch
Data link layer communication control circuit receive input
(20) Port $P7_7$
Direction register
Data bus
Port latch
ADT interrupt input
(When the direction register is “0,” the schmidt input pin is connected to port.)
(21) Port $P8_0$
Direction register
Data bus
Port latch
D-A converter output
DA output enable bit

Fig. 17 Port block diagram (4)



(22) Port P82

(23) Port P83

(24) Port P84

(25) Port P85

(26) Port P86

(27) Port P87

Fig. 18 Port block diagram (5)

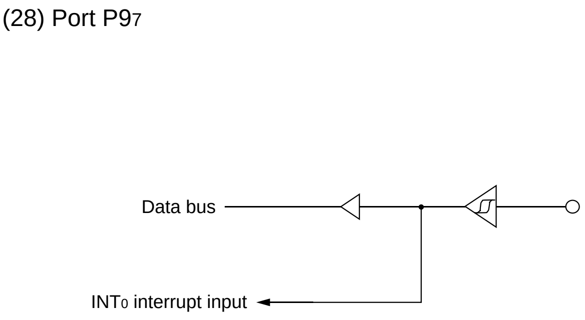


Fig. 19 Port block diagram (6)

INTERRUPTS

Interrupts occur by 27 sources: 10 external, 16 internal, and 1 software.

Interrupt Control

Each interrupt is controlled by an interrupt request bit, an interrupt enable bit, and the interrupt disable flag except for the software interrupt set by the BRK instruction. An interrupt occurs if the corresponding interrupt request and enable bits are "1" and the interrupt disable flag is "0".

Interrupt enable bits can be set or cleared by software.

Interrupt request bits can be cleared by software, but cannot be set by software.

The BRK instruction cannot be disabled with any flag or bit. The I (interrupt disable) flag disables all interrupts except the BRK instruction interrupt.

The interrupt control circuit consists of two types of interrupts: "one factor/one vector interrupt" and "multiple factors/one vector interrupt". The configuration is shown in Figure 20.

Interrupt Operation

When an interrupt occurs, the following operations are automatically performed:

1. The contents of the program counter and the processor status register are pushed onto the stack.
2. The interrupt disable flag is set and the corresponding interrupt request bit for each vector is cleared. (The corresponding interrupt request bit for each interrupt factor is not cleared.)
3. The interrupt jump destination address of interrupt which has the highest priority is loaded to the program counter.

Interrupt Factor Discrimination

The interrupt request bit for each vector of "multiple factors/one vector interrupt" is set to "1" when the interrupt disable flag (I) is "0" and one of the factor interrupt enable bits is "1" and the corresponding factor interrupt request bit changes from "0" to "1". At this time, if the vector interrupt enable bit is "1", the interrupt occurs. (Note that the interrupt request bit for each vector and the factor interrupt request bit are both edge sense.)

When 2 or more interrupt requests of interrupt factors assigned to one interrupt vector are generated at the same time, confirm the interrupt request bits for each interrupt factor assigned to the vector, and process according to the priority.

If the interrupt request bit for the interrupt factor is "1" and the interrupt enable bits for interrupt factor and each vector are both "1"; for example, when an interrupt of another interrupt factor assigned to the same vector occurs while an interrupt processing routine is executed, the interrupt occurs again after returning. Clear the interrupt request bits which are not necessary or which have been already processed before executing the interrupt flag clear (CLI) or interrupt processing routine return (RTI) instruction.

The interrupt request bits for each interrupt factor are not cleared by hardware after an interrupt vector address branching. Clear these bits by software in the interrupt processing routine. Use the LDM, STA, etc. instructions to do it. Do not use the read- modify-write instruction; for example, the CLB.

■ Notes

When the active edge of an external interrupt (INT_0–INT_5, $CNTR_0$, $CNTR_1$) is set, the corresponding interrupt request bit may also be set. Therefore, take following sequence:

(1) Disable the external interrupt which is selected.
(2) Change the active edge in interrupt edge selection register (in case of $CNTR_0$: Timer X mode register; in case of $CNTR_1$: Timer Y mode register).
(3) Clear the set interrupt request bit to "0".
(4) Enable the external interrupt which is selected.

Table 9 Interrupt vector addresses and priority

Interrupt Sources	Priority	Vector Addresses **(Note 1)**		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset **(Note 2)**	1	$FFFD_{16}$	$FFFC_{16}$	At reset	Non-maskable
INT$_0$	2	$FFFB_{16}$	$FFFA_{16}$	At detection of either rising or falling edge of INT$_0$ input	External interrupt (active edge selectable)
INT$_1$	3	$FFF9_{16}$	$FFF8_{16}$	At detection of either rising or falling edge of INT$_1$ input	External interrupt (active edge selectable)
Receive bus interrupt source 1	4	$FFF7_{16}$	$FFF6_{16}$	When receive bus interrupt source 1 request bit becomes “1” from “0”	The condition which the receive bus interrupt factor request bit becomes “1” is defined according to each communication protocol specification confirmation.
Receive bus interrupt source 2				When receive bus interrupt source 2 request bit becomes “1” from “0”	
Receive bus interrupt source 3				When receive bus interrupt source 3 request bit becomes “1” from “0”	
Transmit bus interrupt source 1	5	$FFF5_{16}$	$FFF4_{16}$	When transmit bus interrupt source 1 request bit becomes “1” from “0”	The condition which the transmit bus interrupt factor request bit becomes “1” is defined according to each communication protocol specification confirmation.
Transmit bus interrupt source 2				When transmit bus interrupt source 2 request bit becomes “1” from “0”	
Transmit bus interrupt source 3				When transmit bus interrupt source 3 request bit becomes “1” from “0”	
Timer X	6	$FFF3_{16}$	$FFF2_{16}$	At timer X underflow	
Timer Y	7	$FFF1_{16}$	$FFF0_{16}$	At timer Y underflow	
Timer 2	8	$FFEF_{16}$	$FFEE_{16}$	At timer 2 underflow	
Timer 3	9	$FFED_{16}$	$FFEC_{16}$	At timer 3 underflow	
INT$_2$	10	$FFEB_{16}$	$FFEA_{16}$	At detection of either rising or falling edge of INT$_2$ input	External interrupt (active edge selectable)
Serial I/O3 interrupt	11	$FFE9_{16}$	$FFE8_{16}$	At completion of serial I/O3 data transmission/reception	Valid only when serial I/O3 is selected
CNTR$_0$				At detection of either rising or falling edge of CNTR$_0$ input	External interrupt (active edge selectable)
CNTR$_1$	12	$FFE7_{16}$	$FFE6_{16}$	At detection of either rising or falling edge of CNTR$_1$ input	External interrupt (active edge selectable)
Timer 1	13	$FFE5_{16}$	$FFE4_{16}$	At timer 1 underflow	
INT$_3$	14	$FFE3_{16}$	$FFE2_{16}$	At detection of either rising or falling edge of INT$_3$ input	External interrupt (active edge selectable)
INT$_4$				At detection of either rising or falling edge of INT$_4$ input	External interrupt (active edge selectable)
INT$_5$				At detection of either rising or falling edge of INT$_5$ input	External interrupt (active edge selectable)
ADT	15	$FFE1_{16}$	$FFE0_{16}$	At falling of ADT pin input	Valid only when ADT interrupt is selected External interrupt (falling valid)
A-D converter				At completion of A-D converter	Valid only when A-D converter interrupt is selected
Serial I/O2 interrupt				At completion of serial I/O2 data transmission/reception	Valid only when serial I/O2 is selected
Key input (key-on wake-up)	16	$FFDF_{16}$	$FFDE_{16}$	At falling of port P2$_0$ to P2$_7$ (at input) input logical level AND	External interrupt (falling valid)
Serial I/O1 receive				At completion of serial I/O1 data reception	Valid only when serial I/O1 is selected
Serial I/O1 transmit				At completion of serial I/O1 transmission shift or when transmission buffer is empty	Valid only when serial I/O1 is selected
BRK instruction	17	$FFDD_{16}$	$FFDC_{16}$	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.

3: Either ADT interrupt or A-D converter interrupt can be used. Both ADT interrupt and A-D converter interrupt cannot be used.

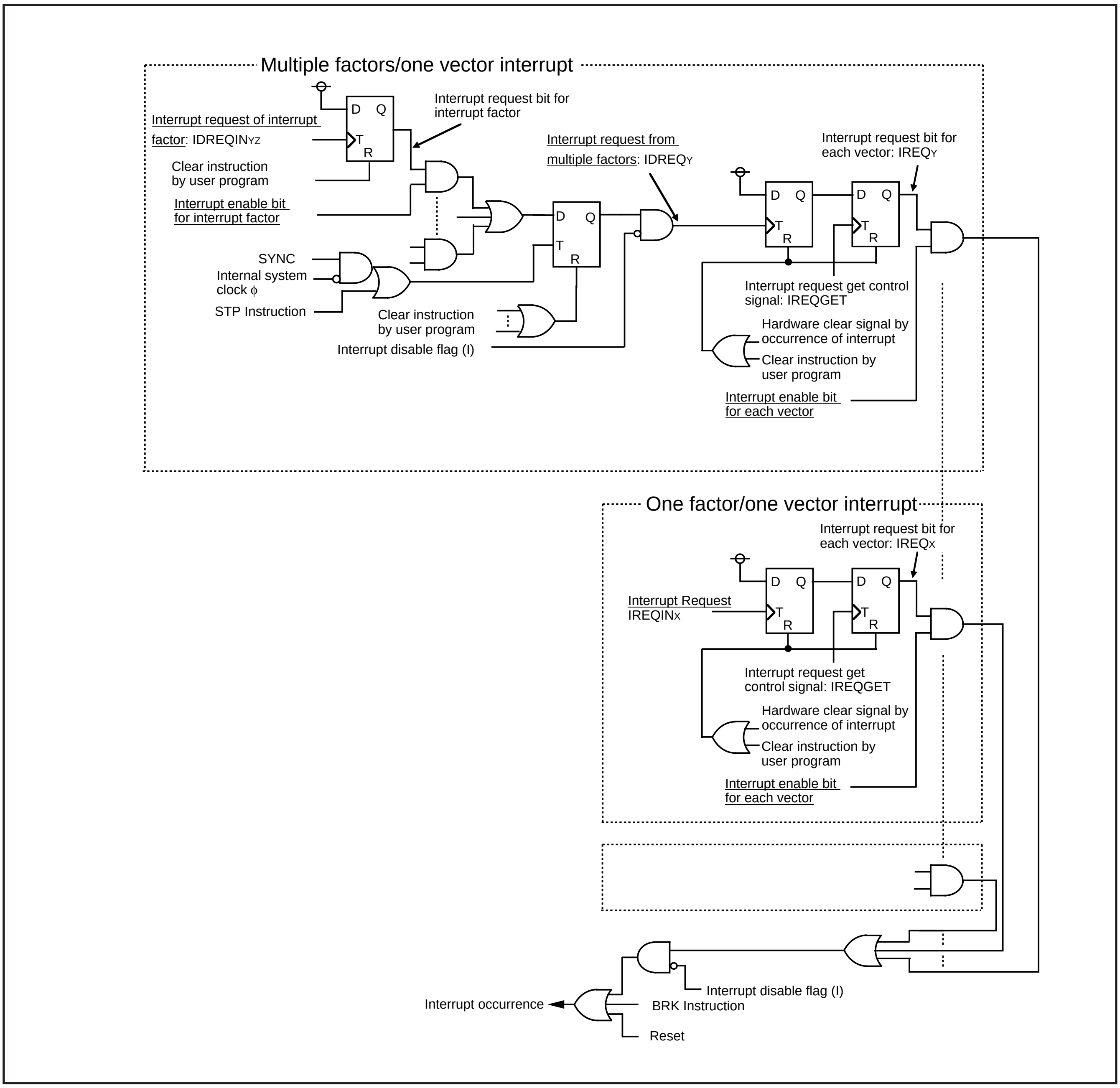

Multiple factors/one vector interrupt
D Q
T R
Interrupt request bit for interrupt factor
Interrupt request of interrupt factor: IDREQINYZ
Clear instruction by user program
Interrupt enable bit for interrupt factor
Interrupt request from multiple factors: IDREQY
Interrupt request bit for each vector: IREQY
SYNC
Internal system clock ϕ
STP Instruction
Clear instruction by user program
Interrupt disable flag (I)
Interrupt request get control signal: IREQGET
Hardware clear signal by occurrence of interrupt
Clear instruction by user program
Interrupt enable bit for each vector
One factor/one vector interrupt
Interrupt request bit for each vector: IREQX
Interrupt Request IREQINX
Interrupt request get control signal: IREQGET
Hardware clear signal by occurrence of interrupt
Clear instruction by user program
Interrupt enable bit for each vector
Interrupt disable flag (I)
Interrupt occurrence
BRK Instruction
Reset

Fig. 20 Interrupt control diagram

Timing to Interrupt Request Acceptance

The cycle number of internal system clock required from occurrence to acceptance of an interrupt request depends on the type of interrupt: “multiple factors/one vector” or “one factor/one vector”.

For “one factor/one vector interrupt”, the CPU starts processing the management after interrupt acceptance at the next instruction execution timing (rising edge of SYNC signal) immediately after the interrupt request is generated. For “multiple factors/one vector interrupt”, the CPU starts processing the management after interrupt acceptance at the second instruction execution timing (rising edge of SYNC signal) after the interrupt request for interrupt source discrimination is generated. In other words, “multiple factors/one vector interrupt” required one instruction execution cycle number (2 to 16 cycles of internal system clock) more than that of “one factor/one vector interrupt” to begin the interrupt sequence.

Figure 20 shows the interrupt control diagram and Figure 21 shows the timing from occurrence to acceptance of interrupt request.

For “one factor/one vector interrupt”, the interrupt request is generated at Timing (A) and the processing after acceptance begins at Timing (B). For “multiple factors/one vector interrupt”, the interrupt factor determination request is generated at Timing (C), the interrupt request is generated at Timing (D), and the processing after acceptance begins at Timing (E).

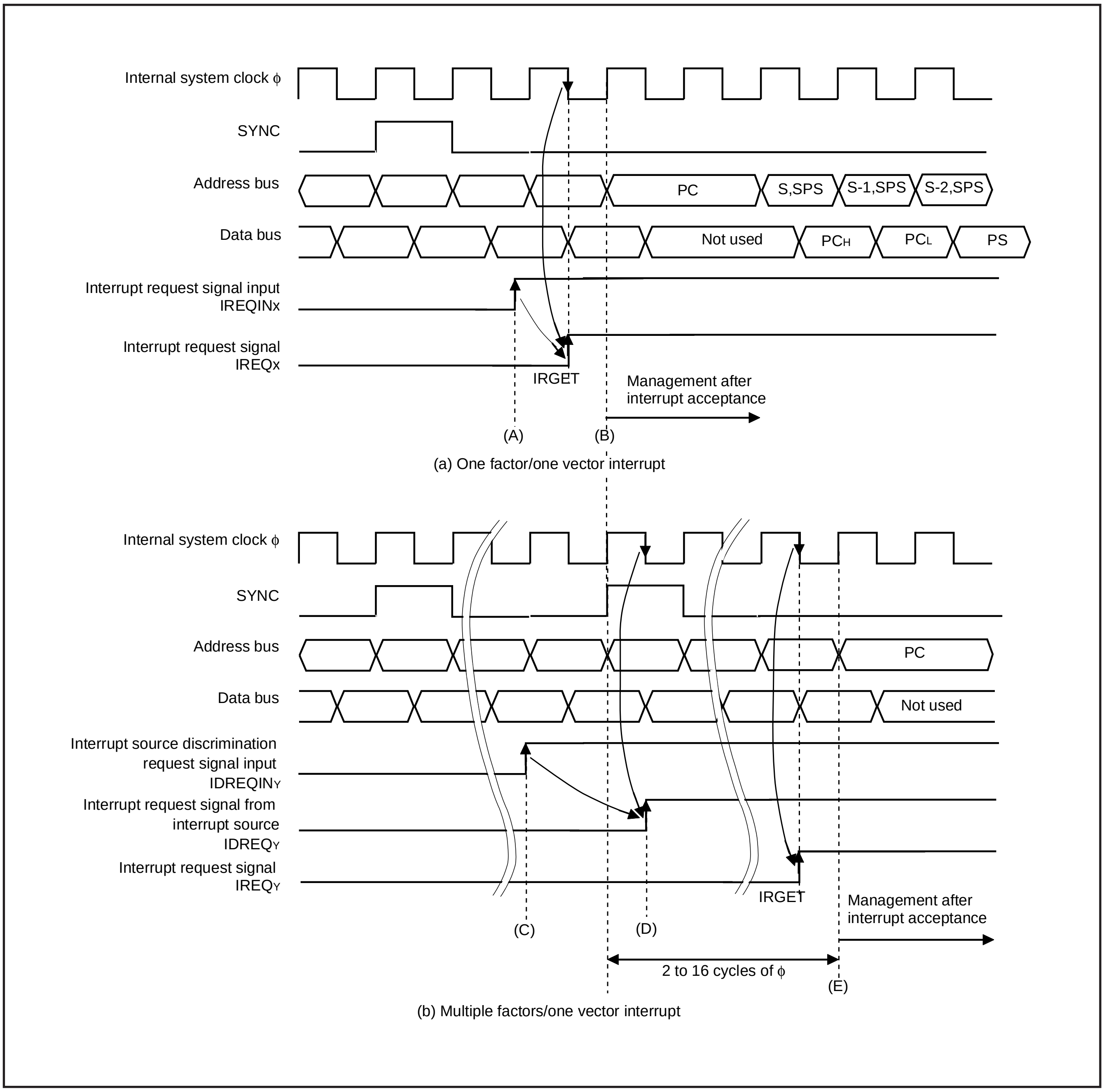


Fig. 21 Timing from occurrence to acceptance of interrupt

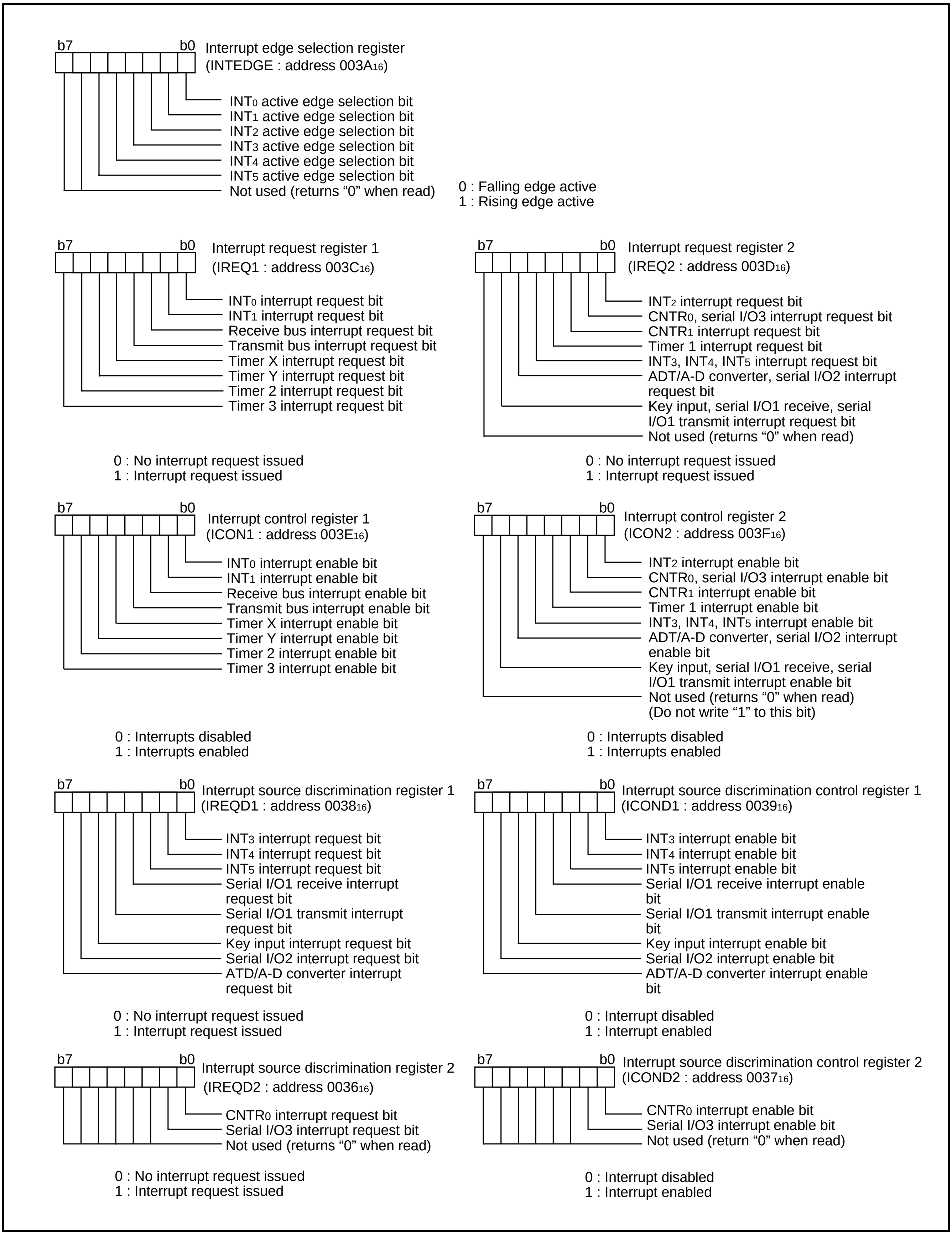

b7
b0
Interrupt edge selection register
(INTEDGE : address 003A$_{16}$)
INT$_0$ active edge selection bit
INT$_1$ active edge selection bit
INT$_2$ active edge selection bit
INT$_3$ active edge selection bit
INT$_4$ active edge selection bit
INT$_5$ active edge selection bit
Not used (returns “0” when read)
0 : Falling edge active
1 : Rising edge active
b7
b0
Interrupt request register 1
(IREQ1 : address 003C$_{16}$)
INT$_0$ interrupt request bit
INT$_1$ interrupt request bit
Receive bus interrupt request bit
Transmit bus interrupt request bit
Timer X interrupt request bit
Timer Y interrupt request bit
Timer 2 interrupt request bit
Timer 3 interrupt request bit
0 : No interrupt request issued
1 : Interrupt request issued
b7
b0
Interrupt request register 2
(IREQ2 : address 003D$_{16}$)
INT$_2$ interrupt request bit
CNTR$_0$, serial I/O3 interrupt request bit
CNTR$_1$ interrupt request bit
Timer 1 interrupt request bit
INT$_3$, INT$_4$, INT$_5$ interrupt request bit
ADT/A-D converter, serial I/O2 interrupt request bit
Key input, serial I/O1 receive, serial I/O1 transmit interrupt request bit
Not used (returns “0” when read)
0 : No interrupt request issued
1 : Interrupt request issued
b7
b0
Interrupt control register 1
(ICON1 : address 003E$_{16}$)
INT$_0$ interrupt enable bit
INT$_1$ interrupt enable bit
Receive bus interrupt enable bit
Transmit bus interrupt enable bit
Timer X interrupt enable bit
Timer Y interrupt enable bit
Timer 2 interrupt enable bit
Timer 3 interrupt enable bit
0 : Interrupts disabled
1 : Interrupts enabled
b7
b0
Interrupt control register 2
(ICON2 : address 003F$_{16}$)
INT$_2$ interrupt enable bit
CNTR$_0$, serial I/O3 interrupt enable bit
CNTR$_1$ interrupt enable bit
Timer 1 interrupt enable bit
INT$_3$, INT$_4$, INT$_5$ interrupt enable bit
ADT/A-D converter, serial I/O2 interrupt enable bit
Key input, serial I/O1 receive, serial I/O1 transmit interrupt enable bit
Not used (returns “0” when read)
(Do not write “1” to this bit)
0 : Interrupts disabled
1 : Interrupts enabled
b7
b0
Interrupt source discrimination register 1
(IREQD1 : address 0038$_{16}$)
INT$_3$ interrupt request bit
INT$_4$ interrupt request bit
INT$_5$ interrupt request bit
Serial I/O1 receive interrupt request bit
Serial I/O1 transmit interrupt request bit
Key input interrupt request bit
Serial I/O2 interrupt request bit
ATD/A-D converter interrupt request bit
0 : No interrupt request issued
1 : Interrupt request issued
b7
b0
Interrupt source discrimination control register 1
(ICOND1 : address 0039$_{16}$)
INT$_3$ interrupt enable bit
INT$_4$ interrupt enable bit
INT$_5$ interrupt enable bit
Serial I/O1 receive interrupt enable bit
Serial I/O1 transmit interrupt enable bit
Key input interrupt enable bit
Serial I/O2 interrupt enable bit
ADT/A-D converter interrupt enable bit
0 : Interrupt disabled
1 : Interrupt enabled
b7
b0
Interrupt source discrimination register 2
(IREQD2 : address 0036$_{16}$)
CNTR$_0$ interrupt request bit
Serial I/O3 interrupt request bit
Not used (returns “0” when read)
0 : No interrupt request issued
1 : Interrupt request issued
b7
b0
Interrupt source discrimination control register 2
(ICOND2 : address 0037$_{16}$)
CNTR$_0$ interrupt enable bit
Serial I/O3 interrupt enable bit
Not used (return “0” when read)
0 : Interrupt disabled
1 : Interrupt enabled

Fig. 22 Structure of interrupt-related registers

Key Input Interrupt

A Key input interrupt request is generated by applying "L" level to any pin of port P2 that have been set to input mode. In other words, it is generated when AND of input level goes from "1" to "0".

An example of using a key input interrupt is shown in Figure 23, where an interrupt request is generated by pressing one of the keys consisted as an active-low key matrix which inputs to ports $P2_0$–$P2_4$.

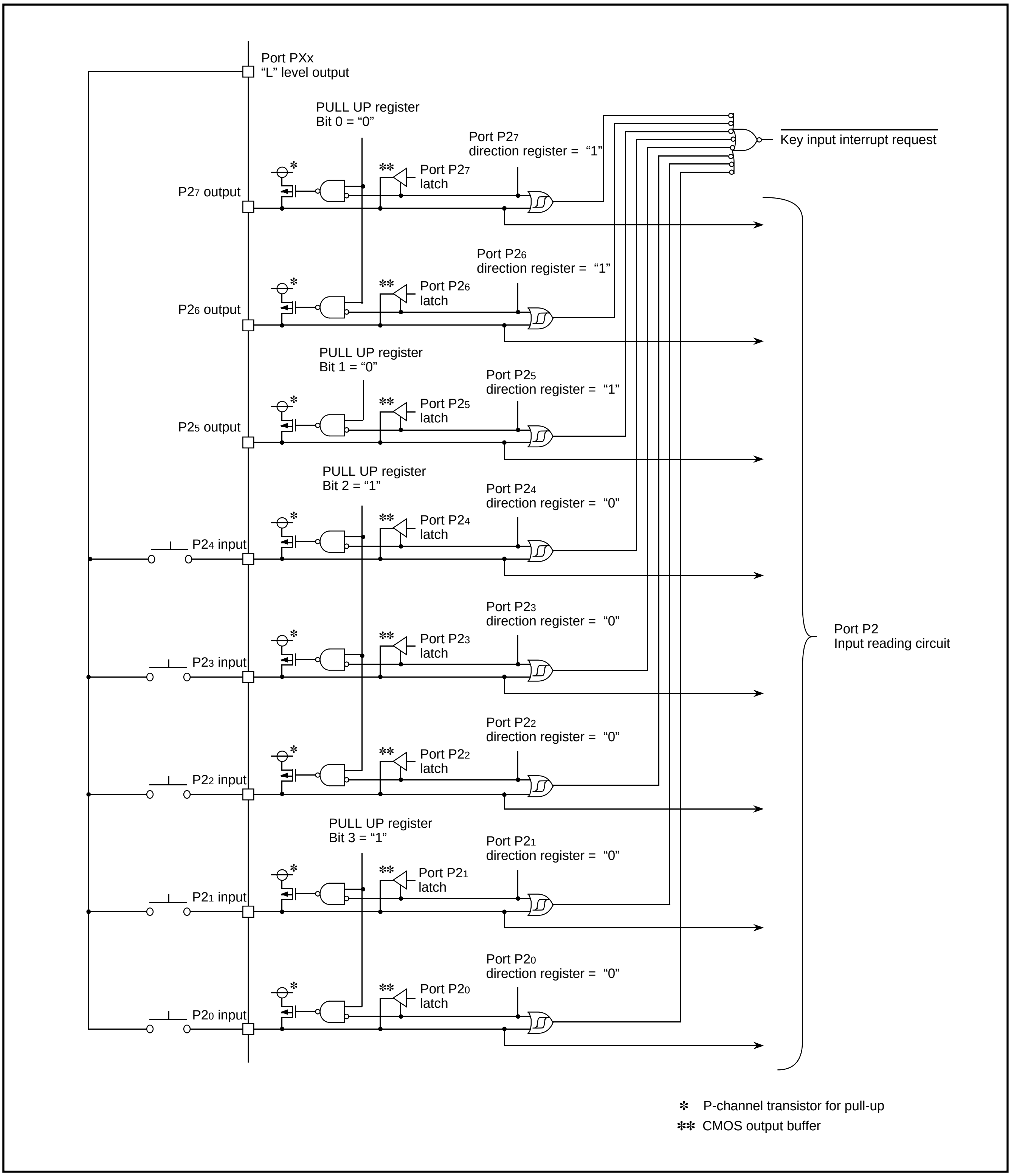


Fig. 23 Connection example when using key input interrupt and port P2 block diagram

TIMERS

The 3874 group has five timers: timer X, timer Y, timer 1, timer 2, and timer 3. Timer X and timer Y are 16-bit timers, and timer 1, timer 2, and timer 3 are 8-bit timers.

All timers are down count timers. When the timer reaches “00$_{16}$” or “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 a timer underflows, the interrupt request bit corresponding to that timer is set to “1”.

Read and write operation on 16-bit timer must be performed for both high and low-order bytes. When reading a 16-bit timer, read the high-order byte first. When writing to a 16-bit timer, write the low-order byte first. The 16-bit timer cannot perform the correct operation when reading during the write operation, or when writing during the read operation.

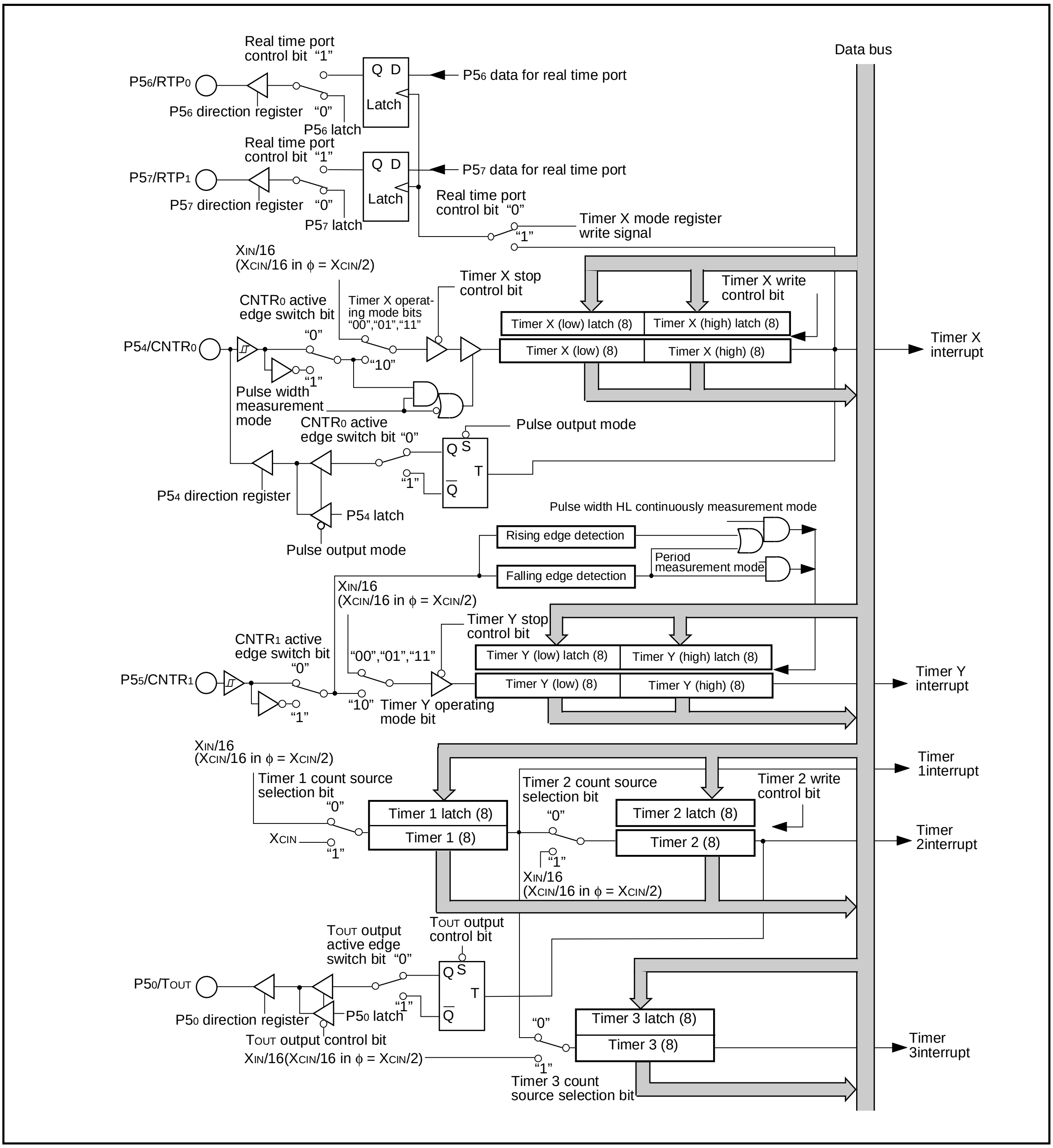

Real time port control bit “1”
P5$_6$/RTP$_0$
P5$_6$ direction register “0”
P5$_6$ latch
Q D
Latch
P5$_6$ data for real time port
Data bus
Real time port control bit “1”
P5$_7$/RTP$_1$
P5$_7$ direction register “0”
P5$_7$ latch
Q D
Latch
P5$_7$ data for real time port
Real time port control bit “0”
“1”
Timer X mode register write signal
X$_{IN}$/16 (X$_{CIN}$/16 in ϕ = X$_{CIN}$/2)
Timer X stop control bit
CNTR$_0$ active edge switch bit
Timer X operating mode bits “00”,“01”,“11”
Timer X write control bit
“0”
Timer X (low) latch (8)
Timer X (high) latch (8)
P5$_4$/CNTR$_0$
“10”
Timer X (low) (8)
Timer X (high) (8)
Timer X interrupt
“1”
Pulse width measurement mode
CNTR$_0$ active edge switch bit “0”
Pulse output mode
Q S
T
Q
“1”
P5$_4$ direction register
P5$_4$ latch
Pulse output mode
Pulse width HL continuously measurement mode
Rising edge detection
Falling edge detection
Period measurement mode
X$_{IN}$/16 (X$_{CIN}$/16 in ϕ = X$_{CIN}$/2)
Timer Y stop control bit
CNTR$_1$ active edge switch bit “0”
“00”,“01”,“11”
Timer Y (low) latch (8)
Timer Y (high) latch (8)
P5$_5$/CNTR$_1$
Timer Y (low) (8)
Timer Y (high) (8)
Timer Y interrupt
“1”
“10” Timer Y operating mode bit
X$_{IN}$/16 (X$_{CIN}$/16 in ϕ = X$_{CIN}$/2)
Timer 1 count source selection bit “0”
Timer 1latch (8)
Timer 1 (8)
X$_{CIN}$
“1”
Timer 2 count source selection bit “0”
Timer 2 latch (8)
Timer 2 (8)
Timer 2 write control bit
Timer 1interrupt
Timer 2interrupt
“1”
X$_{IN}$/16 (X$_{CIN}$/16 in ϕ = X$_{CIN}$/2)
T$_{OUT}$ output active edge switch bit “0”
T$_{OUT}$ output control bit
P5$_0$/T$_{OUT}$
Q S
T
Q
“1”
P5$_0$ direction register
P5$_0$ latch
T$_{OUT}$ output control bit
X$_{IN}$/16(X$_{CIN}$/16 in ϕ = X$_{CIN}$/2)
“0”
Timer 3 latch (8)
Timer 3 (8)
“1”
Timer 3 count source selection bit
Timer 3interrupt

Fig. 24 Timer block diagram

Timer X

Timer X is a 16-bit timer that can be selected in one of four modes and can be controlled the timer X write and the real time port by setting the timer X mode register.

(1) Timer Mode

The timer counts f(X_{IN})/16 (or f(X_{CIN})/16 in system clock ϕ = X_{CIN}/2).

(2) Pulse Output Mode

Each time the timer underflows, a signal output from the $CNTR_0$ pin is inverted. Except for this, the operation in pulse output mode is the same as in timer mode. When using a timer in this mode, set the direction register of corresponding port to output mode.

(3) Event Counter Mode

The timer counts signals input through the $CNTR_0$ pin.

Except for this, the operation in event counter mode is the same as in timer mode.

(4) Pulse Width Measurement Mode

The count source is f(X_{IN})/16 (or f(X_{CIN})/16 in system clock ϕ = X_{CIN}/2. If $CNTR_0$ active edge switch bit is “0”, the timer counts while the input signal of $CNTR_0$ pin is at “H”. If it is “1”, the timer counts while the input signal of $CNTR_0$ pin is at “L”.

■ Notes

● **Timer X write control**

If the timer X write control bit is “1”, when the value is written in the address of timer X, the value is loaded only in the latch. The value in the latch is loaded in timer X after timer X underflows.

If the timer X write control bit is “0”, when the value is written in the address of timer X, the value is loaded in the timer X and the latch at the same time.

When the value is to be written in latch only, if the value is written to the latch at timer X underflows, the value is consequently loaded in the timer X and the latch at the same time. Unexpected value may be set in the high-order counter when the writing in high-order latch and the underflow of timer X are performed at the same timing.

● **$CNTR_0$ interrupt active edge selection**

$CNTR_0$ interrupt active edge depends on the $CNTR_0$ active edge switch bit.

● **Real time port control**

Data for the real time port are output from ports $P5_6$ and $P5_7$ each time the timer X underflows. (However, if the real time port control bit is changed from “0” to “1”, data are output independent of the timer X operation.) When the data for the real time port is changed while the real time port function is valid, the changed data are output at the next underflow of timer X.

Before using this function, set the corresponding port direction registers to output mode.

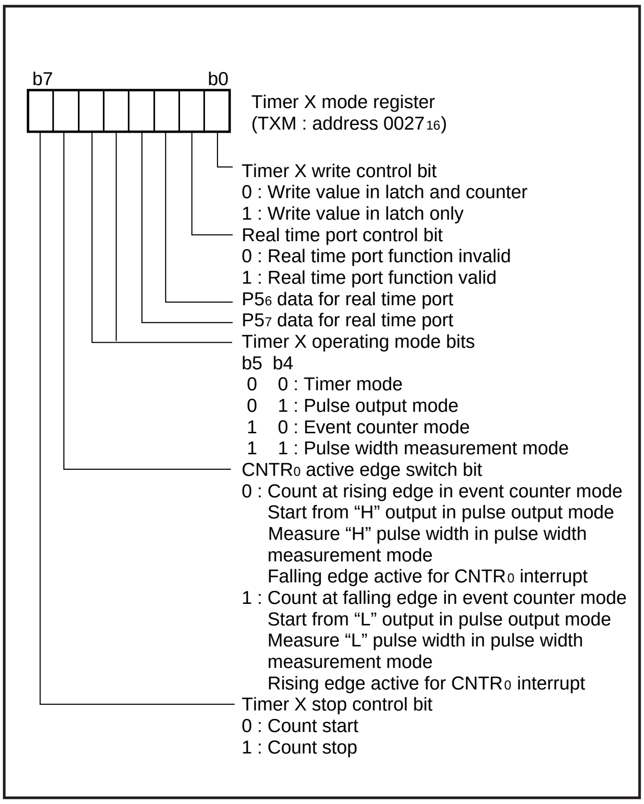

b7
b0
Timer X mode register
(TXM : address 0027_{16})
Timer X write control bit
0 : Write value in latch and counter
1 : Write value in latch only
Real time port control bit
0 : Real time port function invalid
1 : Real time port function valid
$P5_6$ data for real time port
$P5_7$ data for real time port
Timer X operating mode bits
b5 b4
0 0 : Timer mode
0 1 : Pulse output mode
1 0 : Event counter mode
1 1 : Pulse width measurement mode
$CNTR_0$ active edge switch bit
0 : Count at rising edge in event counter mode
Start from “H” output in pulse output mode
Measure “H” pulse width in pulse width measurement mode
Falling edge active for $CNTR_0$ interrupt
1 : Count at falling edge in event counter mode
Start from “L” output in pulse output mode
Measure “L” pulse width in pulse width measurement mode
Rising edge active for $CNTR_0$ interrupt
Timer X stop control bit
0 : Count start
1 : Count stop

Fig. 25 Structure of timer X mode register

Timer Y

Timer Y is a 16-bit timer that can be selected in one of four modes.

(1) Timer Mode

The timer counts f(X$_{IN}$)/16 (or f(X$_{CIN}$)/16 in system clock ϕ = X$_{CIN}$/2).

(2) Period Measurement Mode

CNTR$_1$ interrupt request is generated at rising/falling edge of CNTR$_1$ pin input signal. Simultaneously, the value in timer Y latch is reloaded in timer Y and timer Y continues counting down. Except for the above-mentioned, the operation in period measurement mode is the same as in timer mode.
The timer value just before the reloading at rising/falling of CNTR$_1$ pin input signal is retained until the timer Y is read once after the reload.
The rising/falling timing of CNTR$_1$ pin input signal is found by CNTR$_1$ interrupt.

(3) Event Counter Mode

The timer counts signals input through the CNTR$_1$ pin.
Except for this, the operation in event counter mode is the same as in timer mode.

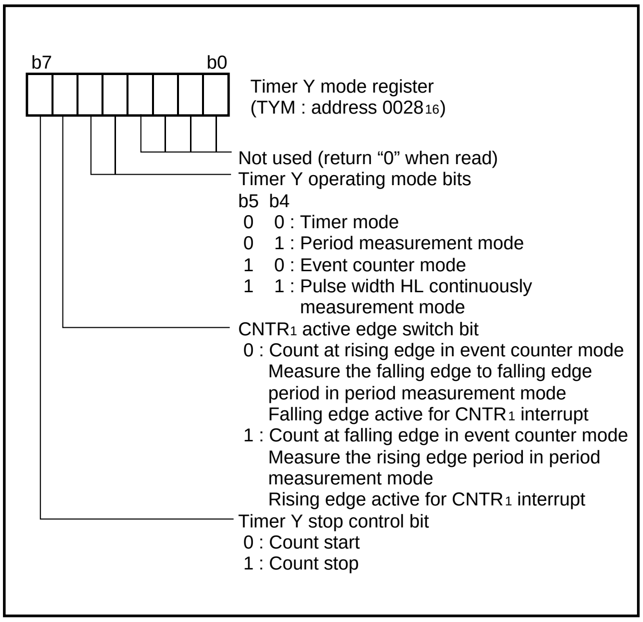
(4) Pulse Width HL Continuously Measurement Mode

CNTR$_1$ interrupt request is generated at both rising and falling edges of CNTR$_1$ pin input signal. Except for this, the operation in pulse width HL continuously measurement mode is the same as in period measurement mode.

■ Note

● CNTR$_1$ interrupt active edge selection

CNTR$_1$ interrupt active edge depends on the CNTR$_1$ active edge switch bit. However, in pulse width HL continuously measurement mode, CNTR$_1$ interrupt request is generated at both rising and falling edges of CNTR$_1$ pin input signal regardless of the setting of CNTR$_1$ active edge switch bit.



Timer Y mode register
(TYM : address 0028$_{16}$)

Not used (return “0” when read)

Timer Y operating mode bits
b5 b4
0 0 : Timer mode
0 1 : Period measurement mode
1 0 : Event counter mode
1 1 : Pulse width HL continuously measurement mode

CNTR$_1$ active edge switch bit
0 : Count at rising edge in event counter mode
Measure the falling edge to falling edge period in period measurement mode
Falling edge active for CNTR$_1$ interrupt
1 : Count at falling edge in event counter mode
Measure the rising edge period in period measurement mode
Rising edge active for CNTR$_1$ interrupt

Timer Y stop control bit
0 : Count start
1 : Count stop

Fig. 26 Structure of timer Y mode register

Timer 1, Timer 2, Timer 3

Timer 1, timer 2, and timer 3 are 8-bit timers. The count source for each timer can be selected by timer 123 mode register.

● **Timer 2 write control**

When the timer 2 write control bit is “1”, and the value is written in the address of timer 2, the value is loaded only in the latch. The value in the latch is loaded in timer 2 after timer 2 underflows.
When the timer 2 write control bit is “0”, and the value is written in the address of timer 2, the value is loaded in the timer 2 and the latch at the same time.

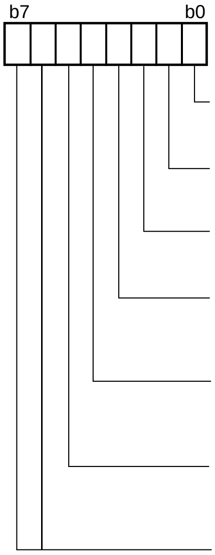
● **Timer 2 output control**

An inversion signal from TOUT pin is output each time timer 2 underflows.
In this case, set the port $P5_0$ direction register to the output mode.

■ Note

● **Timer 1 to timer 3**

When the count source of timer 1 to 3 is changed, the timer counting value may be changed large because a thin pulse is generated in count input of timer. If timer 1 output is selected as the count source of timer 2 or timer 3, when timer 1 is written, the counting value of timer 2 or timer 3 may be changed large because a thin pulse is generated in timer 1 output.
Therefore, set the value of timer in the order of timer 1, timer 2 and timer 3 after the count source selection of timer 1 to 3.



Timer 123 mode register
(T123M :address 0029_{16})

TOUT output active edge switch bit
0 : Start at “H” output
1 : Start at “L” output

TOUT output control bit
0 : TOUT output disabled
1 : TOUT output enabled

Timer 2 write control bit
0 : Write data in latch and counter
1 : Write data in latch only

Timer 2 count source selection bit
0 : Timer 1 output
1 : $f(X_{IN})/16$
(or $f(X_{CIN})/16$ in low-speed mode)

Timer 3 count source selection bit
0 : Timer 1 output
1 : $f(X_{IN})/16$
(or $f(X_{CIN})/16$ in low-speed mode)

Timer 1 count source selection bit
0 : $f(X_{IN})/16$
(or $f(X_{CIN})/16$ in low-speed mode)
1 : $f(X_{CIN})$

Not used (return “0” when read)

Note : Internal clock ϕ is $f(X_{CIN})/2$ in the low-speed mode.

Fig. 27 Structure of timer 123 mode register

SERIAL I/O
Serial I/O1

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

(1) Clock Synchronous Serial I/O1 Mode

Clock synchronous serial I/O1 mode can be selected by setting the serial I/O1 mode selection bit (b6) of the serial I/O1 control register to “1”.

For clock synchronous serial I/O1, 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 (address 0018_{16}).

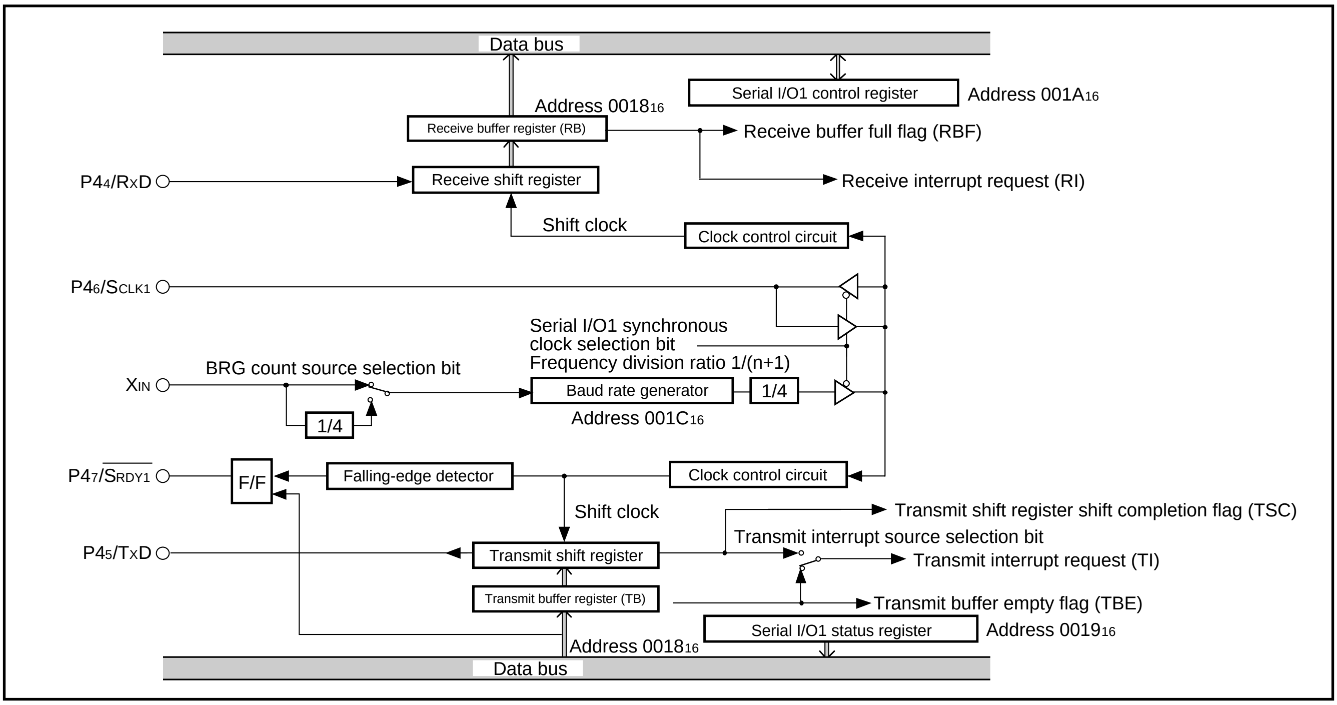
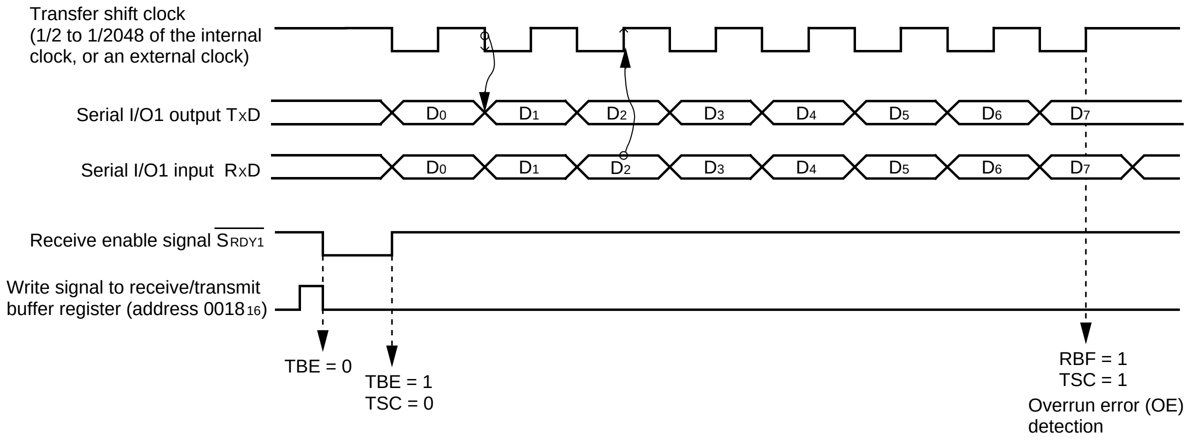

Data bus
Address 0018₁₆
Serial I/O1 control register
Address 001A₁₆
Receive buffer register (RB)
Receive buffer full flag (RBF)
P4₄/RxD
Receive shift register
Receive interrupt request (RI)
Shift clock
Clock control circuit
P4₆/SCLK1
Serial I/O1 synchronous clock selection bit
Frequency division ratio 1/(n+1)
BRG count source selection bit
XIN
Baud rate generator
1/4
1/4
Address 001C₁₆
P4₇/SRDY1
F/F
Falling-edge detector
Clock control circuit
Shift clock
Transmit shift register shift completion flag (TSC)
Transmit interrupt source selection bit
P4₅/TxD
Transmit shift register
Transmit interrupt request (TI)
Transmit buffer register (TB)
Transmit buffer empty flag (TBE)
Serial I/O1 status register
Address 0019₁₆
Address 0018₁₆
Data bus

Fig. 28 Block diagram of clock synchronous serial I/O


Transfer shift clock
(1/2 to 1/2048 of the internal clock, or an external clock)
Serial I/O1 output TxD
D0 D1 D2 D3 D4 D5 D6 D7
Serial I/O1 input RxD
D0 D1 D2 D3 D4 D5 D6 D7
Receive enable signal SRDY1
Write signal to receive/transmit buffer register (address 0018₁₆)
TBE = 0
TBE = 1
TSC = 0
RBF = 1
TSC = 1
Overrun error (OE) detection

Notes 1 : The transmit interrupt (TI) can be selected to occur either when the transmit buffer register 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. 29 Operation of clock synchronous serial I/O1 function

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

The transmit buffer 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.

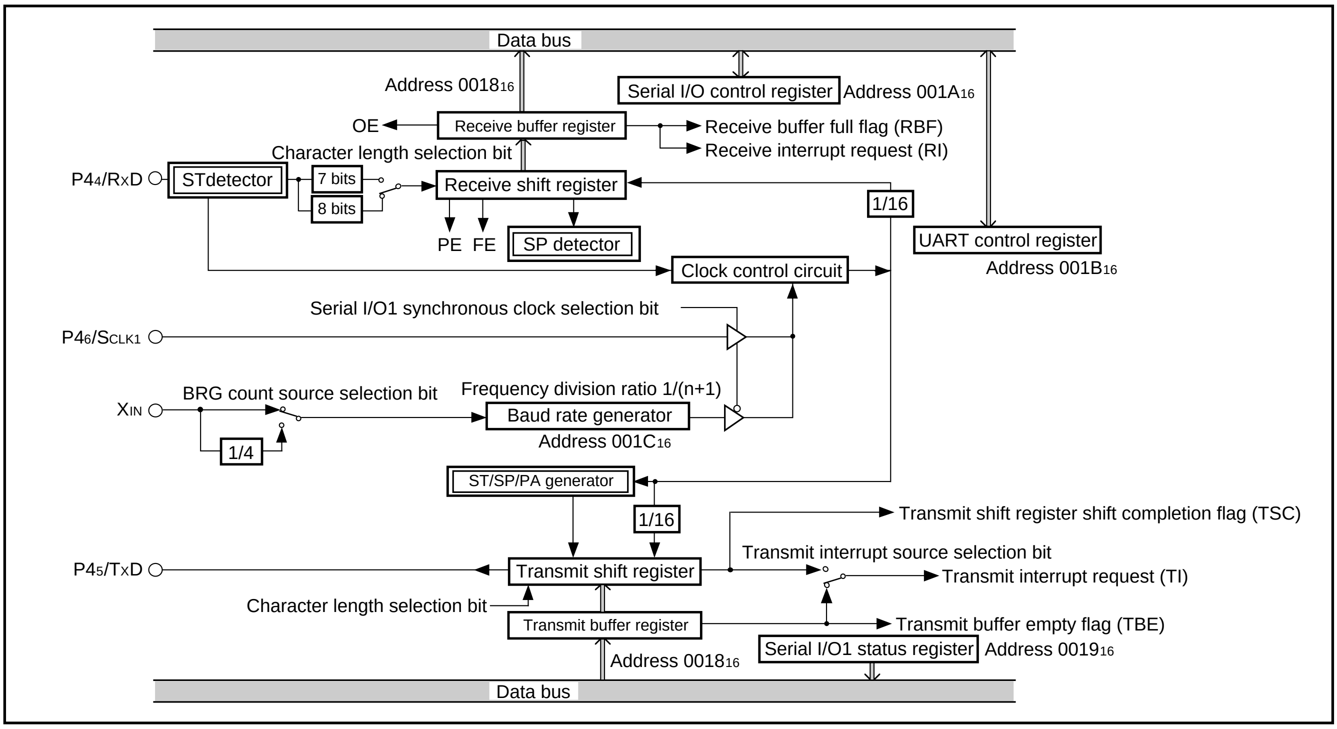
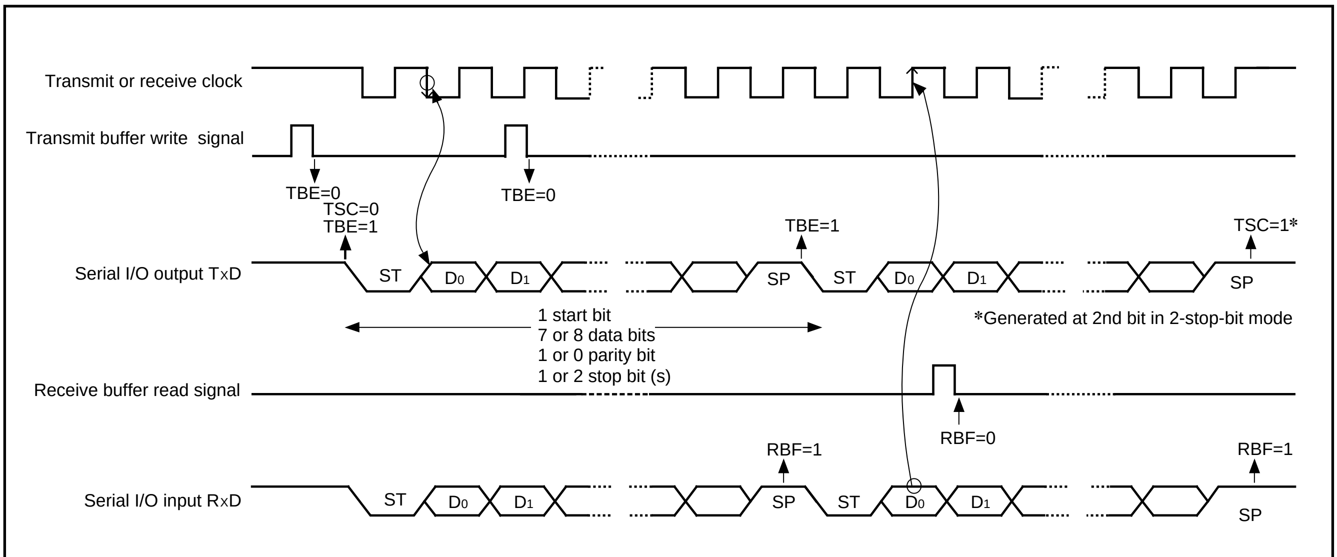

Data bus
Address 0018₁₆
Serial I/O control register
Address 001A₁₆
OE
Receive buffer register
Receive buffer full flag (RBF)
Receive interrupt request (RI)
Character length selection bit
P4₄/RxD
STdetector
7 bits
8 bits
Receive shift register
1/16
PE FE
SP detector
UART control register
Address 001B₁₆
Clock control circuit
Serial I/O1 synchronous clock selection bit
P4₆/SCLK1
BRG count source selection bit
Frequency division ratio 1/(n+1)
XIN
Baud rate generator
1/4
Address 001C₁₆
ST/SP/PA generator
1/16
Transmit shift register shift completion flag (TSC)
Transmit interrupt source selection bit
P4₅/TxD
Transmit shift register
Transmit interrupt request (TI)
Character length selection bit
Transmit buffer register
Transmit buffer empty flag (TBE)
Serial I/O1 status register
Address 0019₁₆
Address 0018₁₆
Data bus

Fig. 30 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 : The transmit interrupt (TI) can be selected to occur when either the TBE or TSC flag becomes “1”, 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 register when TSC=1, 0.5 to 1.5 cycles of the data shift cycle is necessary until changing to TSC=0.

Fig. 31 Operation of UART serial I/O function

[Transmit Buffer/Receive Buffer Register (TB/RB)] 0018_{16}

The transmit buffer register and the receive buffer register are located at the same address. The transmit buffer register is write-only and the receive buffer register is read-only. If a character bit length is 7 bits, the MSB of data stored in the receive buffer register 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/O 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 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 (bit 7) of the Serial I/O1 control register also clears all the status flags, including the error flags.

All bits 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 register 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 contains eight control bits for the serial I/O1 function.

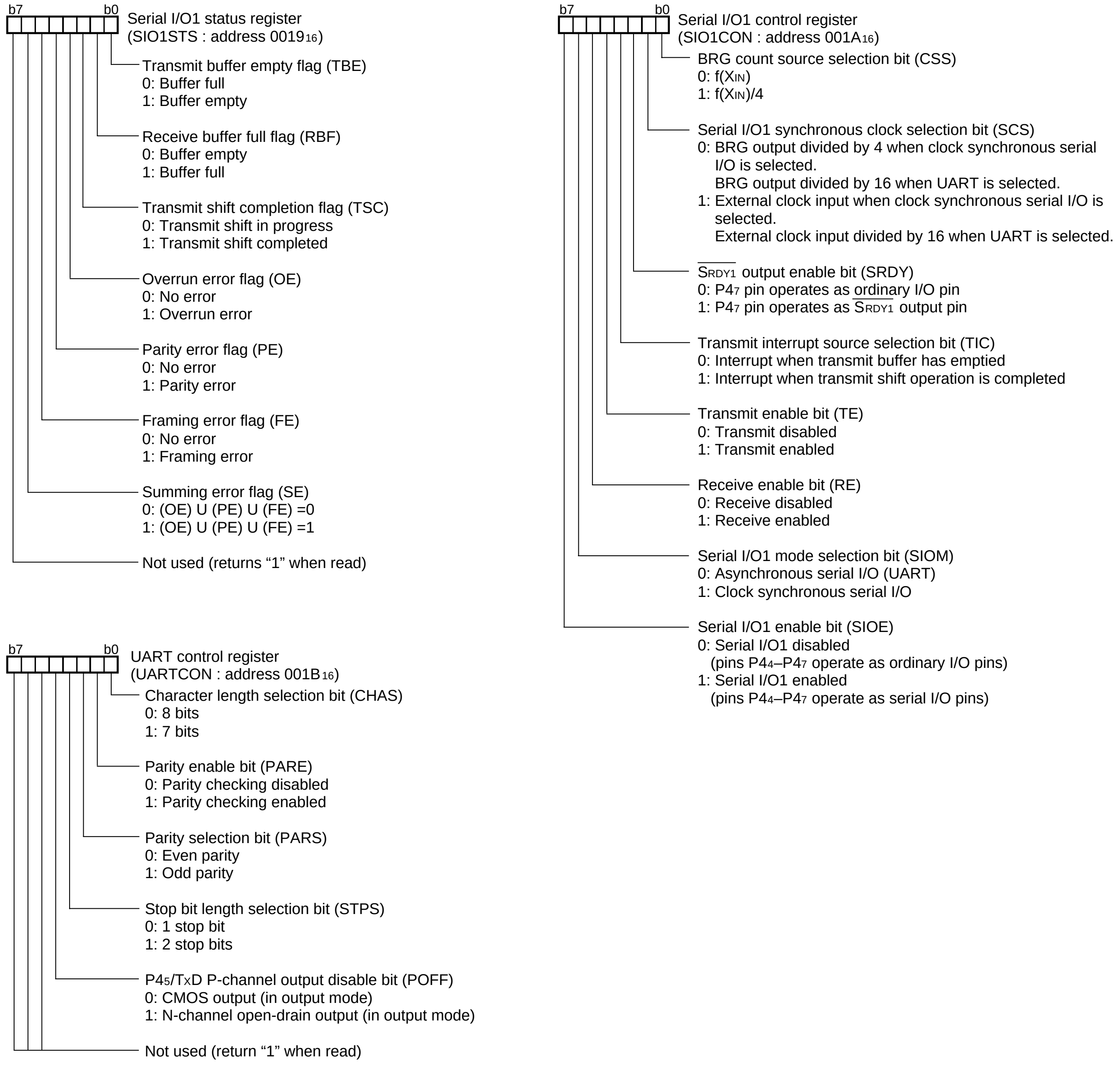
[UART Control Register (UARTCON)] $001B_{16}$

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 a data transfer. One bit in this register (bit 4) is always valid and sets the output structure of the $P4_5$/TXD pin and $P4_6$/SCLK1 pin.

[Baud Rate Generator (BRG)] $001C_{16}$

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.



Serial I/O1 status register
(SIO1STS : address 0019_{16})

Transmit buffer empty flag (TBE)
0: Buffer full
1: Buffer empty

Receive buffer full flag (RBF)
0: Buffer empty
1: Buffer full

Transmit shift completion flag (TSC)
0: Transmit shift in progress
1: Transmit shift completed

Overrun error flag (OE)
0: No error
1: Overrun error

Parity error flag (PE)
0: No error
1: Parity error

Framing error flag (FE)
0: No error
1: Framing error

Summing error flag (SE)
0: (OE) U (PE) U (FE) =0
1: (OE) U (PE) U (FE) =1

Not used (returns “1” when read)

Serial I/O1 control register
(SIO1CON : address $001A_{16}$)

BRG count source selection bit (CSS)
0: $f(X_{IN})$
1: $f(X_{IN})/4$

Serial I/O1 synchronous clock selection bit (SCS)
0: BRG output divided by 4 when clock synchronous serial I/O is selected.
BRG output divided by 16 when UART is selected.
1: External clock input when clock synchronous serial I/O is selected.
External clock input divided by 16 when UART is selected.

$\overline{S_{RDY1}}$ output enable bit (SRDY)
0: $P4_7$ pin operates as ordinary I/O pin
1: $P4_7$ pin operates as $\overline{S_{RDY1}}$ output pin

Transmit interrupt source selection bit (TIC)
0: Interrupt when transmit buffer has emptied
1: Interrupt when transmit shift operation is completed

Transmit enable bit (TE)
0: Transmit disabled
1: Transmit enabled

Receive enable bit (RE)
0: Receive disabled
1: Receive enabled

Serial I/O1 mode selection bit (SIOM)
0: Asynchronous serial I/O (UART)
1: Clock synchronous serial I/O

Serial I/O1 enable bit (SIOE)
0: Serial I/O1 disabled
(pins $P4_4$–$P4_7$ operate as ordinary I/O pins)
1: Serial I/O1 enabled
(pins $P4_4$–$P4_7$ operate as serial I/O pins)

UART control register
(UARTCON : address $001B_{16}$)

Character length selection bit (CHAS)
0: 8 bits
1: 7 bits

Parity enable bit (PARE)
0: Parity checking disabled
1: Parity checking enabled

Parity selection bit (PARS)
0: Even parity
1: Odd parity

Stop bit length selection bit (STPS)
0: 1 stop bit
1: 2 stop bits

$P4_5/T_XD$ P-channel output disable bit (POFF)
0: CMOS output (in output mode)
1: N-channel open-drain output (in output mode)

Not used (return “1” when read)

Fig. 32 Structure of serial I/O1 control register

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. When the internal clock is used, transfer is started by a write signal to the serial I/O2 register.

[Serial I/O2 Control Register (SIO2CON)] $001D_{16}$

The serial I/O2 control register contains 8 bits which control various serial I/O functions.

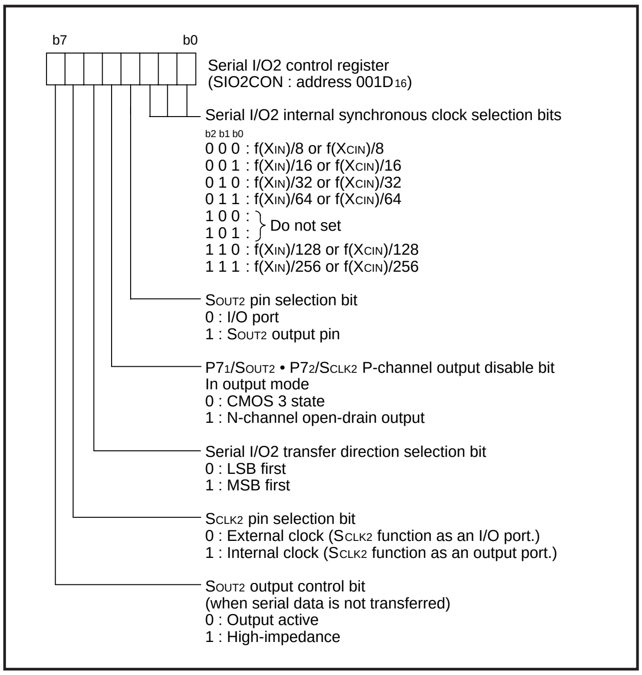

b7
b0
Serial I/O2 control register
(SIO2CON : address $001D_{16}$)
Serial I/O2 internal synchronous clock selection bits
b2 b1 b0
0 0 0 : $f(X_{IN})/8$ or $f(X_{CIN})/8$
0 0 1 : $f(X_{IN})/16$ or $f(X_{CIN})/16$
0 1 0 : $f(X_{IN})/32$ or $f(X_{CIN})/32$
0 1 1 : $f(X_{IN})/64$ or $f(X_{CIN})/64$
1 0 0 :
1 0 1 : Do not set
1 1 0 : $f(X_{IN})/128$ or $f(X_{CIN})/128$
1 1 1 : $f(X_{IN})/256$ or $f(X_{CIN})/256$
S_{OUT2} pin selection bit
0 : I/O port
1 : S_{OUT2} output pin
$P7_1/S_{OUT2}$ • $P7_2/S_{CLK2}$ P-channel output disable bit
In output mode
0 : CMOS 3 state
1 : N-channel open-drain output
Serial I/O2 transfer direction selection bit
0 : LSB first
1 : MSB first
S_{CLK2} pin selection bit
0 : External clock (S_{CLK2} function as an I/O port.)
1 : Internal clock (S_{CLK2} function as an output port.)
S_{OUT2} output control bit
(when serial data is not transferred)
0 : Output active
1 : High-impedance

Fig. 33 Structure of serial I/O2 control register

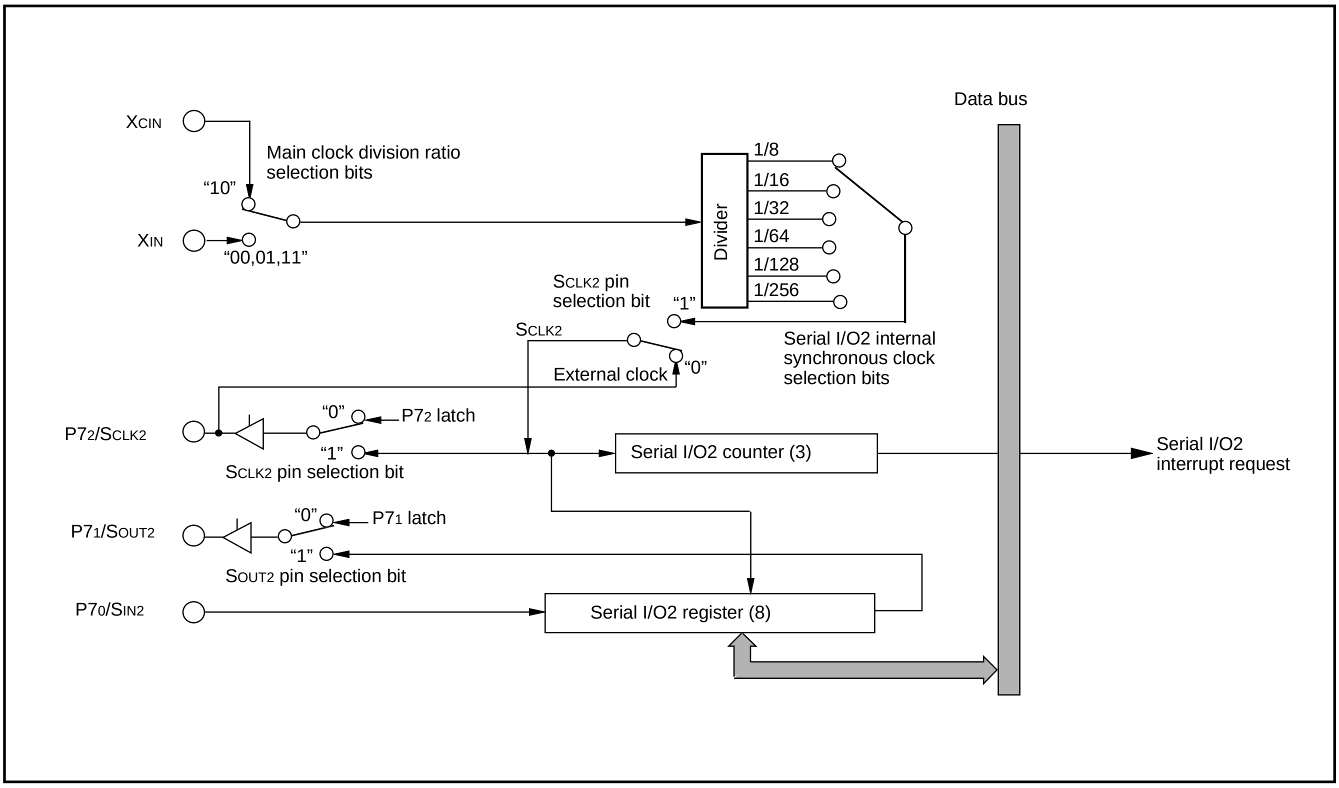


Fig. 34 Block diagram of serial I/O2

●Serial I/O2 operation

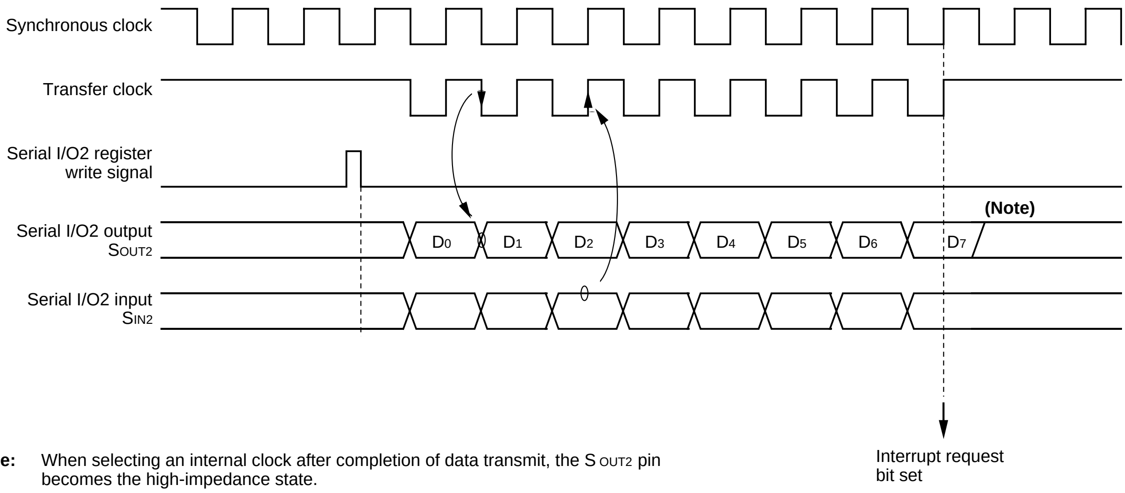
When writing to the serial I/O2 register (001F16), the serial I/O2 counter is set to “7”.

After the write is completed, data is output from the SOUT2 pin each time the transfer clock goes from “H” to “L”. In addition, each time the transfer clock goes from “L” to “H”, the contents of the serial I/O2 register are shifted by 1 bit data is simultaneously received from the SIN2 pin.

When selecting an internal clock as the transfer clock source, the serial I/O2 counter goes to “0” by counting the transfer clock 8 times, and the transfer clock stops at “H”, and the interrupt request bit is set to “1”. In addition, the SOUT2 pin becomes the high-impedance state after the completion of data transfer. (Bit 7 of the serial I/O2 control register does not go to “1” and only the SOUT2 pin becomes the high-impedance state.)

When selecting an external clock as the transfer clock source, the interrupt request bit is set when counting the transfer clock 8 times. However, the transfer clock does not stop, so that control the clock externally. The SOUT2 pin does not become the high-impedance state after completion of data transmit.

In order to set the SOUT2 pin to the high-impedance state when selecting an external clock, set “1” to bit 7 of the serial I/O2 control register after completion of data transmit. Also, make sure that SCLK2 is at “H” for this process. When the next data is transmitted (falling of transfer clock), bit 7 of the serial I/O2 control register goes to “0” and the SOUT2 pin goes to an active state.



Note: When selecting an internal clock after completion of data transmit, the S OUT2 pin becomes the high-impedance state.

Fig. 35 Serial I/O2 timing (LSB first)

Serial I/O3

Serial I/O3 has the following modes: 8-bit serial I/O, arbitrary bits from 1 to 256 serial I/O, up to 256-byte auto-transfer serial I/O. The 8-bit serial I/O transfers through serial I/O3 register (address 0013_{16}). The arbitrary bits and auto-transfer serial I/O modes transfer through the 256-byte serial I/O3 auto-transfer RAM (addresses 0200_{16} to $02FF_{16}$).

The P8₅/SRDY3, P8₆/SBUSY3, and P8₇/SSTB3 pins all have the handshake input/output signal function and can perform active logic high/low selection.

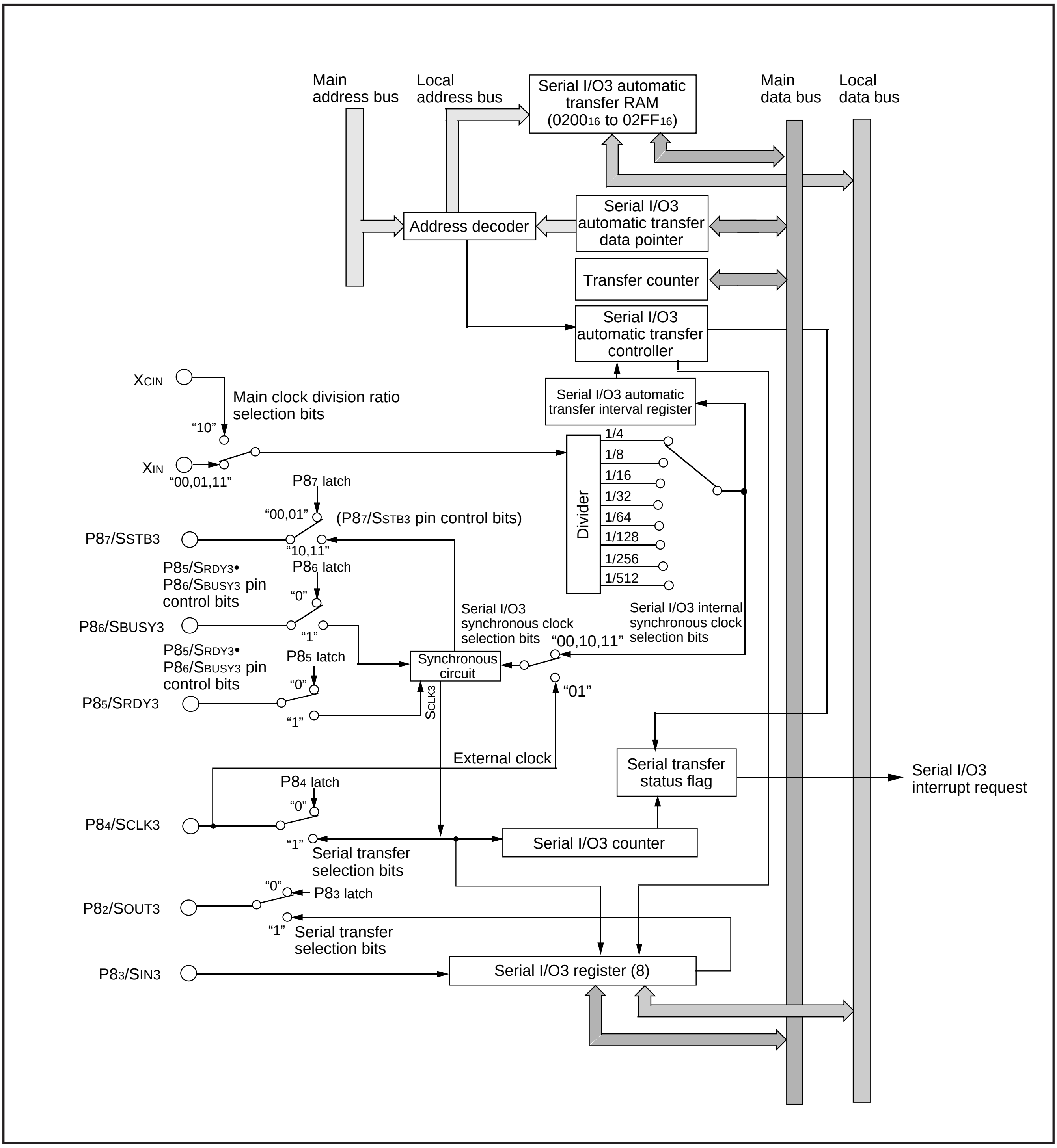
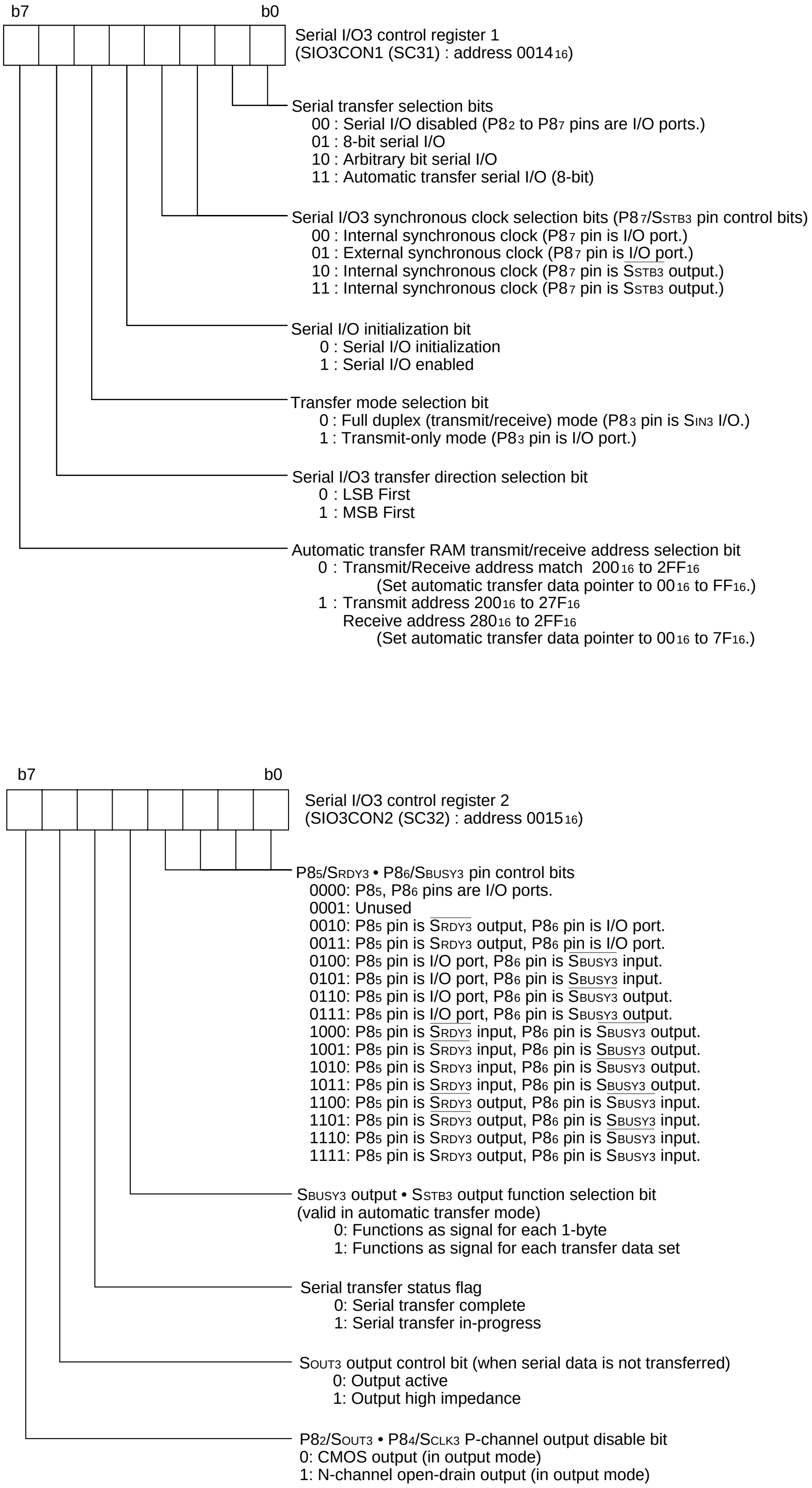

Main address bus
Local address bus
Serial I/O3 automatic transfer RAM (0200₁₆ to 02FF₁₆)
Main data bus
Local data bus
Address decoder
Serial I/O3 automatic transfer data pointer
Transfer counter
Serial I/O3 automatic transfer controller
XCIN
Main clock division ratio selection bits
"10"
Serial I/O3 automatic transfer interval register
XIN
"00,01,11"
P8₇ latch
"00,01"
(P8₇/SSTB3 pin control bits)
P8₇/SSTB3
"10,11"
Divider
1/4
1/8
1/16
1/32
1/64
1/128
1/256
1/512
P8₅/SRDY3• P8₆/SBUSY3 pin control bits
P8₆ latch
"0"
P8₆/SBUSY3
"1"
Serial I/O3 synchronous clock selection bits
"00,10,11"
Serial I/O3 internal synchronous clock selection bits
Synchronous circuit
P8₅/SRDY3• P8₆/SBUSY3 pin control bits
P8₅ latch
"0"
P8₅/SRDY3
"1"
SCLK3
"01"
External clock
Serial transfer status flag
Serial I/O3 interrupt request
P8₄ latch
"0"
P8₄/SCLK3
"1"
Serial transfer selection bits
Serial I/O3 counter
"0"
P8₃ latch
P8₂/SOUT3
"1"
Serial transfer selection bits
P8₃/SIN3
Serial I/O3 register (8)

Fig. 36 Block diagram of serial I/O3



Serial I/O3 control register 1
(SIO3CON1 (SC31) : address 0014$_{16}$)

Serial transfer selection bits
00 : Serial I/O disabled (P8$_2$ to P8$_7$ pins are I/O ports.)
01 : 8-bit serial I/O
10 : Arbitrary bit serial I/O
11 : Automatic transfer serial I/O (8-bit)

Serial I/O3 synchronous clock selection bits (P8$_7$/SSTB3 pin control bits)
00 : Internal synchronous clock (P8$_7$ pin is I/O port.)
01 : External synchronous clock (P8$_7$ pin is I/O port.)
10 : Internal synchronous clock (P8$_7$ pin is SSTB3 output.)
11 : Internal synchronous clock (P8$_7$ pin is SSTB3 output.)

Serial I/O initialization bit
0 : Serial I/O initialization
1 : Serial I/O enabled

Transfer mode selection bit
0 : Full duplex (transmit/receive) mode (P8$_3$ pin is SIN3 I/O.)
1 : Transmit-only mode (P8$_3$ pin is I/O port.)

Serial I/O3 transfer direction selection bit
0 : LSB First
1 : MSB First

Automatic transfer RAM transmit/receive address selection bit
0 : Transmit/Receive address match 200$_{16}$ to 2FF$_{16}$
(Set automatic transfer data pointer to 00$_{16}$ to FF$_{16}$.)
1 : Transmit address 200$_{16}$ to 27F$_{16}$
Receive address 280$_{16}$ to 2FF$_{16}$
(Set automatic transfer data pointer to 00$_{16}$ to 7F$_{16}$.)

Serial I/O3 control register 2
(SIO3CON2 (SC32) : address 0015$_{16}$)

P8$_5$/SRDY3 • P8$_6$/SBUSY3 pin control bits
0000: P8$_5$, P8$_6$ pins are I/O ports.
0001: Unused
0010: P8$_5$ pin is SRDY3 output, P8$_6$ pin is I/O port.
0011: P8$_5$ pin is SRDY3 output, P8$_6$ pin is I/O port.
0100: P8$_5$ pin is I/O port, P8$_6$ pin is SBUSY3 input.
0101: P8$_5$ pin is I/O port, P8$_6$ pin is SBUSY3 input.
0110: P8$_5$ pin is I/O port, P8$_6$ pin is SBUSY3 output.
0111: P8$_5$ pin is I/O port, P8$_6$ pin is SBUSY3 output.
1000: P8$_5$ pin is SRDY3 input, P8$_6$ pin is SBUSY3 output.
1001: P8$_5$ pin is SRDY3 input, P8$_6$ pin is SBUSY3 output.
1010: P8$_5$ pin is SRDY3 input, P8$_6$ pin is SBUSY3 output.
1011: P8$_5$ pin is SRDY3 input, P8$_6$ pin is SBUSY3 output.
1100: P8$_5$ pin is SRDY3 output, P8$_6$ pin is SBUSY3 input.
1101: P8$_5$ pin is SRDY3 output, P8$_6$ pin is SBUSY3 input.
1110: P8$_5$ pin is SRDY3 output, P8$_6$ pin is SBUSY3 input.
1111: P8$_5$ pin is SRDY3 output, P8$_6$ pin is SBUSY3 input.

SBUSY3 output • SSTB3 output function selection bit
(valid in automatic transfer mode)
0: Functions as signal for each 1-byte
1: Functions as signal for each transfer data set

Serial transfer status flag
0: Serial transfer complete
1: Serial transfer in-progress

SOUT3 output control bit (when serial data is not transferred)
0: Output active
1: Output high impedance

P8$_2$/SOUT3 • P8$_4$/SCLK3 P-channel output disable bit
0: CMOS output (in output mode)
1: N-channel open-drain output (in output mode)

Fig. 37 Structure of serial I/O3 control registers 1 and 2

●**Serial I/O3 operation**

An internal or external synchronous clock can be selected as the serial transfer synchronous clock by the serial I/O3 synchronous clock selection bits of the serial I/O3 control register 1.

Since the internal synchronous clock has its own built-in divider, 8 types of clocks can be selected by the serial I/O3 internal synchronous clock selection bits of the serial I/O3 control register 3.

Either I/O port or handshake I/O signal function can be selected for the P8$_5$/SRDY3, P8$_6$/SBUSY3, and P8$_7$/SSTB3 pins by the serial I/O3 synchronous clock selection bits (P8$_7$/SSTB3 pin control bits) of the serial I/O3 control register 1 or the P8$_5$/SRDY3•P8$_6$/SBUSY3 pin control bits of the serial I/O3 control register 2.

CMOS output or N-channel open-drain output can be selected for the SCLK3 and SOUT3 output pins by the P8$_2$/SOUT3 • P8$_4$/SCLK3 P-channel output disable bit of the serial I/O3 control register 2.

The SOUT3 output control bit of the serial I/O3 control register 2 can be used to select the status of the SOUT3 pin when serial data is not transferred; either output active or high-impedance. However, when selecting an external synchronous clock, the SOUT3 pin can go to the high-impedance status by setting the SOUT3 output control bit to “1” when SCLK3 input is at “H” after transfer completion. When the next serial transfer begins and SCLK3 goes to “L”, the SOUT3 output control bit is automatically reset to “0” and goes to an output active status.

Regardless of selecting an internal or external synchronous clock, the serial transfer has both a full duplex mode as well as a transmit-only mode. These modes are set by the transfer mode selection bit of serial I/O3 control register 1.

LSB first or MSB first can be selected for the input/output order of the serial transfer bit string by the serial I/O3 transfer direction selection bit of serial I/O3 control register 1.

In order to use serial I/O3, the following process must be followed after all of the above set have been completed: First, select any one of 8-bit serial I/O, arbitrary bit serial I/O, or automaric transfer serial I/O by setting the serial transfer selection bits of the serial I/O3 control register 1. Then, enable the serial I/O by setting the serial I/O initialization bit of the serial I/O3 control register 1 to “1”. Whether using an internal or external synchronous clock, set the serial I/O initialization bit to “0” when terminating a serial transfer during the transmission.

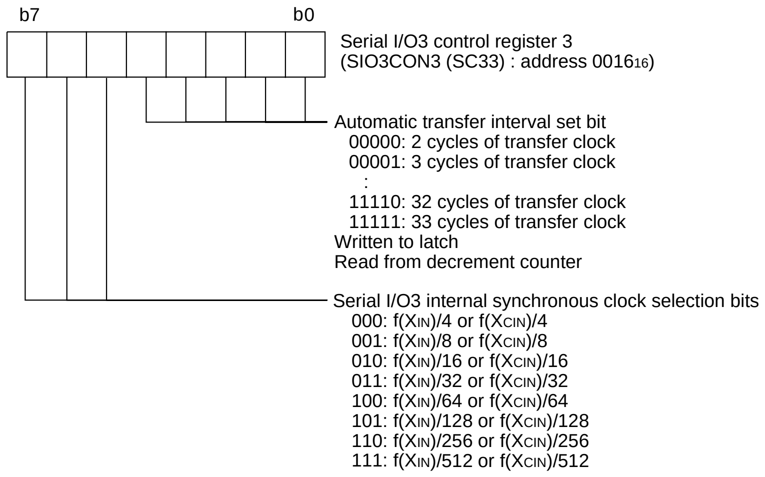

b7
b0
Serial I/O3 control register 3
(SIO3CON3 (SC33) : address 0016$_{16}$)
Automatic transfer interval set bit
00000: 2 cycles of transfer clock
00001: 3 cycles of transfer clock
:
11110: 32 cycles of transfer clock
11111: 33 cycles of transfer clock
Written to latch
Read from decrement counter
Serial I/O3 internal synchronous clock selection bits
000: f(XIN)/4 or f(XCIN)/4
001: f(XIN)/8 or f(XCIN)/8
010: f(XIN)/16 or f(XCIN)/16
011: f(XIN)/32 or f(XCIN)/32
100: f(XIN)/64 or f(XCIN)/64
101: f(XIN)/128 or f(XCIN)/128
110: f(XIN)/256 or f(XCIN)/256
111: f(XIN)/512 or f(XCIN)/512

Fig. 38 Structure of serial I/O3 control register 3

(1) 8-bit serial I/O mode

Address 0013_{16} is the serial I/O3 register. When selecting an internal synchronous clock, serial transfer of the 8-bit serial I/O starts by the write signal to the serial I/O3 register (address 0013_{16}).

The serial transfer status flag of the serial I/O3 control register 2 indicates the serial I/O3 register status. The flag is set to “1” by a serial I/O3 register write, which triggers a transfer start. After the 8-bit transfer is completed, the flag is reset to “0” and a serial I/O3 interrupt request occurs simultaneously.

When an external synchronous clock is selected, the contents of the serial I/O3 register are continually shifted while the transfer clock inputs to SCLK3. In this case, control the clock externally.

(2) Automatic transfer serial I/O mode

Since read and write to the serial I/O3 register are controlled by the serial I/O3 automatic transfer controller, address 0013_{16} functions as the transfer counter (in byte units).

In order to make a serial transfer through the serial I/O3 automatic transfer RAM (addresses 0200_{16} to $02FF_{16}$), it is necessary to set the serial I/O3 automatic transfer data pointer before transferring data. The automatic transfer data pointer set bits indicate the low-order 8 bits of the start data stored address. The automatic transfer RAM transmit/receive address select bit can divide the 256-byte serial I/O3 automatic transfer RAM into two areas: 128-byte transmit data area and 128-byte receive data area.

When an internal synchronous clock is selected and any of the following conditions apply, the transfer interval between each 1-byte data can be set by the automatic transfer interval set bits of the serial I/O3 control register 3:

1. The handshake signal is not used.
2. The handshake signal’s SRDY3 output, SBUSY3 output, and SSTB3 output are used independently.
3. The handshake signal’s output is used in groups: SRDY3/SSTB3 output or SBUSY3/SSTB3.

There are 32 values among 2 and 33 cycles of the transfer clock.

When the automatic transfer interval setting is valid and SBUSY3 output is used, and the SBUSY3 and SSTB3 output function as signal for each transfer data set by the SBUSY3 output•SSTB3 output function selection bit, there is the transfer interval before the first data is transmitted/received, as well as after the last data is transmitted/received. When using SSTB3 output, regardless of the contents of the SBUSY3 output • SSTB3 output function selection bit, this transfer interval become 2 cycles longer than the value set for each 1-byte data. In addition, when using the combined output of SBUSY3 and SSTB3 as the signal for each transfer data set, the transfer interval after completion of transmission/receipt of the last data become 2 cycles longer than the set value.

When selecting an external synchronous clock, the automatic transfer interval cannot be set.

After all of the above bit settings have been completed, and an internal synchronous clock has been selected, serial automatic transfer starts when the value of the number of transfer bytes, decremented by 1, is written to the transfer counter (address 0013_{16}). When an external synchronous clock is selected, write the value of the transfer bytes, decremented by 1, to the transfer counter, and input the transfer clock to SCLK3 after 5 or more cycles of internal clock ϕ.

Set the transfer interval of each 1-byte data transmission to 5 or more cycles of the internal clock ϕ after the rising edge of the last bit of a 1-byte data.

Regardless of internal or external synchronous clock, the automatic transfer data pointer and transfer counter are both decremented after receipt of each 1-byte data is completed and it is written to the automatic transfer RAM. The serial transfer status flag is set to “1” by writing to the transfer counter which triggers the start of transmission. After the last data is written to the automatic transfer RAM, the serial transfer status flag is set to “0” and a serial I/O3 interrupt request occurs simultaneously.

The write values of the automatic transfer data pointer set bits and the automatic transfer interval set bits are kept in the latch. As a transfer counter write occurs, each value is transferred to its corresponding decrement counter.

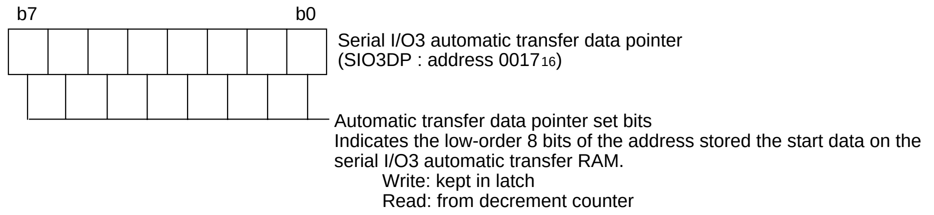

b7
b0
Serial I/O3 automatic transfer data pointer
(SIO3DP : address 0017_{16})
Automatic transfer data pointer set bits
Indicates the low-order 8 bits of the address stored the start data on the serial I/O3 automatic transfer RAM.
Write: kept in latch
Read: from decrement counter

Fig. 39 Structure of serial I/O3 automatic transfer data pointer

(3) Arbitrary bit serial I/O mode

Since read and write of the serial I/O3 register are controlled by the serial I/O3 automatic transfer controller, address 0013_{16} functions as the transfer counter (in byte units).

After the serial I/O3 automatic transfer data pointer and automatic transfer interval set bits have been set, and an internal synchronous clock selected, serial automatic transfer starts when the value of the number of transfer bits decremented by 1 is written to the transfer counter (address 0013_{16}), just as in the automatic transfer serial I/O. When selecting an external synchronous clock, write the value of the transfer bits decremented by 1 to the transfer counter, then input the transfer clock to SCLK3 after 5 or more cycles of internal clock ϕ. The transfer interval after each 8-bit data transfer must be 5 or more cycles of internal clock ϕ after the rising edge of the last bit of the 8-bit data.

When selecting an internal synchronous clock, the automatic transfer interval can be specified regardless of the contents of the selected handshake signal.

In this case, when the automatic transfer interval setting is valid and SBUSY3 output is used there are the transfer interval before the first data is transmitted/received, as well as after the last data is transmitted/received just as in the automatic transfer serial I/O mode. When using SSTB3 output, this transfer interval become 2 cycles longer than the value set for each 8-bit data. In addition, when using the combined output of SBUSY3 and SSTB3, the transfer interval after completion of transmission/receipt of the last data become 2 cycles longer than the set value.

When selecting an external synchronous clock, the automatic transfer interval cannot be specified.

Regardless of internal or external synchronous clock, the automatic transfer data pointer is decremented after each 8-bit data is received and then written to the auto-transfer RAM. The transfer counter is decremented with the transfer clock. The serial transfer status flag is set to “1” by writing to the transfer counter which triggers the start of transmission. After the last data is written to the automatic transfer RAM, the serial transfer status flag is set to “0” and a serial I/O3 interrupt request occurs simultaneously.

The write values of the automatic transfer data pointer set bits and the automatic transfer interval set bits are kept in the latch. As a transfer counter write occurs, each value is transferred to its corresponding decrement counter.

If the last data does not fill 8 bits, the receive data stored in the serial I/O3 automatic transfer RAM become the closest MSB odd bit if the transfer direction select bit is set to LSB first, or the closest LSB odd bit if the transfer direction select bit is set to MSB first.

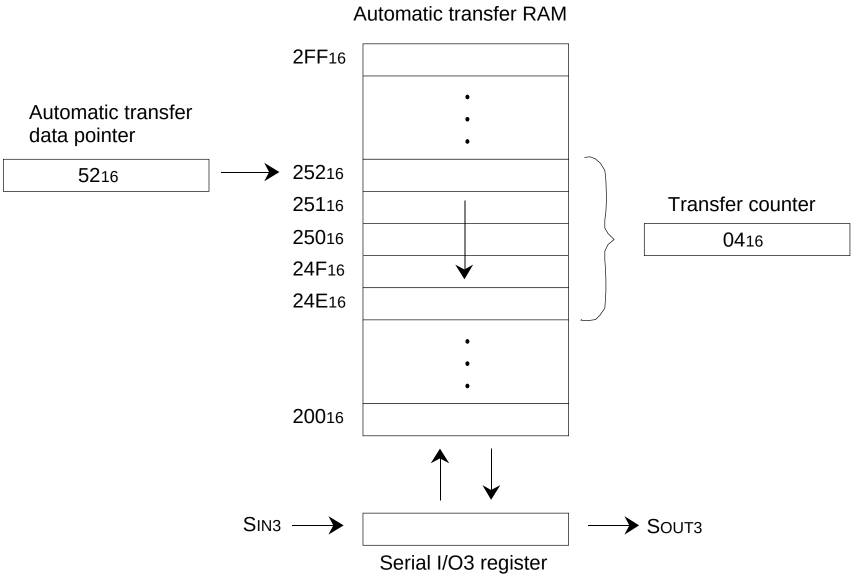
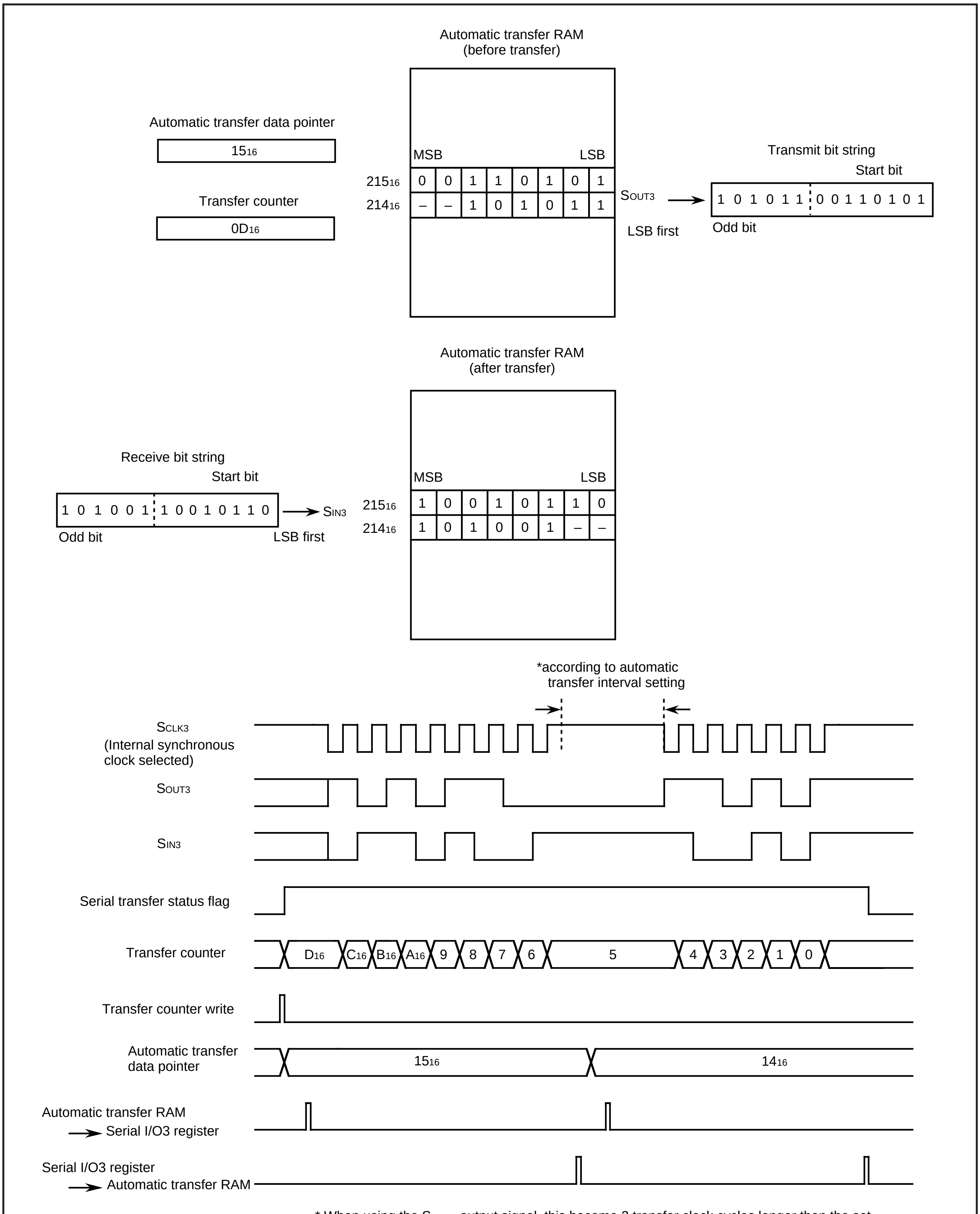


Fig. 40 Automatic transfer serial I/O operation



* When using the S_{STB3} output signal, this become 2 transfer clock cycles longer than the set interval.

Fig. 41 Arbitrary bit serial I/O operation

Handshake Signal

● **SSTB3 output signal**

The SSTB3 output is a signal to inform an end of transmission/reception to the serial transfer destination . The SSTB3 output signal can be used only when the internal synchronous clock is selected.

In the initial status, that is, in the status in which the serial I/O initialization bit (b4) is reset to “0”, the SSTB3 output goes to “L”, and the $\overline{\text{SSTB3}}$ output goes to “H”.

At the end of transmit/receive operation, when the data of the serial I/O3 register is all output from SOUT3, pulses which are the SSTB3 output of “H” and the $\overline{\text{SSTB3}}$ output of “L” are output in the period of 1 cycle of the transfer clock. After that, each pulse is returned to the initial status in which SSTB3 output goes to “L” and the $\overline{\text{SSTB3}}$ output goes to “H”.

Furthermore, after 1 cycle, the serial transfer status flag (b5) is reset to “0”.

In the automatic transfer serial I/O mode, whether making the SSTB3 output active at an end of each 1-byte data or after completion of transfer of all data can be selected by the SBUSY3 output • SSTB3 output function selection bit (b4 of address 0015_{16}) of serial I/O3 control register 2.

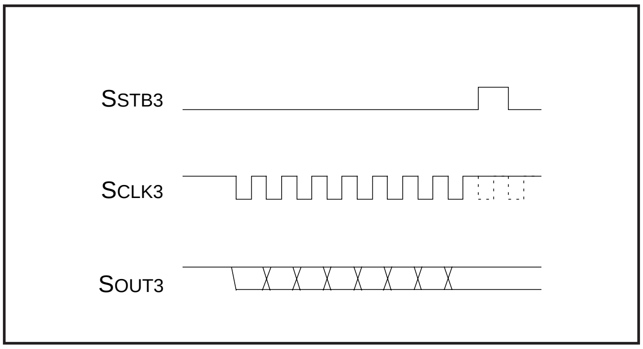


Fig. 42 SSTB3 output operation

● **SBUSY3 input signal**

The SBUSY3 input is a signal which receives a request for a stop of transmission/reception from the serial transfer destination.

When the internal synchronous clock is selected, input an “H” level signal into the SBUSY3 input and an “L” level signal into the $\overline{\text{SBUSY3}}$ input in the initial status in which transfer is stopped.

When starting a transmit/receive operation, input an “L” level signal into the SBUSY3 input and an “H” level signal into the $\overline{\text{SBUSY3}}$ input in the period of 1.5 cycles or more of the transfer clock. Then, transfer clocks are output from the SCLK3 output.

When an “H” level signal is input into the SBUSY3 input and an “L” level signal into the $\overline{\text{SBUSY3}}$ input after a transmit/receive operation is started, this transmit/receive operation are not stopped immediately and the transfer clocks from the SCLK3 output are not stopped until the specified number of bits is transmitted and received.

The handshake unit of the 8-bit serial I/O is 8 bits and that of the arbitrary bit serial I/O is the bit number adding “1” to the set value to the transfer counter, and that of the automatic transfer serial I/O is 8 bits.

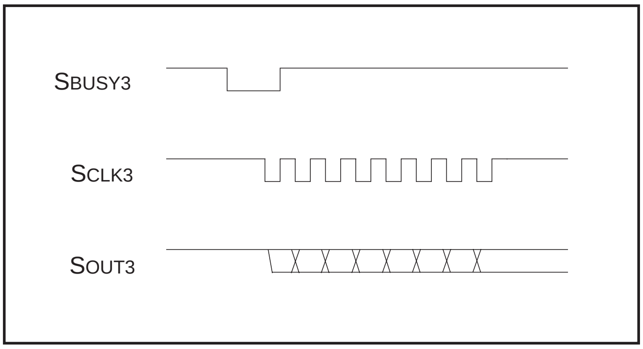


Fig. 43 SBUSY3 input operation (internal synchronous clock)

When the external synchronous clock is selected, input an “H” level signal into the SBUSY3 input and an “L” level signal into the $\overline{\text{SBUSY3}}$ input in the initial status in which transfer is stopped. At this time, the transfer clocks to be input in SCLK3 become invalid.

During serial transfer, the transfer clocks to be input in SCLK3 become valid, enabling a transmit/receive operation, while an “L” level signal is input into the SBUSY3 input and an “H” level signal is input into the $\overline{\text{SBUSY3}}$ input.

When changing the input values in to the SBUSY3 input and the $\overline{\text{SBUSY3}}$ input in these operations, change them while the SCLK3 input is in a high state.

When the high impedance of the SOUT3 output is selected by the SOUT3 output control bit (b6), the SOUT3 output becomes active, enabling serial transfer by inputting a transfer clock to SCLK3, while an “L” level signal is input into the SBUSY3 input and an “H” level signal is input into the $\overline{\text{SBUSY3}}$ input.

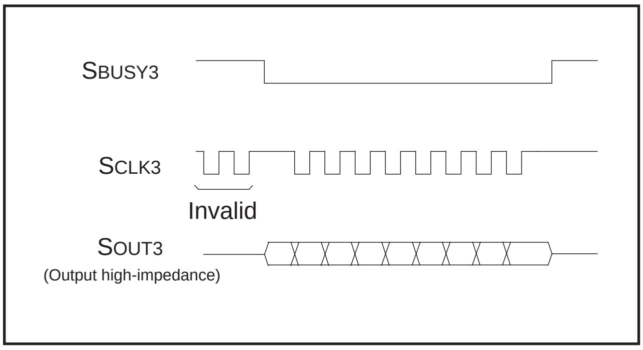


Fig. 44 SBUSY3 input operation (external synchronous clock)

● **SBUSY3 output signal**

The SBUSY3 output is a signal which requests a stop of transmission/reception to the serial transfer destination. In the automatic transfer serial I/O mode, regardless of the internal or external synchronous clock, whether making the SBUSY3 output active at transfer of each 1-byte data or during transfer of all data can be selected by the SBUSY3 output • SSTB3 output function selection bit (b4).

In the initial status, that is, the status in which the serial I/O initialization bit (b4) is reset to “0”, the SBUSY3 output goes to “H” and the $\overline{\text{SBUSY3}}$ output goes to “L”.

When the internal synchronous clock is selected, in the 8-bit serial I/O mode and the automatic transfer serial I/O mode (SBUSY3 output function outputs in 1-byte units), the SBUSY3 output goes to “L” and the $\overline{\text{SBUSY3}}$ output goes to “H” before 0.5 cycle (transfer clock) of the timing at which the transfer clock from the SCLK3 output goes to “L” at a start of transmit/receive operation.

In the automatic transfer serial I/O mode (the SBUSY3 output function outputs all transfer data), the SBUSY3 output goes to “L” and the $\overline{\text{SBUSY3}}$ output goes to “H” when the first transmit data is written into the serial I/O3 register (address 0013₁₆).

When the external synchronous clock is selected, the SBUSY3 output goes to “L” and the $\overline{\text{SBUSY3}}$ output goes to “H” when transmit data is written into the serial I/O3 register to start a transmit operation, regardless of the serial I/O transfer mode.

At termination of transmit/receive operation, the SBUSY3 output returns to “H” and the $\overline{\text{SBUSY3}}$ output returns to “L”, the initial status, when the serial transfer status flag is set to “0”, regardless of selecting the internal or external synchronous clock.

Furthermore, in the automatic transfer serial I/O mode (SBUSY3 output function outputs in 1-byte units), the SBUSY3 output goes to “H” and the $\overline{\text{SBUSY3}}$ output goes to “L” each time 1-byte of receive data is written into the automatic transfer RAM.

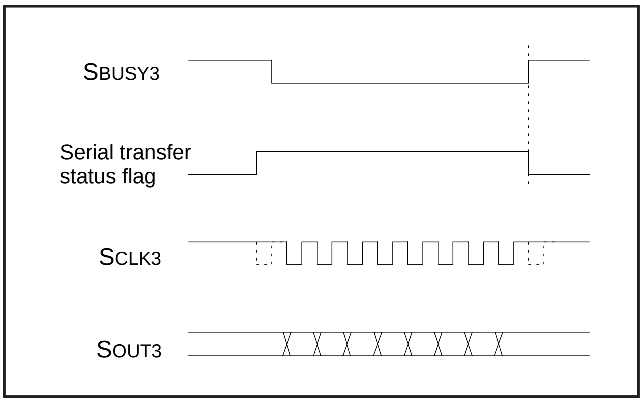


Fig. 45 SBUSY3 output operation (internal synchronous clock, 8-bit serial I/O)

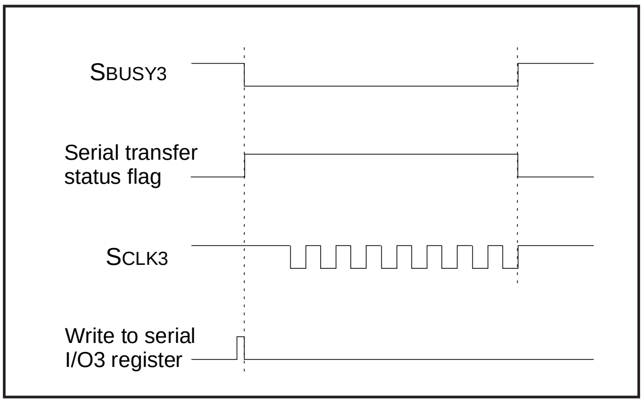


Fig. 46 SBUSY3 output operation (external synchronous clock, 8-bit serial I/O)

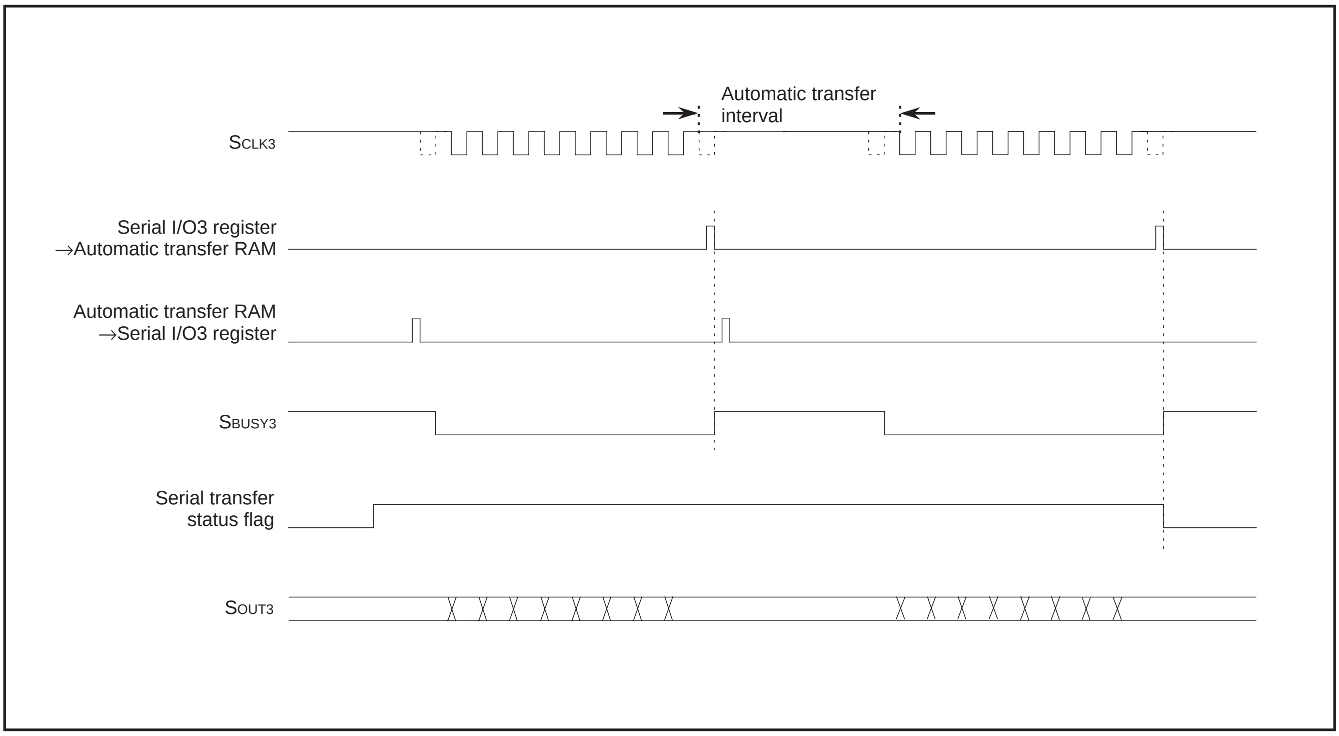

Automatic transfer interval
SCLK3
Serial I/O3 register →Automatic transfer RAM
Automatic transfer RAM →Serial I/O3 register
SBUSY3
Serial transfer status flag
SOUT3

Fig. 47 SBUSY3 output operation in automatic transfer serial I/O mode (internal synchronous clock, SBUSY3 output function outputs each 1-byte)

● **SRDY3 output signal**

The SRDY3 output is a transmit/receive enable signal which informs the serial transfer destination that transmit/receive is ready. In the initial status, that is, when the serial I/O initialization bit (b4) is reset to “0”, the SRDY3 output goes to “L” and the $\overline{\text{SRDY3}}$ output goes to “H”. After transmitted data is stored in the serial I/O3 register (address 0013_{16}) and a transmit/receive operation becomes ready, the SRDY3 output goes to “H” and the $\overline{\text{SRDY3}}$ output goes to “L”. When a transmit/receive operation is started and the transfer clock goes to “L”, the SRDY3 output goes to “L” and the $\overline{\text{SRDY3}}$ output goes to “H”.

● **SRDY3 input signal**

The SRDY3 input signal becomes valid only when the SRDY3 input and the SBUSY3 output are used. The SRDY3 input is a signal for receiving a transmit/receive ready completion signal from the serial transfer destination.

When the internal synchronous clock is selected, input a low level signal into the SRDY3 input and a high level signal into the $\overline{\text{SRDY3}}$ input in the initial status in which the transfer is stopped.

When an “H” level signal is input into the SRDY3 input and an “L” level signal is input into the $\overline{\text{SRDY3}}$ input for a period of 1.5 cycles or more of transfer clock, transfer clocks are output from the SCLK3 output and a transmit/receive operation is started.

After the transmit/receive operation is started and an “L” level signal is input into the SRDY3 input and an “H” level signal into the $\overline{\text{SRDY3}}$ input, this operation cannot be immediately stopped.

After the specified number of bits are transmitted and received, the transfer clocks from the SCLK3 output is stopped. The handshake unit of the 8-bit serial I/O and that of the automatic transfer serial I/O are of 8 bits. That of the arbitrary bit serial I/O is the bit number adding “1” to the set value to the transfer counter.

When the external synchronous clock is selected, the SRDY3 input becomes one of the triggers to output the SBUSY3 signal.

To start a transmit/receive operation (SBUSY3 output to “L”, $\overline{\text{SBUSY3}}$ output to “H”), input an “H” level signal into the SRDY3 input and an “L” level signal into the $\overline{\text{SRDY3}}$ input, and also write transmit data into the serial I/O3 register.

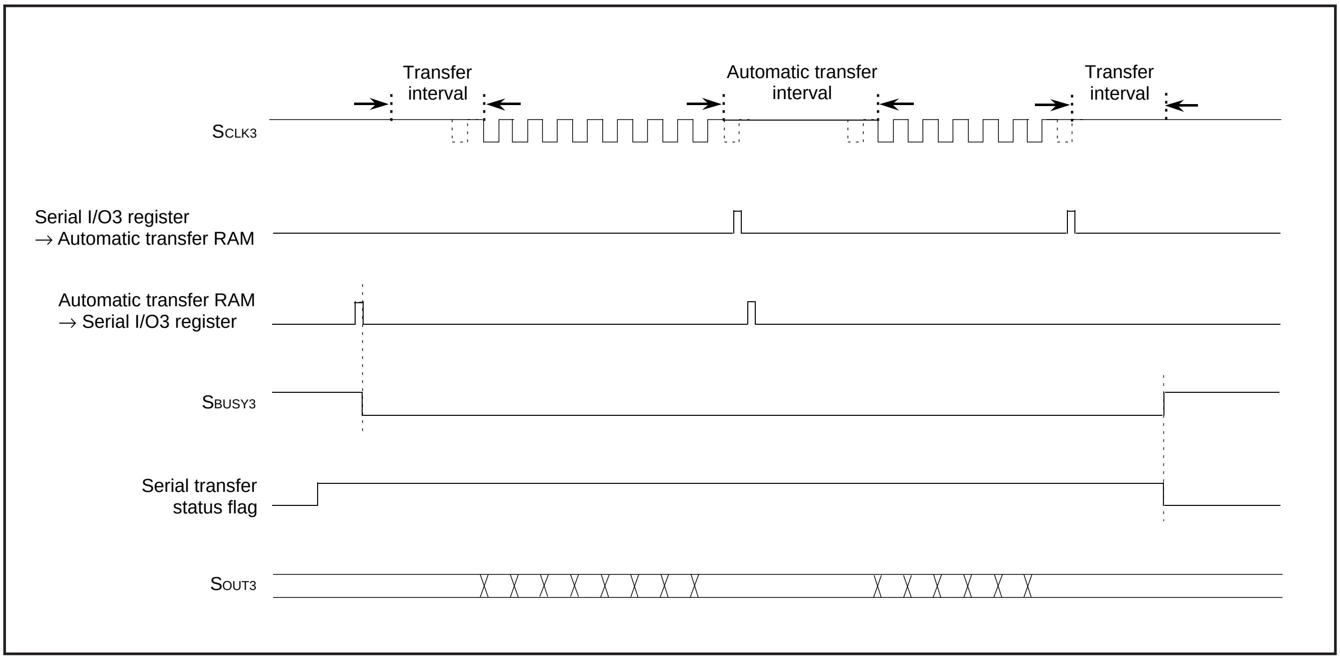

Transfer interval
Automatic transfer interval
Transfer interval
SCLK3
Serial I/O3 register → Automatic transfer RAM
Automatic transfer RAM → Serial I/O3 register
SBUSY3
Serial transfer status flag
SOUT3

Fig. 48 SBUSY3 output operation in arbitrary bit serial I/O mode (internal synchronous clock)

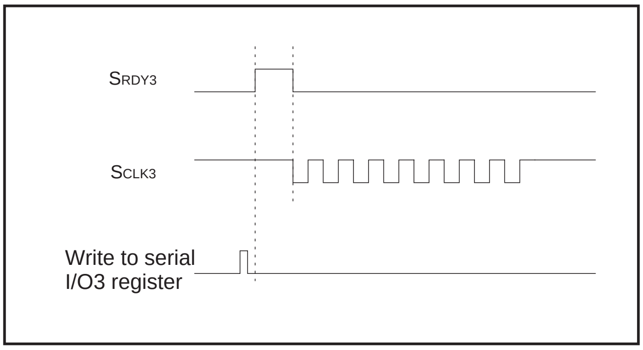


Fig. 49 SRDY3 output operation

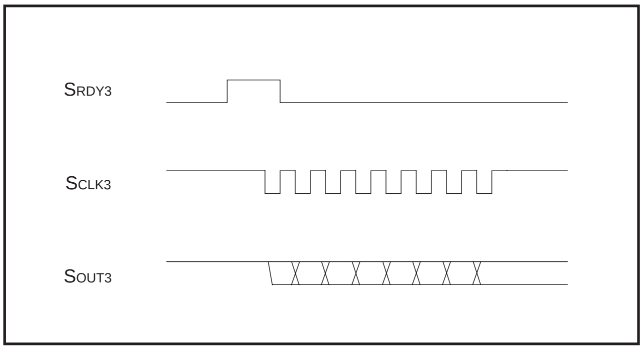


Fig. 50 SRDY3 input operation (internal synchronous clock)

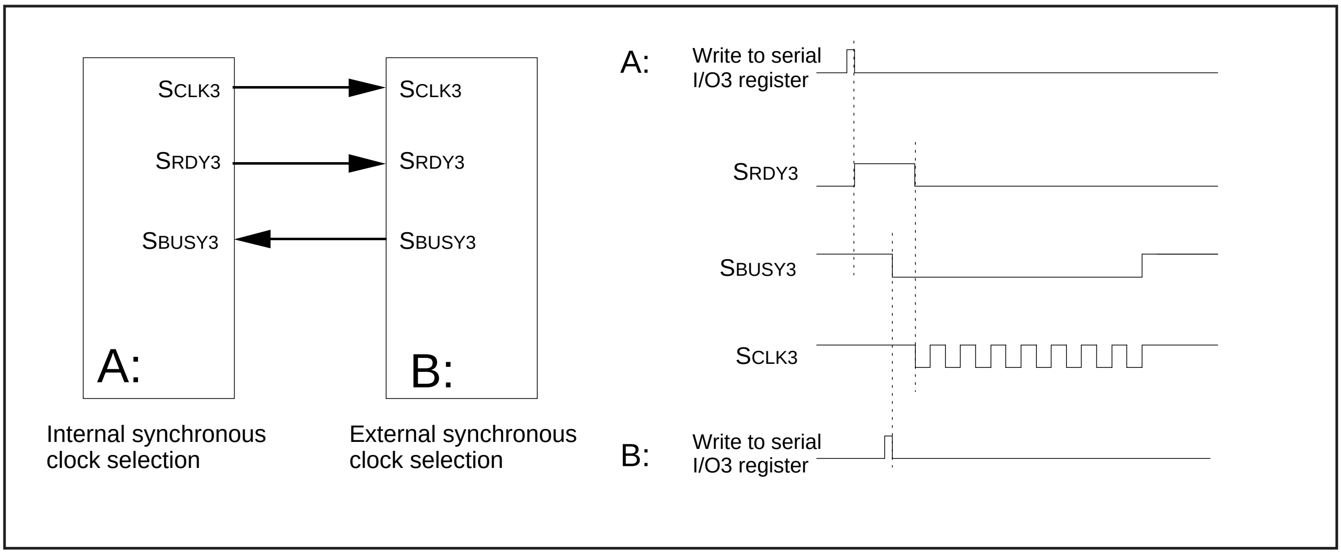

SCLK3
SRDY3
SBUSY3
A:
SCLK3
SRDY3
SBUSY3
B:
Internal synchronous clock selection
External synchronous clock selection
A:
Write to serial I/O3 register
SRDY3
SBUSY3
SCLK3
B:
Write to serial I/O3 register

Fig. 51 Handshake operation at serial I/O3 mutual connecting (1)

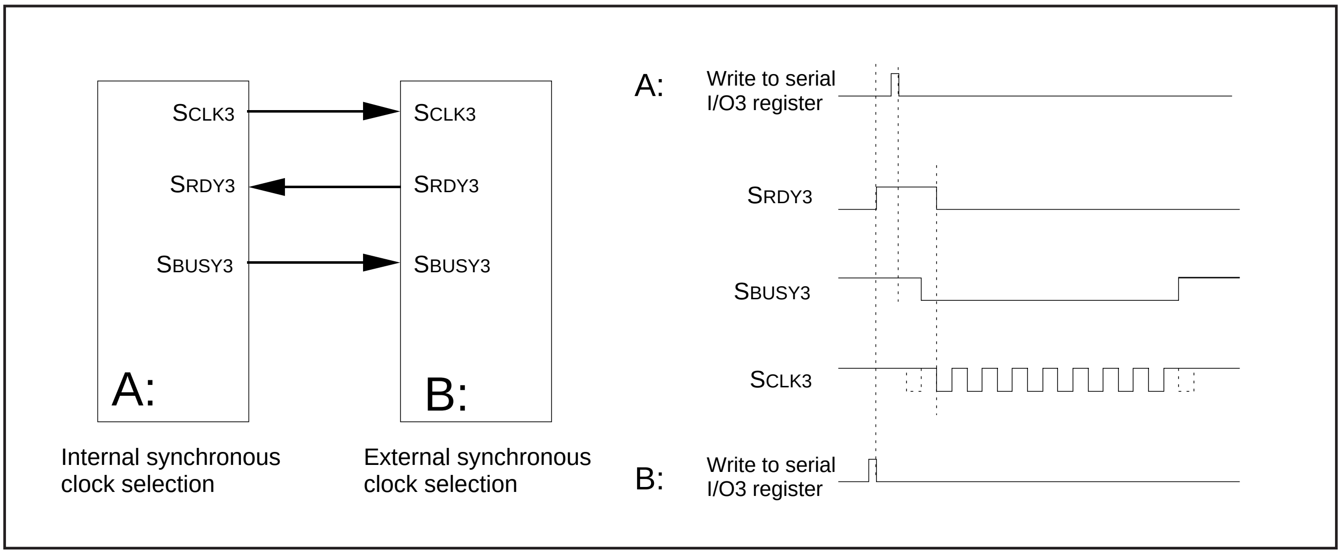

SCLK3
SRDY3
SBUSY3
A:
SCLK3
SRDY3
SBUSY3
B:
Internal synchronous clock selection
External synchronous clock selection
A:
Write to serial I/O3 register
SRDY3
SBUSY3
SCLK3
B:
Write to serial I/O3 register

Fig. 52 Handshake operation at serial I/O3 mutual connecting (2)

DATA LINK LAYER COMMUNICATION CONTROL CIRCUIT

The 3874 Group has a built-in data link layer communication control circuit.

This data link layer communication control circuit is applicable for multi-master serial bus communication control used only with data lines through an external driver/receiver.

The data link layer communication control circuit consists of following.

- Communication mode register (address 002A$_{16}$)
- Transmit control register (address 002B$_{16}$)
- Transmit status register (address 002C$_{16}$)
- Receive control register (address 002D$_{16}$)
- Receive status register (address 002E$_{16}$)
- Bus interrupt factor discrimination control register (address 002F$_{16}$)
- Control field selection register (address 0030$_{16}$)
- Control field register (address 0031$_{16}$)
- Transmit/Receive FIFO (address 0032$_{16}$)

This function is realized by hardware and firmware so that communication protocol can be partially modified according to the user's specification.

The following are the standard communication rate and functions which the data link layer communication control circuit can perform.

- Communication rate: Approx. 40 kbps
  The communication rate depends on frame or bit protocol.
- Synchronous method: Half-duplex asynchronous
- Modification method: PWM method, NRZ, etc.
- Communication functions:
  ①Bus arbitration
  (CSMA/CD method, etc.)
  ②Error detection
  (parity, acknowledge, CRC, etc.)
  ③Frame, data retry

The transmission signal is output from the $\overline{\text{BUS}_{\text{OUT}}}$ pin and input to the $\overline{\text{BUS}_{\text{IN}}}$ pin.

Detailed specifications for communication protocol, bit assignment, function, etc. of each register are defined according to each communication protocol specification confirmation.

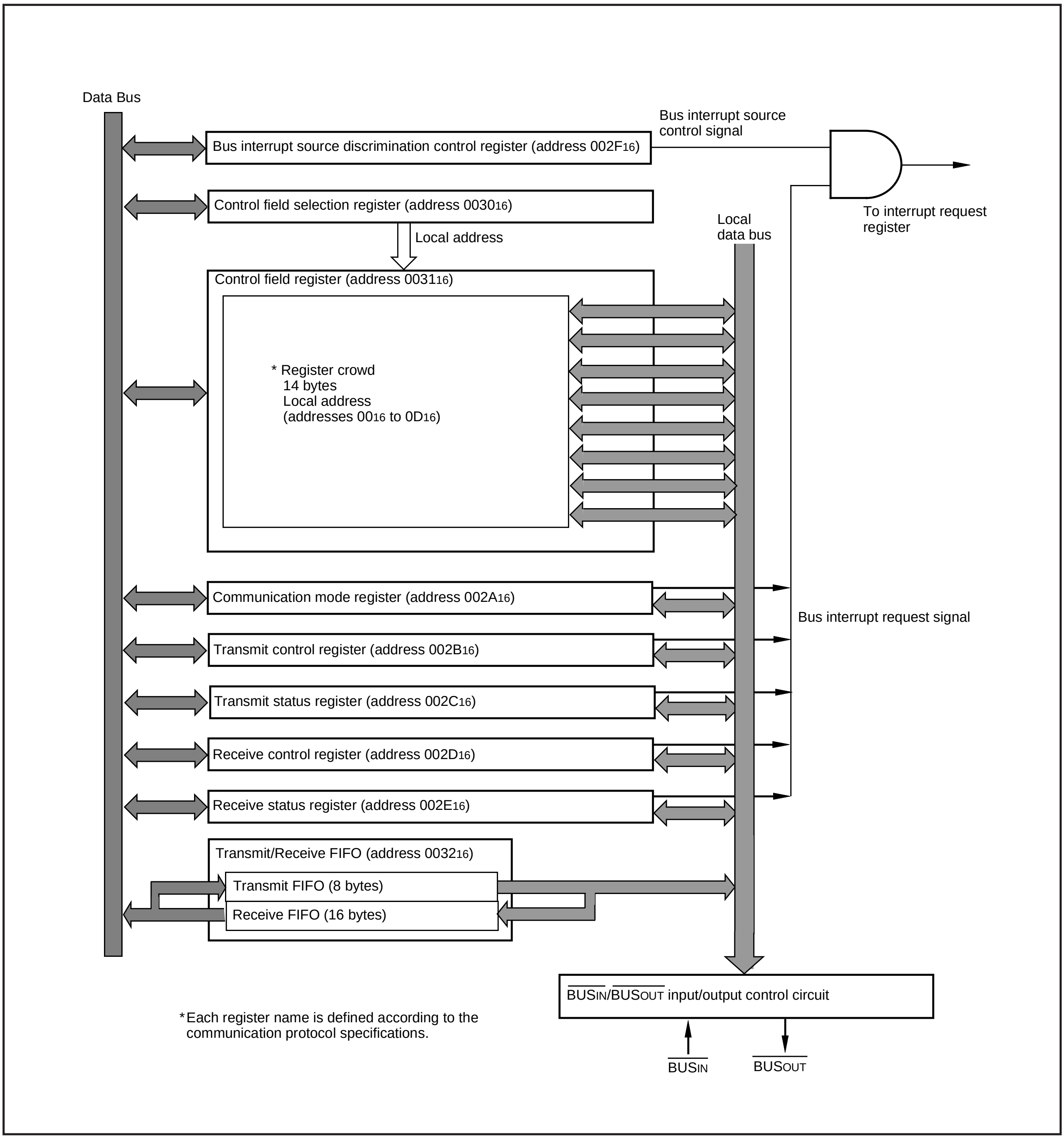

Data Bus
Bus interrupt source discrimination control register (address 002F16)
Bus interrupt source control signal
To interrupt request register
Control field selection register (address 003016)
Local address
Local data bus
Control field register (address 003116)
* Register crowd
14 bytes
Local address
(addresses 0016 to 0D16)
Communication mode register (address 002A16)
Bus interrupt request signal
Transmit control register (address 002B16)
Transmit status register (address 002C16)
Receive control register (address 002D16)
Receive status register (address 002E16)
Transmit/Receive FIFO (address 003216)
Transmit FIFO (8 bytes)
Receive FIFO (16 bytes)
BUSIN/BUSOUT input/output control circuit
*Each register name is defined according to the communication protocol specifications.
BUSIN
BUSOUT

Fig. 53 Data link layer communication control circuit block example

[Communication Mode Register (BUSM)] 002A$_{16}$

The communication mode register (address 002A$_{16}$) has 6 bits and consists of all the control bits for the communication mode.

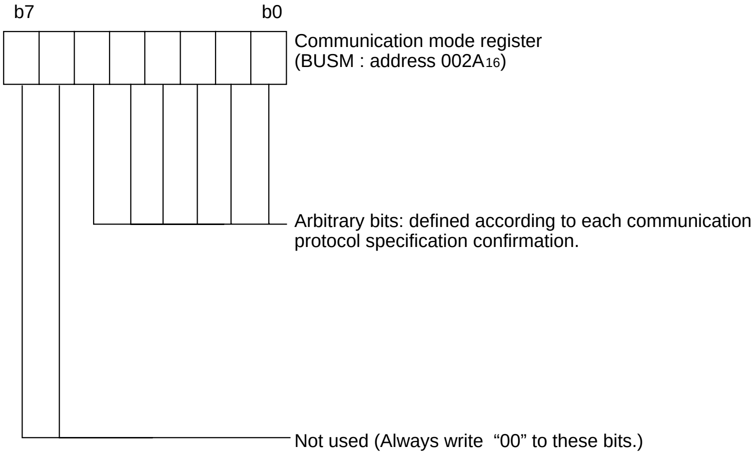

b7
b0
Communication mode register
(BUSM : address 002A16)
Arbitrary bits: defined according to each communication protocol specification confirmation.
Not used (Always write "00" to these bits.)

Fig. 54 Structure of communication mode register

[Transmit Control Register (TXDCON)] 002B16

The transmit control register (address 002B16) has 7 bits and consists of the transmit control and transmit status flags.

[Transmit Status Register (TXDSTS)] 002C16

The transmit status register (address 002C16) has 8 bits and consists of the transmit error flag and transmit interrupt request flag.

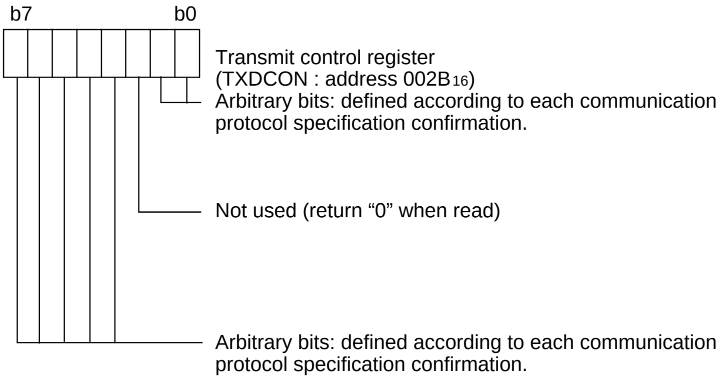
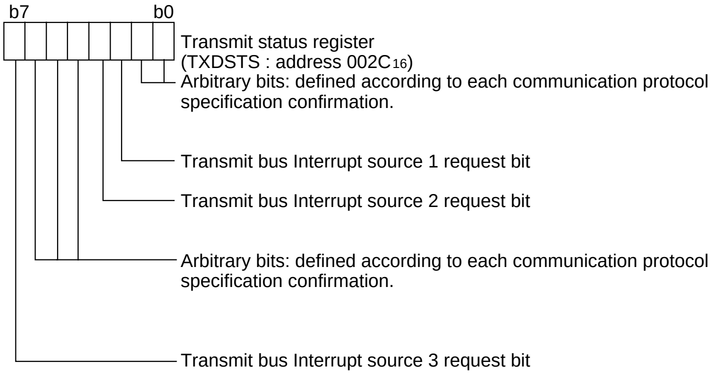

b7
b0
Transmit control register
(TXDCON : address 002B16)
Arbitrary bits: defined according to each communication protocol specification confirmation.
Not used (return "0" when read)
Arbitrary bits: defined according to each communication protocol specification confirmation.

Fig. 55 Structure of transmit control register


b7
b0
Transmit status register
(TXDSTS : address 002C16)
Arbitrary bits: defined according to each communication protocol specification confirmation.
Transmit bus Interrupt source 1 request bit
Transmit bus Interrupt source 2 request bit
Arbitrary bits: defined according to each communication protocol specification confirmation.
Transmit bus Interrupt source 3 request bit

Note: Bits 0 to 3, bit 5, and bit 7 can be cleared only by software.

When a transmit bus interrupt source request bit is "1," an interrupt request occurs. The name and function of each transmit bus interrupt source is defined according to the communication protocol specification confirmation.

Fig. 56 Structure of transmit status register

[Receive control register (RXDCON)] 002D$_{16}$

The receive control register has 7 bits and consists of the receive control and receive status flags.

[Receive status register (RXDSTS)] 002E$_{16}$

The receive status register has 8 bits and consists of the receive error flag and receive interrupt request flags.

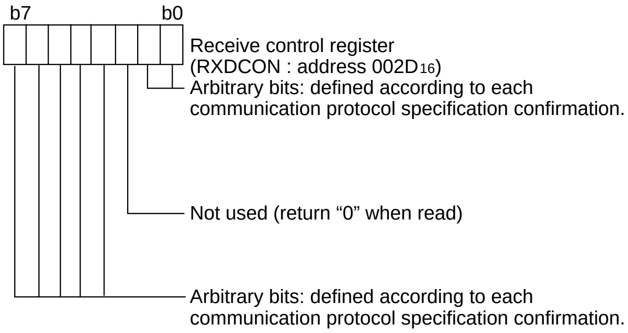
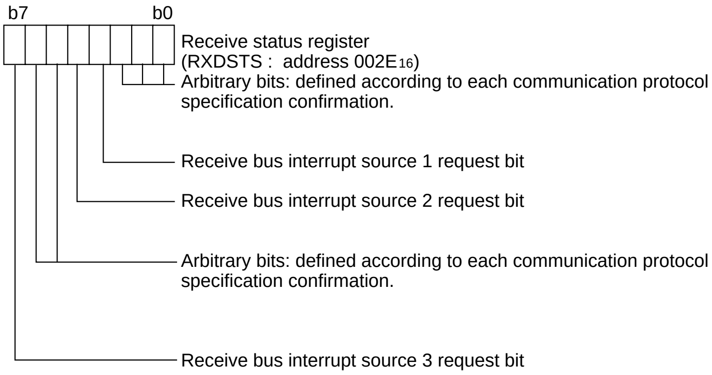

b7
b0
Receive control register
(RXDCON : address 002D16)
Arbitrary bits: defined according to each communication protocol specification confirmation.
Not used (return “0” when read)
Arbitrary bits: defined according to each communication protocol specification confirmation.

Fig. 57 Structure of receive control register


b7
b0
Receive status register
(RXDSTS : address 002E16)
Arbitrary bits: defined according to each communication protocol specification confirmation.
Receive bus interrupt source 1 request bit
Receive bus interrupt source 2 request bit
Arbitrary bits: defined according to each communication protocol specification confirmation.
Receive bus interrupt source 3 request bit

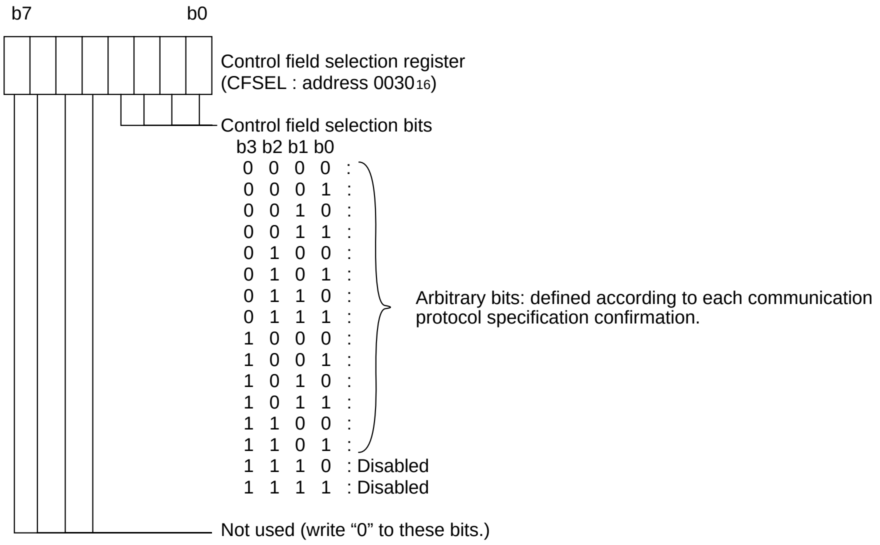
When a receive bus interrupt source request bit is “1”, an interrupt request occurs. The name and function of each receive bus interrupt source is defined according to the communication protocol specification confirmation.

Fig. 58 Structure of receive status register

[Control field selection register (CFSEL)] 0030_{16}
[Control field register (CF)] 0031_{16}

The control field data select the control field selection register (address 0030_{16}) value as the pointer. The data can be confirmed and changed by a read/write of the control field register (address 0030_{16}).

For example, when reading/writing the local address "00_{16}," the control field selection register is set to "00_{16}" and the control field register is read/written.

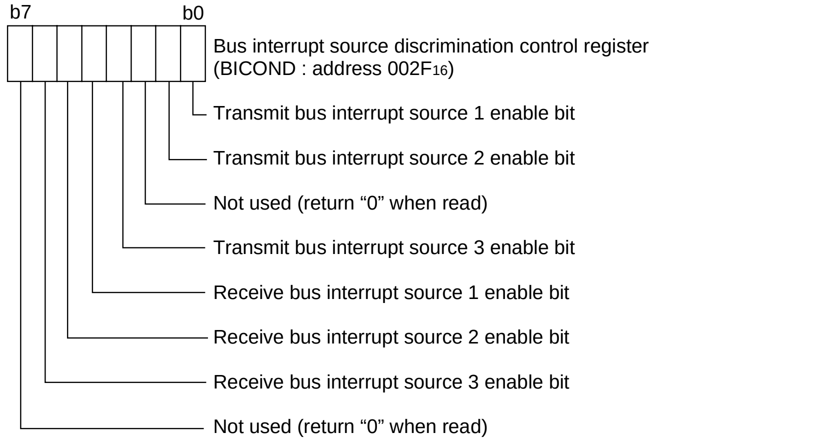

b7
b0
Control field selection register
(CFSEL : address 0030_{16})
Control field selection bits
Not used (write "0" to these bits.)

b3	b2	b1	b0	
0	0	0	0	: Arbitrary bits: defined according to each communication protocol specification confirmation.
0	0	0	1	:
0	0	1	0	:
0	0	1	1	:
0	1	0	0	:
0	1	0	1	:
0	1	1	0	:
0	1	1	1	:
1	0	0	0	:
1	0	0	1	:
1	0	1	0	:
1	0	1	1	:
1	1	0	0	:
1	1	0	1	:
1	1	1	0	: Disabled
1	1	1	1	: Disabled

Fig. 59 Structure of control field selection register

[Bus interrupt source discrimination control register (BICOND)] 002F16

The bus interrupt source discrimination control register (address 002F16) has 6 bits and controls bus-related interrupts. Refer to the section concerning interrupts for details about priority and vector addresses.


b7
b0
Bus interrupt source discrimination control register
(BICOND : address 002F16)
Transmit bus interrupt source 1 enable bit
Transmit bus interrupt source 2 enable bit
Not used (return “0” when read)
Transmit bus interrupt source 3 enable bit
Receive bus interrupt source 1 enable bit
Receive bus interrupt source 2 enable bit
Receive bus interrupt source 3 enable bit
Not used (return “0” when read)

0: Interrupt disabled
1: Interrupt enabled

The name and function of each transmit/receive bus interrupt source is defined according to the communication protocol specification confirmation.

Fig. 60 Structure of bus interrupt source discrimination control register

A-D CONVERTER
[A-D/D-A Conversion Register (AD)] 0035_{16}

The A-D/D-A conversion register is a register (at reading) that contains the result of an A-D conversion. When reading this register during an A-D conversion, the previous conversion result is read.

[A-D Control Register (ADCON)] 0034_{16}

The A-D control register controls the A-D/D-A conversion process. Bits 0 to 2 of this register select specific analog input pins. Bit 3 signals the completion of an A-D conversion. The value of this bit remains at “0” during an A-D conversion, then changes to “1” when the A-D conversion is completed. Writing “0” to this bit starts the A-D conversion. When bit 5, which is the AD external trigger valid bit, is set to “1”, this bit enables A-D conversion even by a falling edge of an ADT input. Set “0” (input port) to the direction register corresponding the ADT pin. Bit 6 is the interrupt source selection bit. Writing “0” to this bit, A-D converter interrupt request occurs at completion of A-D conversion. Writing “1” to this bit the interrupt request occurs at falling edge of an ADT input.

Comparison Voltage Generator

The comparison voltage generator divides the voltage between AV_{SS} and V_{REF} by 256, and outputs the divided voltages.

Channel Selector

The channel selector selects one of the input ports $P6_7/AN_7$ to $P6_0/AN_0$ and inputs it to the comparator.

Comparator and Control Circuit

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

Note that the comparator is constructed linked to a capacitor, so set $f(X_{IN})$ to at least 500 kHz during A-D conversion. Use a CPU system clock dividing the main clock X_{IN}.

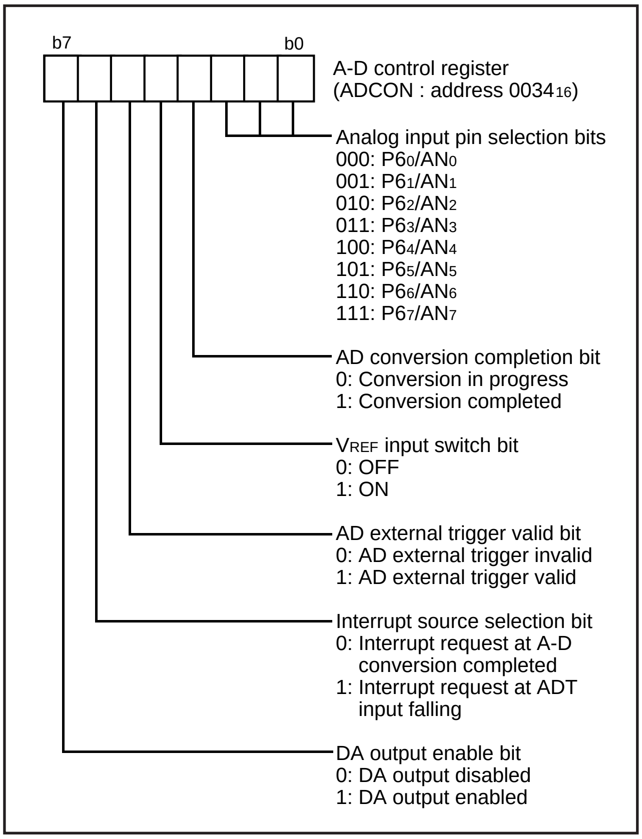

b7
b0
A-D control register
(ADCON : address 0034_{16})
Analog input pin selection bits
000: $P6_0/AN_0$
001: $P6_1/AN_1$
010: $P6_2/AN_2$
011: $P6_3/AN_3$
100: $P6_4/AN_4$
101: $P6_5/AN_5$
110: $P6_6/AN_6$
111: $P6_7/AN_7$
AD conversion completion bit
0: Conversion in progress
1: Conversion completed
V_{REF} input switch bit
0: OFF
1: ON
AD external trigger valid bit
0: AD external trigger invalid
1: AD external trigger valid
Interrupt source selection bit
0: Interrupt request at A-D conversion completed
1: Interrupt request at ADT input falling
DA output enable bit
0: DA output disabled
1: DA output enabled

Fig. 61 Structure of A-D control register

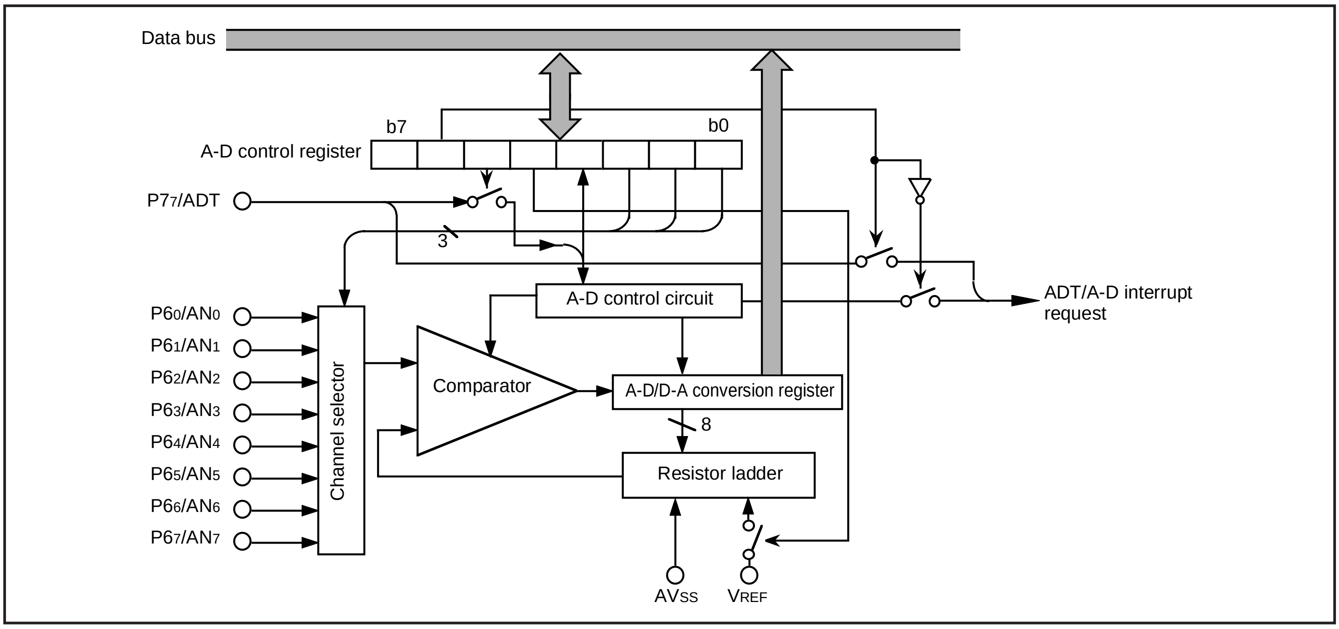


Fig. 62 Block diagram of A-D converter

D-A CONVERTER

The 3874 group has an on-chip D-A converter with 8-bit resolution and 1 channel. The D-A conversion is performed by setting the value in the A-D/D-A conversion register. The result of D-A converter is output from DA pin by setting the DA output enable bit to “1”. When using the D-A converter, the corresponding port direction register bit (P8$_0$/DA) should be set to “0” (input status).

The output analog voltage V is determined by the value n (base 10) in the A-D/D-A conversion register as follows:

V=VREF × n/256 (n=0 to 255)
Where VREF is the reference voltage.

At reset, the A-D/D-A conversion registers are cleared to “00$_{16}$”, the DA output enable bit is cleared to “0”, and P8$_0$/DA pin becomes high impedance. The DA output is not buffered, so connect an external buffer when driving a low-impedance load. When using D-A converter, set 4.0 V or more to VCC.

■ Note

When reading the A-D/D-A conversion register, the A-D conversion result is read, and the set value for D-A conversion is not read.

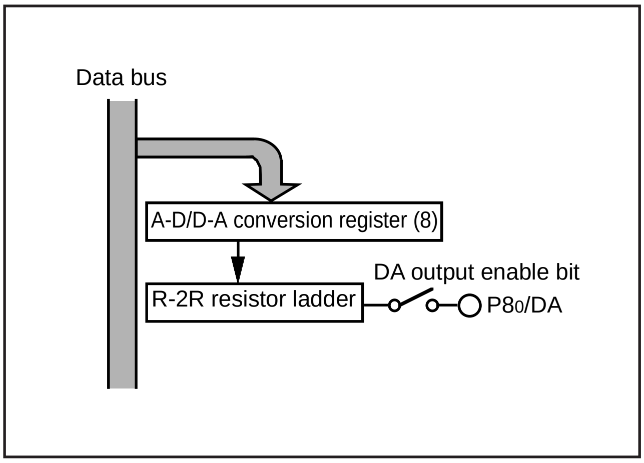


Fig. 63 Block diagram of D-A converter

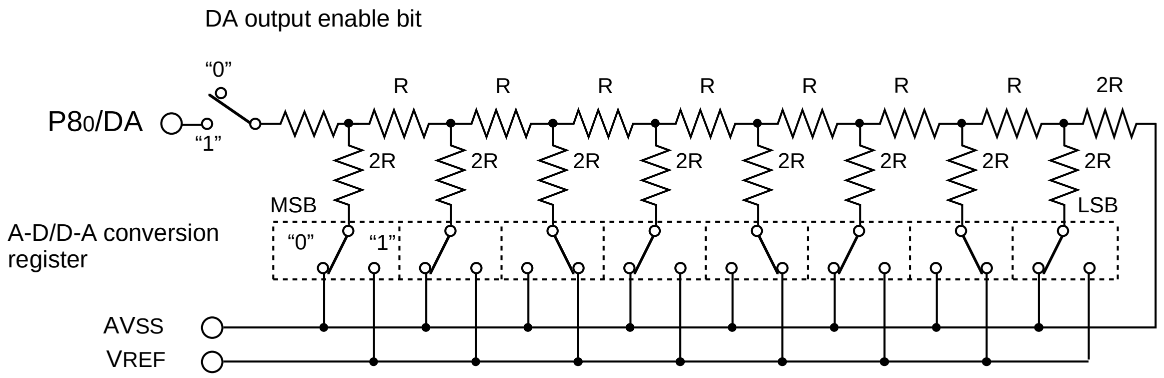


Fig. 64 Equivalent connection circuit of D-A converter

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 a 12-bit watchdog timer H.

Watchdog Timer Initial Value

Watchdog timer L is set to "FF_{16}" and watchdog timer H is set to "FFF_{16}" by writing to the watchdog timer control register 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 control register. Regardless of the value written to bits 0 to 5, the above-mentioned value will be set to each timer.

Watchdog Timer Operations

The watchdog timer stops at reset and a countdown is started by the writing to the watchdog timer control register. An internal reset occurs when watchdog timer H underflows. The reset is released after its release time. After the release, the program is restarted from the reset vector address. Usually, write to the watchdog timer control register by software before an underflow of the watchdog timer H. The watchdog timer does not function if the watchdog timer control register is not written to at least once.

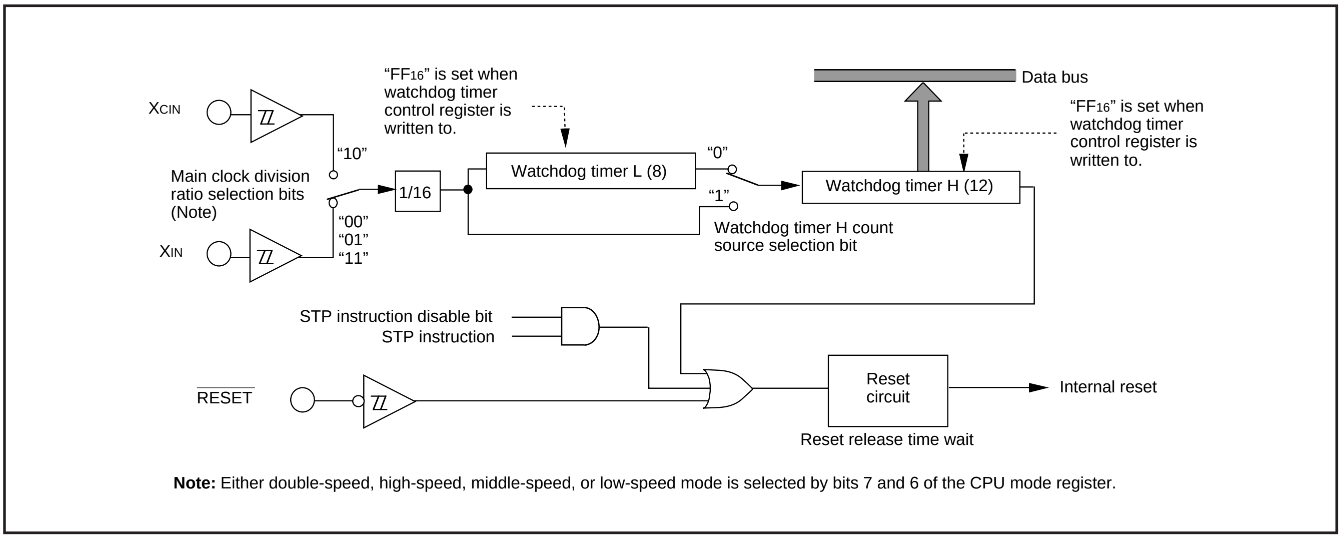
When bit 6 of the watchdog timer control register is kept at "0", the STP instruction is enabled. When that is executed, both the clock and the watchdog timer stop. Count re-starts at the same time as the release of stop mode **(Note)**. The watchdog timer does not stop while a WIT instruction is executed. In addition, the STP instruction is disabled by writing "1" to this bit again. When the STP instruction is executed at this time, it is processed as an undefined instruction, and an internal reset occurs. Once a "1" is written to this bit, it cannot be programmed to "0" again.

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 X_{CIN} = 32 kHz; 524 s
when X_{IN} = 6.4 MHz; 2.6 s

Bit 7 of the watchdog timer control register is "1":
when X_{CIN} = 32 kHz; 2 s
when X_{IN} = 6.4 MHz; 10 ms

Note: The watchdog timer continues to count even while waiting for a stop release. Therefore, make sure that watchdog timer H does not underflow during this period.


"FF16" is set when watchdog timer control register is written to.
Data bus
"FF16" is set when watchdog timer control register is written to.
XCIN
"10"
Main clock division ratio selection bits (Note)
1/16
Watchdog timer L (8)
"0"
"1"
Watchdog timer H (12)
Watchdog timer H count source selection bit
"00"
"01"
"11"
XIN
STP instruction disable bit
STP instruction
RESET
Reset circuit
Internal reset
Reset release time wait

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

Fig. 65 Block diagram of Watchdog timer

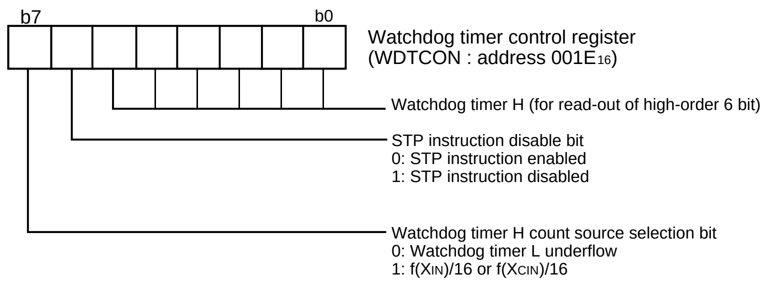

b7
b0
Watchdog timer control register
(WDTCON : address 001E16)
Watchdog timer H (for read-out of high-order 6 bit)
STP instruction disable bit
0: STP instruction enabled
1: STP instruction disabled
Watchdog timer H count source selection bit
0: Watchdog timer L underflow
1: f(XIN)/16 or f(XCIN)/16

Fig. 66 Structure of Watchdog timer control register

RESET CIRCUIT

To reset the microcomputer, $\overline{\text{RESET}}$ pin should be held at an "L" level for 2 µs or more. Then the $\overline{\text{RESET}}$ pin is returned to an "H" level (the power source voltage should be between 3.0 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 $FFFD_{16}$ (high-order byte) and address $FFFC_{16}$ (low-order byte). Make sure that the reset input voltage is 0.6 V or less for V_{CC} of 3.0 V.

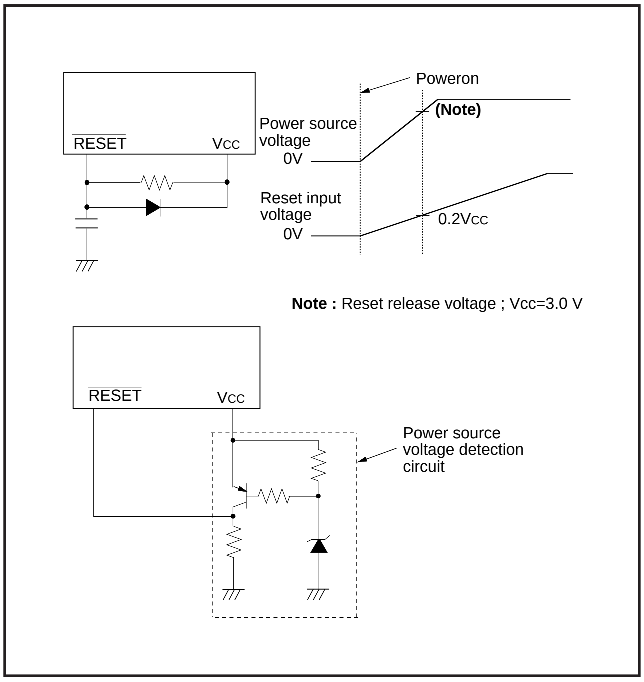
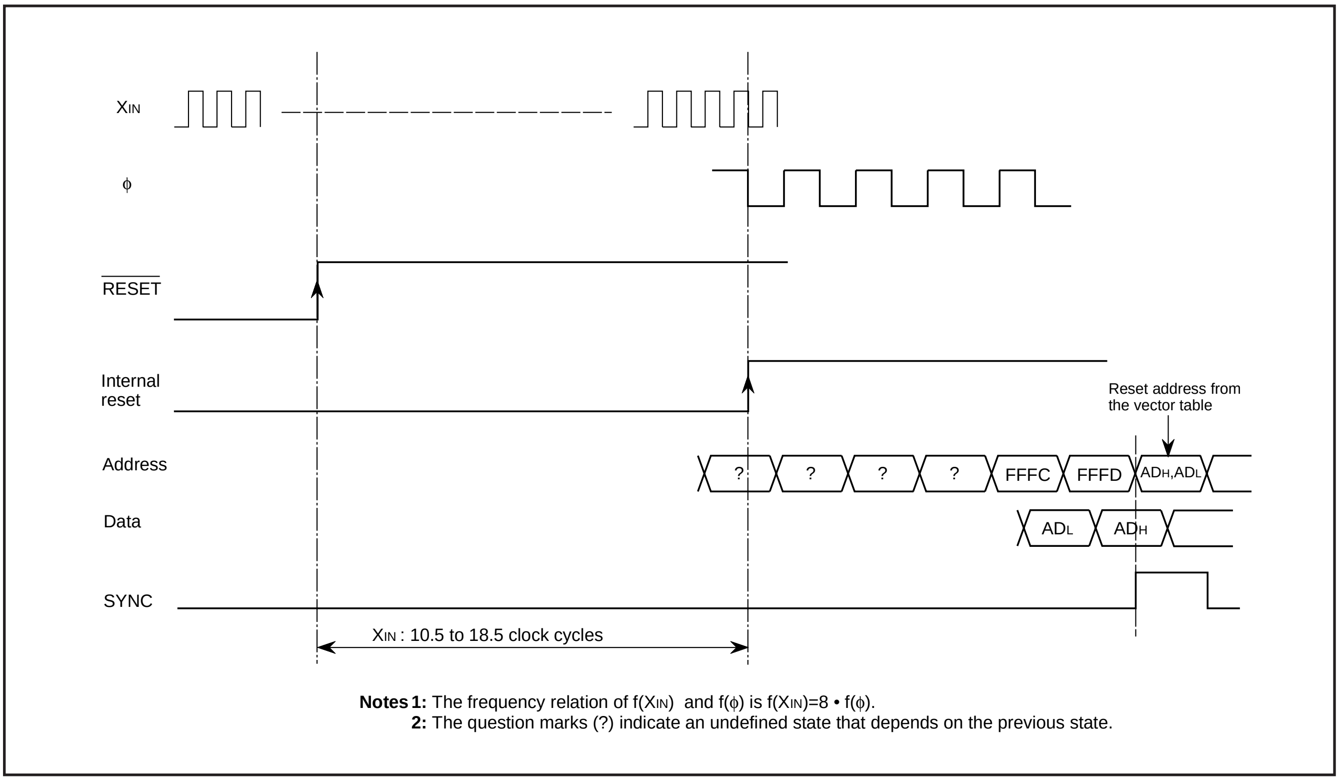

Poweron
(Note)
Power source voltage
0V
Reset input voltage
0V
0.2VCC
RESET
VCC
Note : Reset release voltage ; Vcc=3.0 V
RESET
VCC
Power source voltage detection circuit

Fig. 67 Reset circuit example


XIN
φ
RESET
Internal reset
Address
?
?
?
?
FFFC
FFFD
ADH,ADL
Data
ADL
ADH
SYNC
Reset address from the vector table
XIN : 10.5 to 18.5 clock cycles

Notes 1: The frequency relation of $f(X_{IN})$ and $f(\phi)$ is $f(X_{IN})=8 \cdot f(\phi)$.
2: The question marks (?) indicate an undefined state that depends on the previous state.

Fig. 68 Reset sequence

		Address	Register contents
(1)	Port P0	0000_{16}	00_{16}
(2)	Port P0 direction register	0001_{16}	00_{16}
(3)	Port P1	0002_{16}	00_{16}
(4)	Port P1 direction register	0003_{16}	00_{16}
(5)	Port P2	0004_{16}	00_{16}
(6)	Port P2 direction register	0005_{16}	00_{16}
(7)	Port P3	0006_{16}	00_{16}
(8)	Port P3 direction register	0007_{16}	00_{16}
(9)	Port P4	0008_{16}	00_{16}
(10)	Port P4 direction register	0009_{16}	00_{16}
(11)	Port P5	$000A_{16}$	00_{16}
(12)	Port P5 direction register	$000B_{16}$	00_{16}
(13)	Port P6	$000C_{16}$	00_{16}
(14)	Port P6 direction register	$000D_{16}$	00_{16}
(15)	Port P7	$000E_{16}$	00_{16}
(16)	Port P7 direction register	$000F_{16}$	00_{16}
(17)	Port P8	0010_{16}	00_{16}
(18)	Port P8 direction register	0011_{16}	00_{16}
(19)	Port P9	0012_{16}	× 0 0 0 0 0 0 0
(20)	Serial I/O3 control register 1	0014_{16}	00_{16}
(21)	Serial I/O3 control register 2	0015_{16}	00_{16}
(22)	Serial I/O3 control register 3	0016_{16}	00_{16}
(23)	Serial I/O3 automatic transfer data pointer	0017_{16}	00_{16}
(24)	Serial I/O1 status register	0019_{16}	80_{16}
(25)	Serial I/O1 control register	$001A_{16}$	00_{16}
(26)	UART control register	$001B_{16}$	$E0_{16}$
(27)	Serial I/O2 control register	$001D_{16}$	00_{16}
(28)	Watchdog timer control register	$001E_{16}$	$3F_{16}$
(29)	Timer X (low-order)	0020_{16}	FF_{16}
(30)	Timer X (high-order)	0021_{16}	FF_{16}
(31)	Timer Y (low-order)	0022_{16}	FF_{16}
(32)	Timer Y (high-order)	0023_{16}	FF_{16}
(33)	Timer 1	0024_{16}	FF_{16}
(34)	Timer 2	0025_{16}	01_{16}
(35)	Timer 3	0026_{16}	FF_{16}
(36)	Timer X mode register	0027_{16}	00_{16}
(37)	Timer Y mode register	0028_{16}	00_{16}
(38)	Timer 123 mode register	0029_{16}	00_{16}
(39)	Communication mode register	$002A_{16}$	0 0 0 0 × × × 0
(40)	Transmit control register	$002B_{16}$	20_{16}
(41)	Transmit status register	$002C_{16}$	00_{16}
(42)	Receive control register	$002D_{16}$	10_{16}
(43)	Receive status register	$002E_{16}$	01_{16}
(44)	Bus interrupt source discrimination control register	$002F_{16}$	00_{16}
(45)	Control field selection register	0030_{16}	00_{16}
(46)	PULL UP register	0033_{16}	00_{16}
(47)	A-D control register	0034_{16}	08_{16}
(48)	Interrupt source discrimination register 2	0036_{16}	00_{16}
(49)	Interrupt source discrimination control register 2	0037_{16}	00_{16}
(50)	Interrupt source discrimination register 1	0038_{16}	00_{16}
(51)	Interrupt source discrimination control register 1	0039_{16}	00_{16}
(52)	Interrupt edge selection register	$003A_{16}$	00_{16}
(53)	CPU mode register	$003B_{16}$	48_{16}
(54)	Interrupt request register 1	$003C_{16}$	00_{16}
(55)	Interrupt request register 2	$003D_{16}$	00_{16}
(56)	Interrupt control register 1	$003E_{16}$	00_{16}
(57)	Interrupt control register 2	$003F_{16}$	00_{16}
(58)	Processor status register	(PS)	× × × × × 1 × ×
(59)	Program counter	(PCH)	$FFFD_{16}$ contents
		(PCL)	$FFFC_{16}$ contents

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

Fig. 69 Internal status at reset

CLOCK GENERATING CIRCUIT

The 3874 group has two built-in oscillation circuits. 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. 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, this mode is selected.

(2) Double-speed mode

The internal clock ϕ is the frequency of XIN.

(3) High-speed mode

The internal clock ϕ is half the frequency of XIN.

(4) Low-speed mode

The internal clock ϕ is half the frequency of XCIN.

■ Note

When switching the mode between double/middle/high-speed and low-speed, stabilize both XIN and XCIN oscillations. 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 double/middle/high-speed and low-speed, set the frequency on condition that f(XIN) > 3f(XCIN).

It takes the cycle number mentioned below to switch between each mode (machine cycle = cycle of internal clock ϕ).

Double-speed mode→Except double-speed mode
    1 to 8 machine cycles
High-speed mode→Except high-speed mode
    1 to 4 machine cycles
Middle-speed mode→Except middle-speed mode
    1 machine cycle
Low-speed mode→Except low-speed mode
    1 to 4 machine cycles

The 3874 group operates in the previous mode while the mode is switched.

(5) 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.

By clearing furthermore the XCOUT drivability selection bit (b3) of the CPU mode register to "0", low power consumption operation can be realized by reducing the drivability between XCIN and XCOUT. At reset or during STP instruction execution this bit is set to "1" and a reduced drivability that has an easy oscillation start is set. The sub-clock XCIN-XCOUT oscillating circuit can no directly input clocks that are generated externally. Accordingly, make sure to cause an external resonator to oscillate.

Oscillation Control
(1) Stop mode

When the STP instruction is executed, the internal clock ϕ stops at an "H" level, and XIN and XCIN oscillators stop. The value set to the timer 1 latch and the timer 2 latch is set to timer 1 and timer 2. Either XIN or XCIN divided by 16 is input to timer 1 as count source, and the output of timer 1 is connected to timer 2. The bits of the timer 123 mode register except the timer 3 count source selection bit (b4) are cleared to "0". Set the interrupt enable bits of timer 1 and timer 2 to the disabled state ("0") before executing the STP instruction.

Oscillator restarts at reset or when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU until timer 2 underflows. This allows time for the clock circuit oscillation to stabilize. Timer 1 latch and timer 2 latch should be set to proper values for stabilizing oscillation before executing the STP instruction.

(2) Wait mode

If the WIT instruction is executed, the internal clock ϕ stops at an "H" level. The states of XIN and XCIN are the same as the state before executing the WIT instruction. 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.

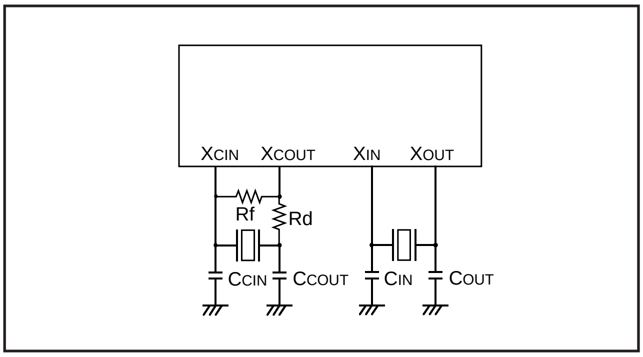


Fig. 70 Ceramic resonator circuit

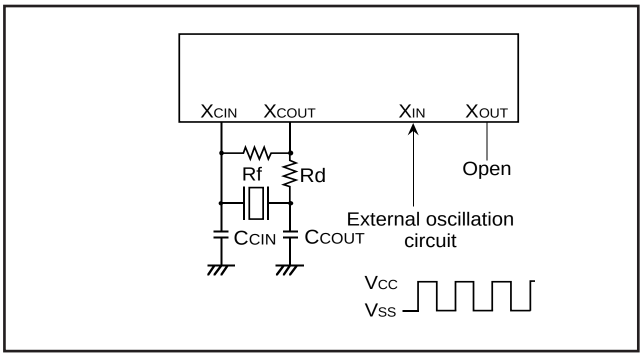
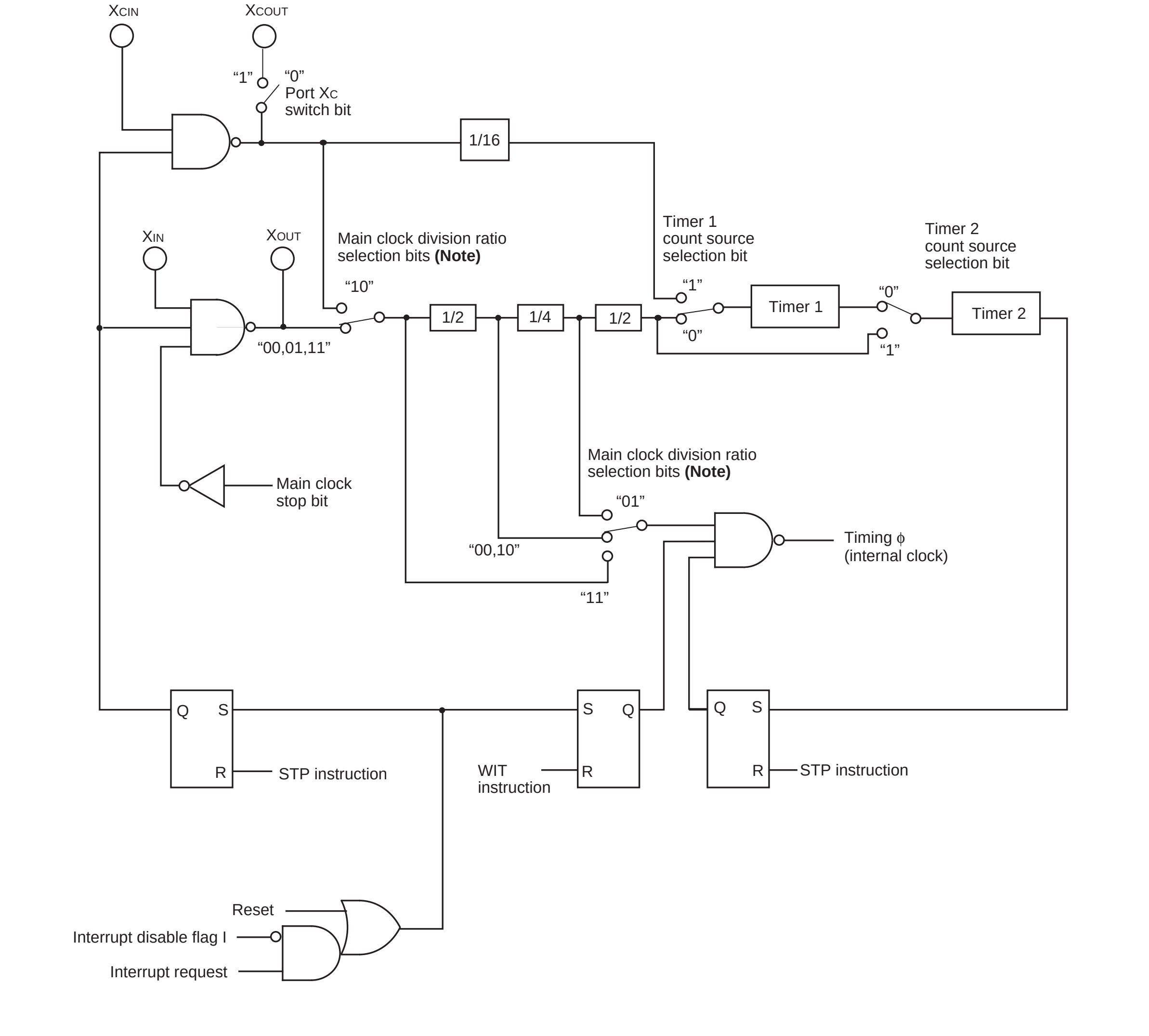
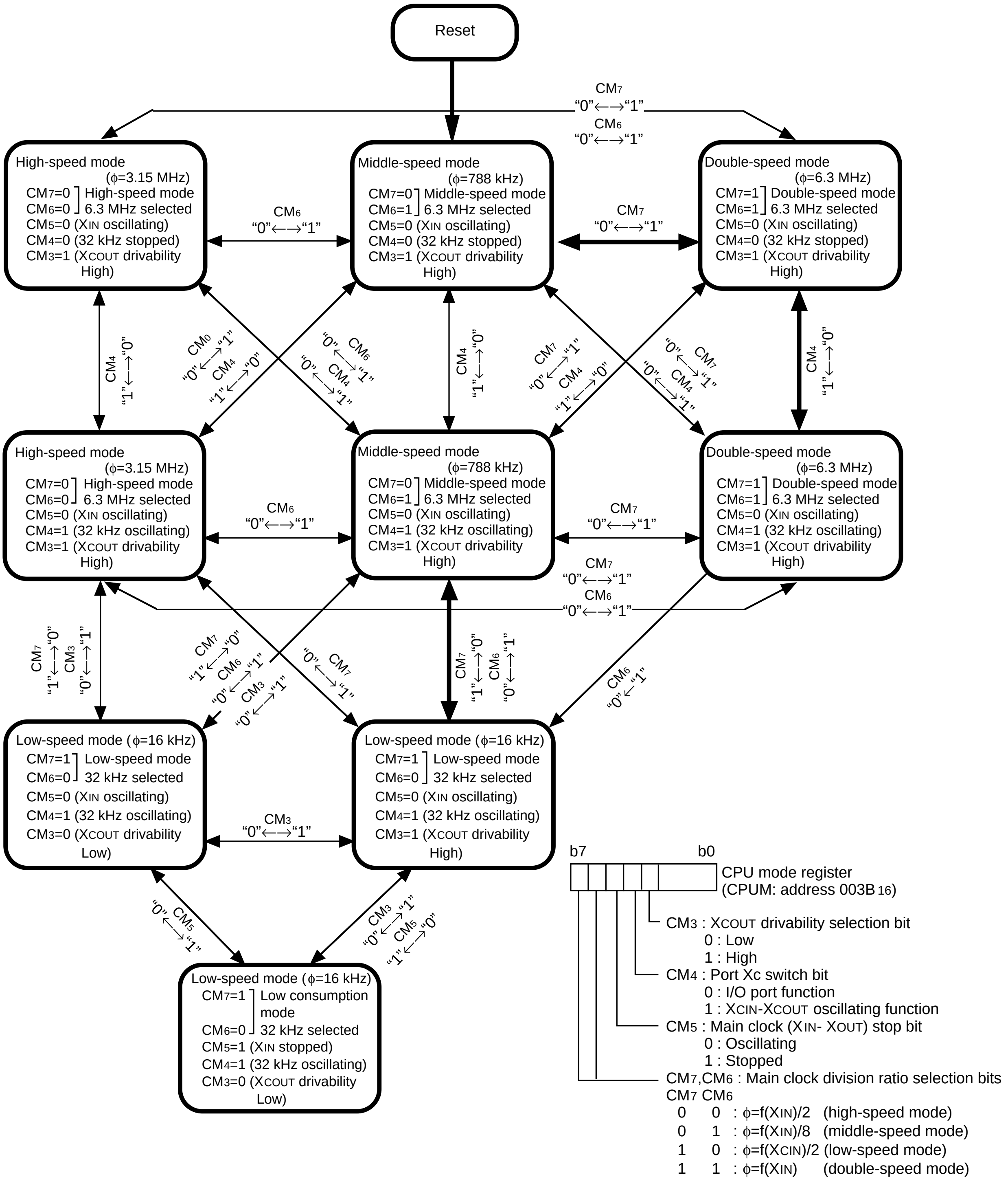


Fig. 71 External clock input circuit



Note: When low-speed mode is selected, set port XC switch bit (b4) to "1."

Fig. 72 System clock generating circuit block diagram


Reset
CM7 "0"←→"1" CM6 "0"←→"1"
High-speed mode (ϕ=3.15 MHz)
CM7=0 CM6=0 High-speed mode 6.3 MHz selected
CM5=0 (XIN oscillating)
CM4=0 (32 kHz stopped)
CM3=1 (XCOUT drivability High)
Middle-speed mode (ϕ=788 kHz)
CM7=0 CM6=1 Middle-speed mode 6.3 MHz selected
CM5=0 (XIN oscillating)
CM4=0 (32 kHz stopped)
CM3=1 (XCOUT drivability High)
Double-speed mode (ϕ=6.3 MHz)
CM7=1 CM6=1 Double-speed mode 6.3 MHz selected
CM5=0 (XIN oscillating)
CM4=0 (32 kHz stopped)
CM3=1 (XCOUT drivability High)
CM6 "0"←→"1"
CM7 "0"←→"1"
CM4 "1"←→"0"
CM6 "1"←→"0" CM4 "0"←→"1"
"0" CM6 ←→"1" "0" CM4 ←→"1"
CM7 "1"←→"0" CM4 "0"←→"1"
"0" CM7 ←→"1" "0" CM4 ←→"1"
High-speed mode (ϕ=3.15 MHz)
CM7=0 CM6=0 High-speed mode 6.3 MHz selected
CM5=0 (XIN oscillating)
CM4=1 (32 kHz oscillating)
CM3=1 (XCOUT drivability High)
Middle-speed mode (ϕ=788 kHz)
CM7=0 CM6=1 Middle-speed mode 6.3 MHz selected
CM5=0 (XIN oscillating)
CM4=1 (32 kHz oscillating)
CM3=1 (XCOUT drivability High)
Double-speed mode (ϕ=6.3 MHz)
CM7=1 CM6=1 Double-speed mode 6.3 MHz selected
CM5=0 (XIN oscillating)
CM4=1 (32 kHz oscillating)
CM3=1 (XCOUT drivability High)
CM6 "0"←→"1"
CM7 "0"←→"1"
CM7 "0"←→"1" CM6 "0"←→"1"
CM7 "1"←→"0" CM3 "0"←→"1"
CM7 "1"←→"0" CM6 "0"←→"1" CM3 "0"←→"1"
"0" CM7 ←→"1"
CM7 "1"←→"0" CM6 "0"←→"1"
CM6 "0"←"1"
Low-speed mode (ϕ=16 kHz)
CM7=1 CM6=0 Low-speed mode 32 kHz selected
CM5=0 (XIN oscillating)
CM4=1 (32 kHz oscillating)
CM3=0 (XCOUT drivability Low)
Low-speed mode (ϕ=16 kHz)
CM7=1 CM6=0 Low-speed mode 32 kHz selected
CM5=0 (XIN oscillating)
CM4=1 (32 kHz oscillating)
CM3=1 (XCOUT drivability High)
CM3 "0"←→"1"
"0" CM5 ←→"1"
CM3 "0"←→"1" CM5 "1"←→"0"
Low-speed mode (ϕ=16 kHz)
CM7=1 CM6=0 Low consumption mode 32 kHz selected
CM5=1 (XIN stopped)
CM4=1 (32 kHz oscillating)
CM3=0 (XCOUT drivability Low)
b7 b0
CPU mode register (CPUM: address 003B16)
CM3 : XCOUT drivability selection bit
0 : Low
1 : High
CM4 : Port Xc switch bit
0 : I/O port function
1 : XCIN-XCOUT oscillating function
CM5 : Main clock (XIN- XOUT) stop bit
0 : Oscillating
1 : Stopped
CM7,CM6 : Main clock division ratio selection bits
CM7 CM6
0 0 : ϕ=f(XIN)/2 (high-speed mode)
0 1 : ϕ=f(XIN)/8 (middle-speed mode)
1 0 : ϕ=f(XCIN)/2 (low-speed mode)
1 1 : ϕ=f(XIN) (double-speed mode)

Notes 1: Switch the mode by the arrows shown between the mode blocks. (Do not switch between the modes directly without an arrow.)
2: 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, wait time is generated automatically by connecting timer 1 and timer 2.
5: The example assumes that 6.3 MHz is being applied to the X IN pin and 32 kHz to the X CIN pin. ϕ indicates the internal clock.
6: We recommend that X COUT drivability selection bit is set to "1" (high) because reliance of oscillation stability is improved.

Fig. 73 State transitions of system clock

NOTES ON PROGRAMMING

Processor Status Register

The contents of the processor status register (PS) after a reset are undefined, except for the interrupt disable flag (I) which is “1”. After a reset, initialize flags which affect program execution. In particular, it is essential to initialize the index X mode (T) and the decimal mode (D) flags because of their effect on calculations.

Interrupts

The contents of the interrupt request bits do not change immediately after they have been written. After writing to an interrupt request register, execute at least one instruction before performing a BBC or BBS instruction.

Interrupt Source Discrimination

- Use LDM, STA, etc., instructions to clear interrupt request bits assigned to the interrupt source discrimination register 1, the interrupt source discrimination register 2, the transmit status register, or the receive status register. (Do not use read-modify-write instructions such as CLB, SEB, etc. Use the LDM or STA instruction to clear these bits.)
- Request bits of interrupt source discrimination registers are not automatically cleared when an interrupt occurs. After an interrupt source has been discriminated, and before execution of the RTI or CLI instruction, the user must clear the bit by program. (Use the LDM or STA instruction to clear.)
- The interrupt assigned to the interrupt source discrimination registers occur 1 instruction execution later than a normal interrupt. The maximum timing is 16 machine cycles in the MUL, DIV instructions.

Decimal Calculations

- To calculate in decimal notation, set the decimal mode flag (D) to “1”, then execute an ADC or SBC instruction. After executing an ADC or SBC instruction, execute at least one instruction before executing a SEC, CLC, or CLD instruction.
- In decimal mode, the values of the negative (N), overflow (V), and zero (Z) flags are invalid.

Timers

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

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.

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 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.

Serial I/O1

- In clock synchronous serial I/O, if the receive side is using an external clock and it is to output the $\overline{\text{SRDY1}}$ signal, set the transmit enable bit, the receive enable bit, and the $\overline{\text{SRDY}}$ output enable bit to “1”.
  Serial I/O1 continues to output the final bit from the TxD pin after transmission is completed.
- In order to stop a transmit, set the transmit enable bit to “0” (transmit disable).
  Do not set only the serial I/O1 enable bit to “0”.
- A receive operation can be stopped by either setting the receive enable bit to “0” or the serial I/O1 enable bit to “0”.
- To stop a transmit when transferring in clock synchronous serial I/O mode, set both the transmit enable bit and the receive enable bit to “0” at the same time.
- To set the serial I/O1 control register again, first set the transmit enable/receive enable bits to “0”. Next, reset the transmit/receive circuits, and, finally, reset the serial I/O1 control register.
- Note when confirming the transmit shift register completion flag and controlling the data transmit after writing a transmit data to the transmit buffer. There is a delay of 0.5 to 1.5 shift clock cycles while the transmit shift register completion flag goes from “1” to “0”.

Serial I/O3

- When writing “1” to the serial I/O initialization bit of the serial I/O3 control register 1, serial I/O3 is enabled, but each register is not initialized. Set the value of each register by program.
- A serial I/O3 interrupt request occurs when “0” is written to the serial I/O initialization bit during an operation in automatic transfer serial I/O mode. Disable the interrupt enable bit as necessary by program.

A-D Converter/D-A Converter

- The A-D/D-A conversion register functions as an A-D conversion register during a read and a D-A conversion during a write. Accordingly, the D-A conversion register set value cannot be read out.
- The comparator for A-D converter uses capacitive coupling amplifier whose charge will be lost if the clock frequency is too low. Therefore, make sure that f(X_{IN}) is at least on 500 kHz during an A-D conversion.

  Do not execute the STP or WIT instruction during an A-D conversion.

Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency of the internal clock ϕ by the number of cycles needed to execute an instruction. The number of cycles required to execute an instruction is shown in the list of machine instructions.

The frequency of the internal clock ϕ is half of the X_{IN} frequency.

Data Link Layer Communication Control

- The data link layer communication control circuit stops after a reset. To restart or change modes, write “$00XXXXX1_2$” to the communication mode register. Note that bits 4 and 5 are read-only bits.
- The $P7_5/\overline{BUS_{OUT}}$ pin operates as a general-purpose pin after release from reset. As a general-purpose port, its input/output can be switched by the direction register.

Clock Changes

- Use the LDM, STA, etc. instructions to modify the division ratio of internal system clock ϕ. (Do not use read-modify-write instructions such as CLB, SEB, etc.)
- Do not modify the division ratio of the internal system clock until the mode has been changed. For details concerning the number of cycles necessary to change modes, refer to the clock section in the explanation of about function blocks.
- Use the LDM, STA, etc., instructions to clear interrupt request bits assigned to the interrupt source discrimination register 1, the interrupt source discrimination register 2, the transmit status register, or the receive status register. (Do not use read-modify-write instructions such as CLB, SEB, etc.)
- Before executing the CLI or RTI instruction during an interrupt processing routine, use the LDM or STA instruction to clear the interrupt request bits of interrupt source discrimination registers which have completed the interrupt processing.
- If switching the mode between low-speed and double-speed, switch the mode to middle/high-speed first, and then switch the mode to double-speed by program. Do not switch the mode from low-speed to double-speed directly. 1 to 4 machine cycles are required for switching from low-speed mode to other mode. Insert “clock switch timing wait” for switching the mode to middle/high-speed, and then switch the mode to double-speed.

  Table 10 lists the recommended transition process for system clock switch.

  Figure 74 shows the program example.

Table 10 Clock switch combination

Recommended transition process	
Low-speed→High-speed	Middle-speed→High-speed
Low-speed→Middle-speed	Middle-speed→Middle-speed
Double-speed→High-speed	Middle-speed→Low-speed
Double-speed→Middle-speed	High-speed→Double-speed
Double-speed→Low-speed	High-speed→MIddle-speed
	High-speed→Low-speed

```
Low-speed mode → Middle/High-speed mode → Double-speed mode switch
  LDM  xx, CPUM  •••Low-speed mode → Middle/High-speed mode switch
  NOP        )      Clock switch timing wait
  NOP        )      (1 to 4 machine cycles are required for switching mode.)
  LDM  yy, CPUM  •••Switch mode to double-speed

         Note: CPUM = CPU mode register (address 003B16)
```

Fig. 74 Program example

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

1. Mask ROM Order Confirmation Form
2. Mark Specification Form
3. Data to be written to ROM, in EPROM form (three identical copies)

DATA REQUIRED FOR ROM WRITING ORDERS

The following are necessary when ordering a ROM writing:

1. ROM Writing Confirmation Form
2. Mark Specification Form
3. Data to be written to ROM, in EPROM form (three identical copies)

ROM PROGRAMMING METHOD

The PROM of the blank One Time PROM version is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Figure 75 is recommended to verify programming.

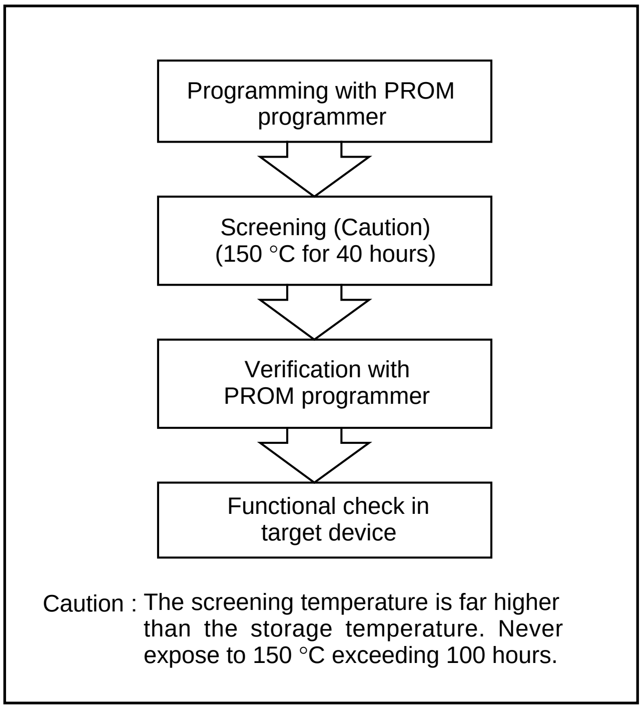

Programming with PROM programmer
Screening (Caution) (150 °C for 40 hours)
Verification with PROM programmer
Functional check in target device
Caution : The screening temperature is far higher than the storage temperature. Never expose to 150 °C exceeding 100 hours.

Fig. 75 Programming and testing of One Time PROM version

FUNCTIONAL DESCRIPTION SUPPLEMENT

Interrupt

3874 group permits interrupts on the basis of 27 sources. The interrupt control circuit consists of “one factor/one vector interrupt” and “multiple factors/one vector interrupt.”

It is vector interrupts with a fixed priority system. Accordingly, when two or more interrupt requests occur during the same sampling, the higher-priority interrupt is accepted first. This priority is determined by hardware, but various priority processing can be performed by software, using an interrupt enable bit and an interrupt disable flag.

For “multiple factors/one vector interrupt,” the priority in the same vector can be controlled freely by software, referring an interrupt request bit.

For interrupt sources, vector addresses and interrupt priority, refer to Table 11.

Table 11 Interrupt sources, vector addresses and interrupt priority

Priority	Interrupt sources	Vector addresses (Note 1)		Remarks
		High-order	Low-order	
1	Reset **(Note 2)**	FFFD$_{16}$	FFFC$_{16}$	Non-maskable
2	INT$_0$	FFFB$_{16}$	FFFA$_{16}$	External interrupt (active edge selectable)
3	INT$_1$	FFF9$_{16}$	FFF8$_{16}$	External interrupt (active edge selectable)
4	Receive bus interrupt source 1	FFF7$_{16}$	FFF6$_{16}$	The condition which the receive bus interrupt factor request bit becomes “1” is defined according to each communication protocol specification confirmation.
	Receive bus interrupt source 2			
	Receive bus interrupt source 3			
5	Transmit bus interrupt source 1	FFF5$_{16}$	FFF4$_{16}$	The condition which the transmit bus interrupt factor request bit becomes “1” is defined according to each communication protocol specification confirmation.
	Transmit bus interrupt source 2			
	Transmit bus interrupt source 3			
6	Timer X	FFF3$_{16}$	FFF2$_{16}$	
7	Timer Y	FFF1$_{16}$	FFF0$_{16}$	
8	Timer 2	FFEF$_{16}$	FFEE$_{16}$	
9	Timer 3	FFED$_{16}$	FFEC$_{16}$	
10	INT$_2$	FFEB$_{16}$	FFEA$_{16}$	External interrupt (active edge selectable)
11	Serial I/O3	FFE9$_{16}$	FFE8$_{16}$	Valid only when serial I/O3 is selected
	CNTR$_0$			External interrupt (active edge selectable)
12	CNTR$_1$	FFE7$_{16}$	FFE6$_{16}$	External interrupt (active edge selectable)
13	Timer 1	FFE5$_{16}$	FFE4$_{16}$	
14	INT$_3$	FFE3$_{16}$	FFE2$_{16}$	External interrupt (active edge selectable)
	INT$_4$			External interrupt (active edge selectable)
	INT$_5$			External interrupt (active edge selectable)
15	ADT	FFE1$_{16}$	FFE0$_{16}$	Valid only when ADT interrupt is selected External interrupt (falling valid)
	A-D converter			Valid when A-D converter interrupt is selected
	Serial I/O2			Valid only when serial I/O2 is selected
16	Key input (key-on wake-up)	FFDF$_{16}$	FFDE$_{16}$	External interrupt (falling valid)
	Serial I/O1 receive			Valid only when serial I/O1 is selected
	Serial I/O1 transmit			Valid only when serial I/O1 is selected
17	BRK instruction	FFDD$_{16}$	FFDC$_{16}$	Non-maskable software interrupt

Notes 1: Vector addresses contain interrupt jump destination addresses.

2: Reset functions in the same way as an interrupt with the highest priority.

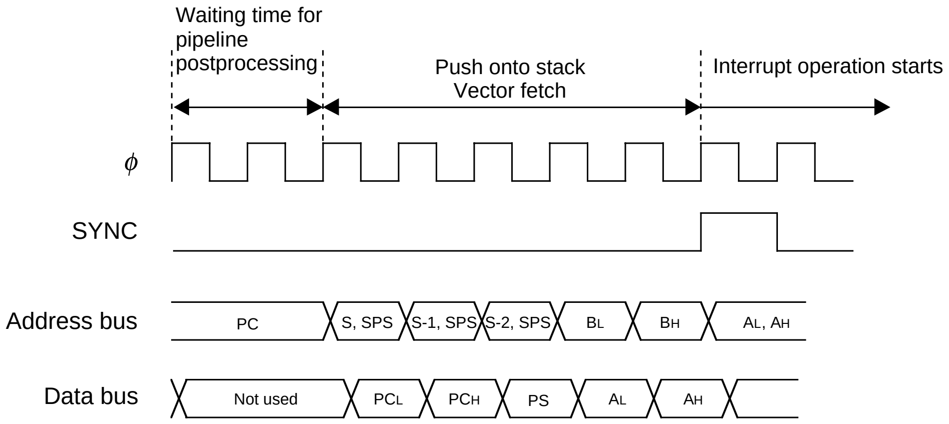
3: Either ADT interrupt or A-D converter interrupt can be used. Both ADT interrupt and A-D converter interrupt cannot be used.

Timing After Interrupt

For “one factor/one vector interrupt,” the CPU starts processing after interrupt acceptance at the next instruction execution timing (rising edge of SYNC signal) immediately after the interrupt request is generated. For “multiple factors/one vector interrupt,” the CPU starts processing after interrupt acceptance at the second instruction execution timing (rising edge of SYNC signal) after the interrupt request for interrupt factor discrimination is generated.

Figure 76 shows a timing chart after interrupt occurs, and Figure 77 shows the time up to execution of interrupt processing routine.

For details of interrupt request acceptance, refer to “INTERRUPTS” of “FUNCTIONAL DESCRIPTION.”



SYNC : CPU operation code fetch cycle
(This is an internal signal which cannot be observed from the external unit.)
B_L, B_H : Vector address of each interrupt
A_L, A_H : Jump destination address of each interrupt
SPS : “00$_{16}$” or “01$_{16}$”

Fig. 76 Timing chart after interrupt occurs

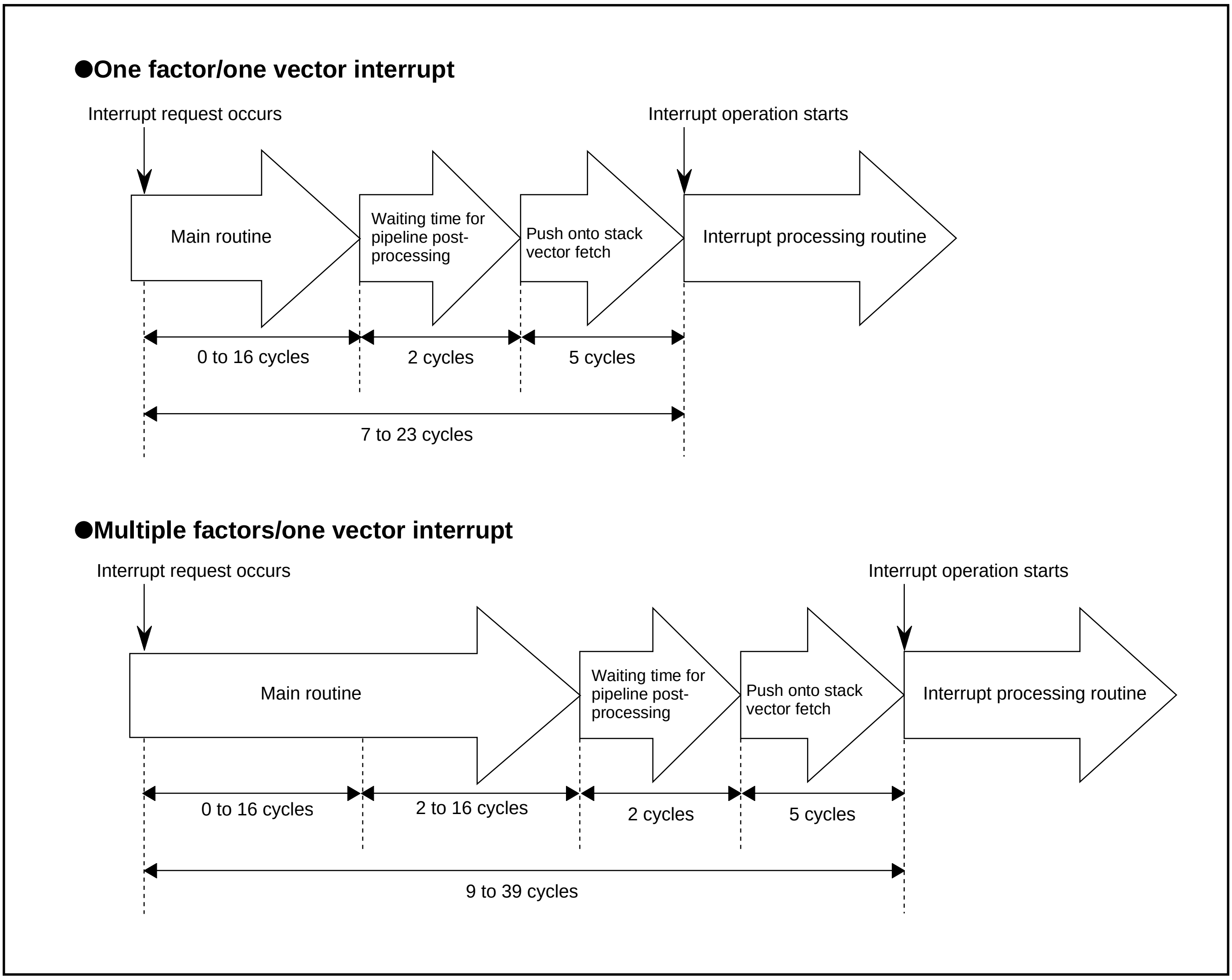

●One factor/one vector interrupt
Interrupt request occurs
Interrupt operation starts
Main routine
Waiting time for pipeline post-processing
Push onto stack vector fetch
Interrupt processing routine
0 to 16 cycles
2 cycles
5 cycles
7 to 23 cycles
●Multiple factors/one vector interrupt
Interrupt request occurs
Interrupt operation starts
Main routine
Waiting time for pipeline post-processing
Push onto stack vector fetch
Interrupt processing routine
0 to 16 cycles
2 to 16 cycles
2 cycles
5 cycles
9 to 39 cycles

Fig. 77 Time up to execution of interrupt processing routine

A-D Converter

A-D conversion is started by setting AD conversion completion bit to "0." During A-D conversion, internal operations are performed as follows.

1. After the start of A-D conversion, A-D conversion register goes to "00_{16}."
2. The highest-order bit of A-D conversion register is set to "1," and the comparison voltage Vref is input to the comparator. Then, V_{ref} is compared with analog input voltage V_{IN}.
3. As a result of comparison, when $V_{ref} < V_{IN}$, the highest-order bit of A-D conversion register becomes "1." When $V_{ref} > V_{IN}$, the highest-order bit becomes "0."

By repeating the above operations up to the lowest-order bit of the A-D conversion register, an analog value converts into a digital value. A-D conversion completes at 50 clock cycles after it is started, and the result of the conversion is stored into the A-D conversion register. Concurrently with the completion of A-D conversion, A-D conversion interrupt request occurs, so that the AD conversion interrupt request bit is set to "1."

●Relative formula for a reference voltage V_{REF} of A-D converter and V_{ref}

When n = 0 $\quad V_{ref} = 0$

When n = 1 to 255 $\quad V_{ref} = \dfrac{V_{REF}}{256} \times (n - 0.5)$

n : Value of A-D converter (decimal numeral)

Fig. 78 Relative formula for a reference voltage V_{REF} of A-D converter and V_{ref}

Table 12 Change of A-D conversion register during A-D conversion

	Change of A-D conversion register	Value of comparison voltage (V_{ref})
At start of conversion	0 0 0 0 0 0 0 0	0
First comparison	1 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{512}$
Second comparison	✻1 1 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{512}$
Third comparison	✻1 ✻2 1 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{512}$
	≈	≈
After completion of eighth comparison	A result of A-D conversion ✻1 ✻2 ✻3 ✻4 ✻5 ✻6 ✻7 ✻8	

✻1: A result of the first comparison **✻2:** A result of the second comparison
✻3: A result of the third comparison **✻4:** A result of the fourth comparison
✻5: A result of the fifth comparison **✻6:** A result of the sixth comparison
✻7: A result of the seventh comparison **✻8:** A result of the eighth comparison

FUNCTIONAL DESCRIPTION SUPPLEMENT

Figures 79 shows the A-D conversion equivalent circuit, and Figure 80 shows the A-D conversion timing chart.

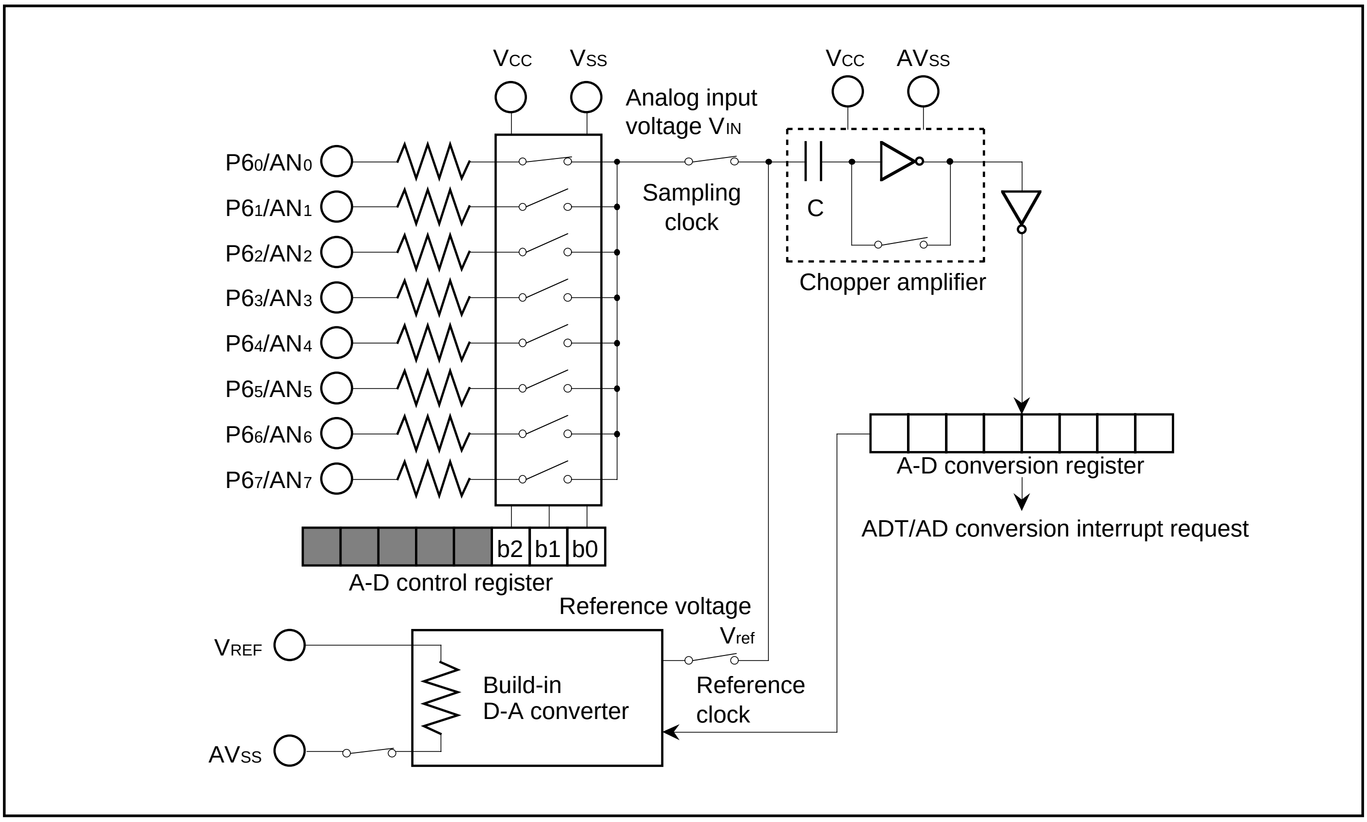


Fig. 79 A-D conversion equivalent circuit

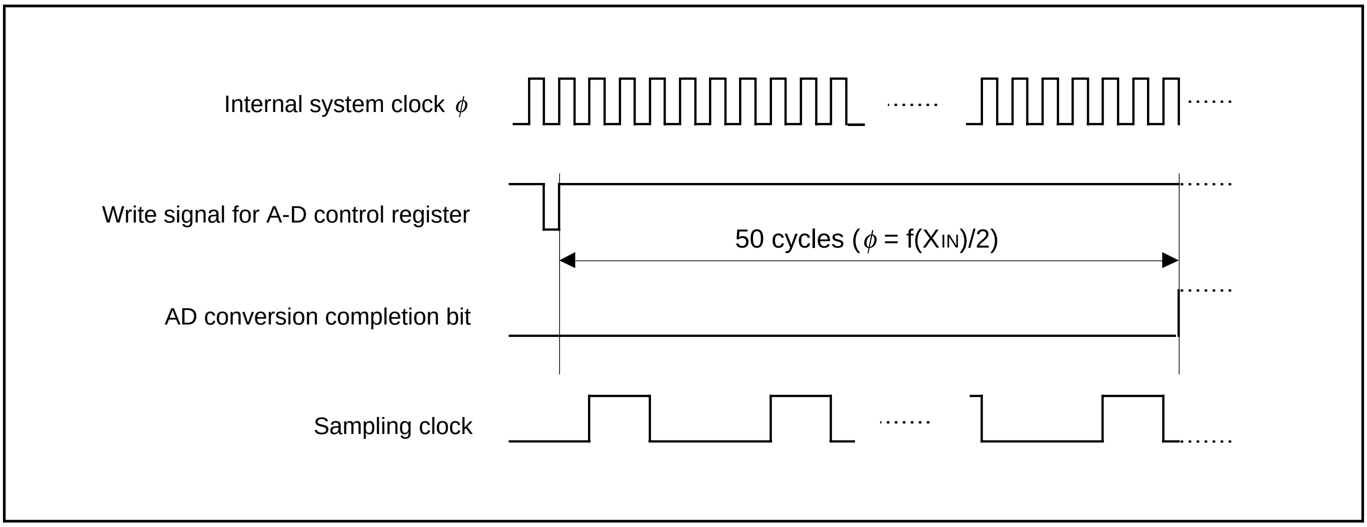

Internal system clock ϕ
Write signal for A-D control register
50 cycles (ϕ = f(XIN)/2)
AD conversion completion bit
Sampling clock

Fig. 80 A-D conversion timing chart

CHAPTER 2
APPLICATION

2.1 I/O port

2.1.1 Memory assignment

Figure 2.1.1 shows the memory assignment of I/O port relevant registers. Each of these registers is described below.

Address	
0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	Port P5 (P5)
$000B_{16}$	Port P5 direction register (P5D)
$000C_{16}$	Port P6 (P6)
$000D_{16}$	Port P6 direction register (P6D)
$000E_{16}$	Port P7 (P7)
$000F_{16}$	Port P7 direction register (P7D)
0010_{16}	Port P8 (P8)
0011_{16}	Port P8 direction register (P8D)
0012_{16}	Port P9 (P9)
	≈
0033_{16}	PULL UP register (PULLU)

Fig. 2.1.1 Memory assignment of I/O port relevant registers

2.1.2 Relevant registers

(1) I/O port write and read

■ **Input-only ports and programmable I/O ports set for input mode**

The value (pin state) input to the input-only ports and the programmable I/O ports set for the input mode is read by reading the port register corresponding to each port.

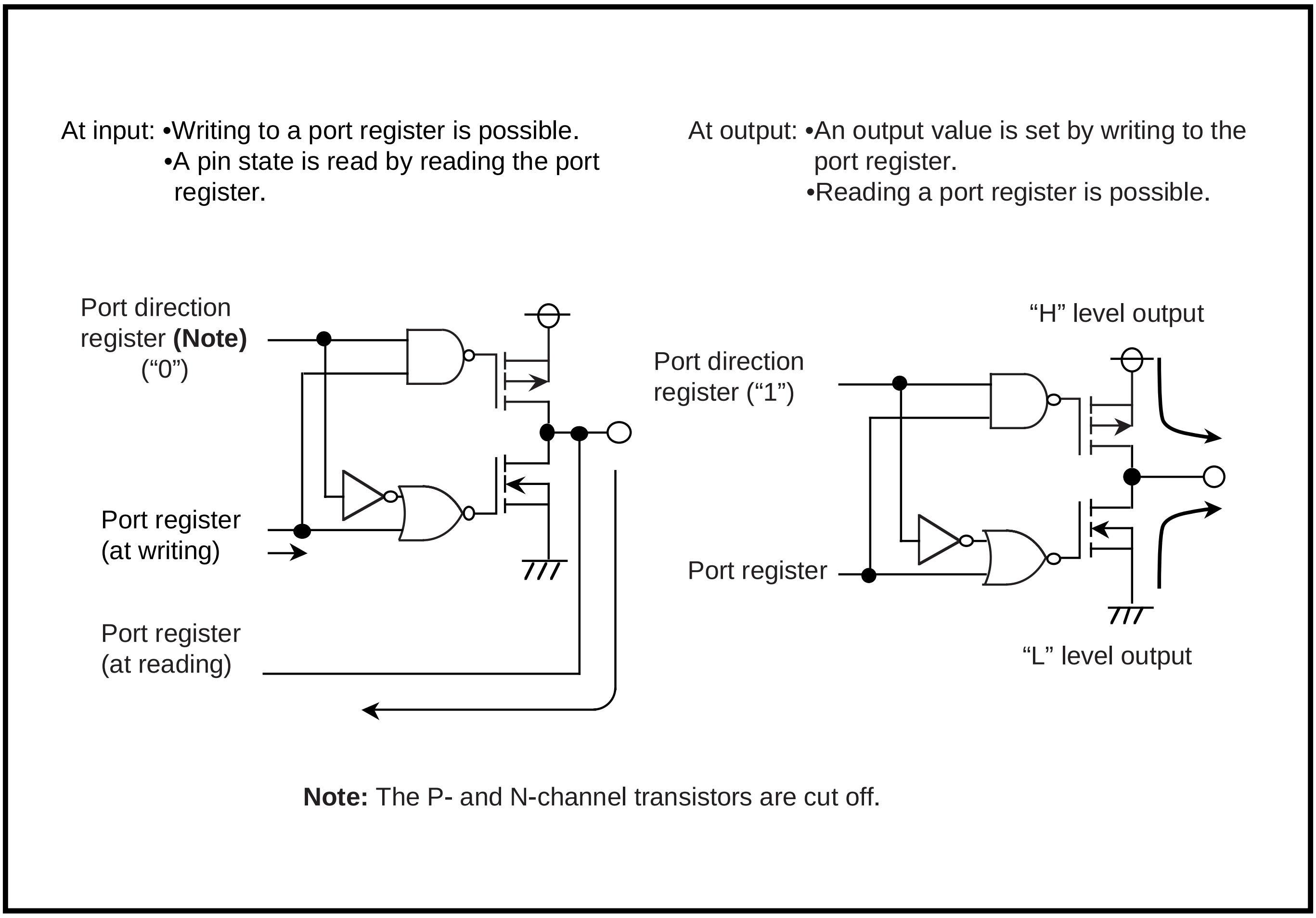
When writing data into the port register corresponding to each port, the data is only written to the port register, but the pin state is not affected.

■ **Programmable I/O ports set for output mode**

The value written to the port register corresponding to a programmable I/O set for the output mode is output externally through a transistor.

When reading the data of the port transistor corresponding to each port, the pin state is not read, but the value written to the port register is read.

Figure 2.1.2 shows the I/O port write and read.



At input: •Writing to a port register is possible.
•A pin state is read by reading the port register.

At output: •An output value is set by writing to the port register.
•Reading a port register is possible.

Port direction register (Note) (“0”)
Port register (at writing)
Port register (at reading)
Port direction register (“1”)
Port register
“H” level output
“L” level output

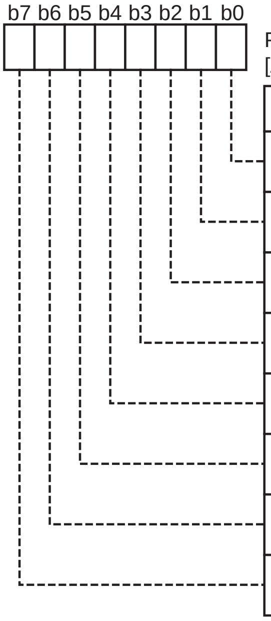
Note: The P- and N-channel transistors are cut off.

Fig. 2.1.2 I/O port write and read

(2) Input/output switching of programmable I/O ports

Input/output switching of the programmable I/O ports is performed by the port direction register corresponding to each port. Each direction register is initialized to “00_{16}” at reset, so that I/O ports are set for the input mode. Figure 2.1.3 shows the structure of the port Pi (i =0 to 8) direction register.

Port Pi direction register



Port Pi direction register (PiD) (i = 0 to 8)
[Addresses 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, $0F_{16}$, 11_{16}]

b	Name	Functions	At reset	R	W
0	Port Pi direction register	0 : Port Pi_0 input mode 1 : Port Pi_0 output mode	0	×	○
1		0 : Port Pi_1 input mode 1 : Port Pi_1 output mode	0	×	○
2		0 : Port Pi_2 input mode 1 : Port Pi_2 output mode	0	×	○
3		0 : Port Pi_3 input mode 1 : Port Pi_3 output mode	0	×	○
4		0 : Port Pi_4 input mode 1 : Port Pi_4 output mode	0	×	○
5		0 : Port Pi_5 input mode 1 : Port Pi_5 output mode	0	×	○
6		0 : Port Pi_6 input mode 1 : Port Pi_6 output mode	0	×	○
7		0 : Port Pi_7 input mode 1 : Port Pi_7 output mode	0	×	○

Fig. 2.1.3 Structure of port Pi (i = 0 to 8) direction register

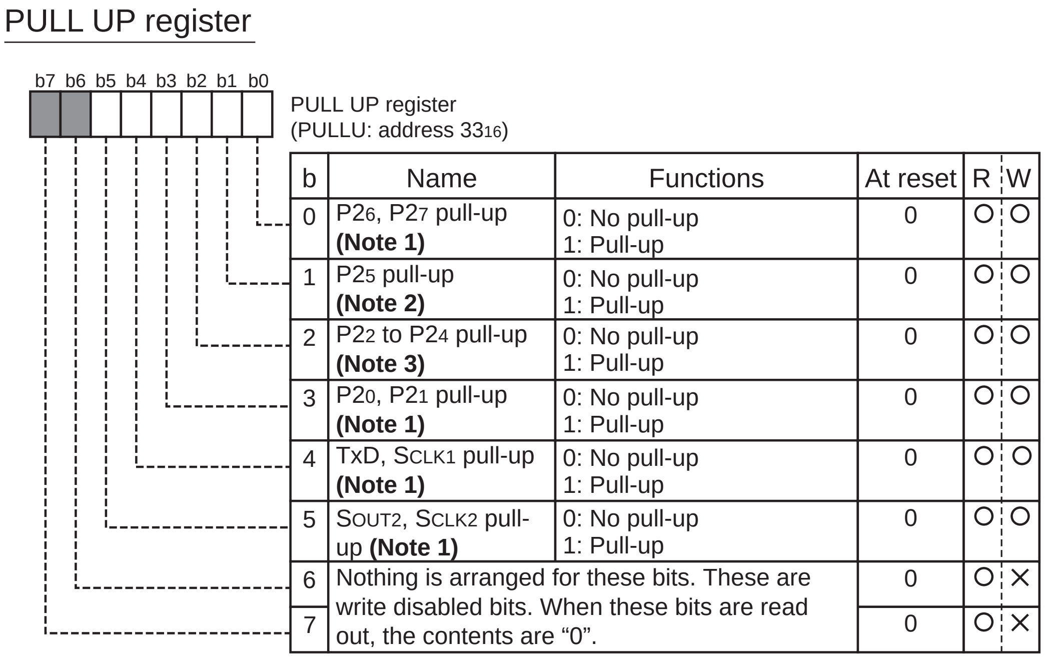
(3) Pull-up control

The pins shown in Table 2.1.1 are controlled for pull-up by software. The pull-up transistor becomes ON state by setting “1” to the corresponding bit of the PULL UP register, and pull-up becomes valid. Figure 2.1.4 shows the structure of the PULL UP register.

Table 2.1.1 Pins which pull-up is controlled by software

Pins	Input/Output set which pull-up is valid
$P2_0$–$P2_7$	At input
TxD, SCLK1, SOUT2, SCLK2	At output

PULL UP register



b	Name	Functions	At reset	R	W
0	$P2_6$, $P2_7$ pull-up **(Note 1)**	0: No pull-up 1: Pull-up	0	○	○
1	$P2_5$ pull-up **(Note 2)**	0: No pull-up 1: Pull-up	0	○	○
2	$P2_2$ to $P2_4$ pull-up **(Note 3)**	0: No pull-up 1: Pull-up	0	○	○
3	$P2_0$, $P2_1$ pull-up **(Note 1)**	0: No pull-up 1: Pull-up	0	○	○
4	TxD, SCLK1 pull-up **(Note 1)**	0: No pull-up 1: Pull-up	0	○	○
5	SOUT2, SCLK2 pull-up **(Note 1)**	0: No pull-up 1: Pull-up	0	○	○
6	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
7			0	○	×

Notes 1: This controls pull-up in a unit of 2 bits.
2: This controls pull-up in a unit of 1 bit.
3: This controls pull-up in a unit of 3 bits.

Fig. 2.1.4 Structure of PULL UP register

2.1.3 Notes on use

When using I/O ports, note the following.

(1) Modifying port latch of I/O port 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

The bit managing instructions are read-modify-write form instructions for reading and writing data by a byte unit. Accordingly, when these instructions are executed on a bit of the port latch of an I/O port, the following is executed to all bits of the port latch.

•As for bit which is set for input port:
The pin state is read in the CPU, and is written to this bit after bit managing.

•As for bit which is set for output port:
The bit value is read in the CPU, and is written to this bit after bit managing.

Note the following:

•Even when a port which is set as an output port is changed for an input port, its port latch holds the output data.

•As for a bit of which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its port latch contents.

✻1 Bit managing instructions : **SEB** and **CLB** instructions

(2) Reading port direction register

The value of the port direction register cannot be read. The following cannot be used:

- the data transfer instruction (**LDA**, etc.) when the index X mode flag (T) is “1”
- 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 instruction (bit managing instructions of **CLB** and **SEB**, operation instructions of **ROR**, etc.) to a direction register.

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

(3) Pull-up control

To pull-up ports by software, note the following.

•Ports $P2_0$–$P2_7$s’ pull-up is valid only at input, and TxD, S_{CLK1}, S_{OUT2} and S_{CLK2}s’ pull-up is valid only at output.

•Ports except port $P2_5$ cannot be pull-up in a unit of 1 bit. For example, when Port $P2_6$ (pull-up control in a unit of 2 bits) is set to pull-up, port $P2_7$ is set to pull-up, too.

(4) Notes in standby state

In stand-by state*[1] for low-power dissipation, do not make input levels of an input port and an I/O port “undefined”, especially for I/O ports of the N-channel open-drain.

Pull-up (connect the port to V_{CC}) or pull-down (connect the port to V_{SS}) 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**

Even when setting as an output port with its direction register, in the following state :

- P-channel......when the content of the port latch is “0”
- N-channel......when the content of the port latch is “1”

the transistor becomes the OFF state, which causes the ports to be the high-impedance state. Note that the level becomes “undefined” depending on external circuits.

Accordingly, the potential which is input to the input buffer in a microcomputer is unstable in the state that input levels of a input port and an I/O port are “undefined”. This may cause power source current.

*[1] stand-by state : the stop mode by executing the **STP** instruction
the wait mode by executing the **WIT** instruction

(5) Notes on external circuit design for I/O ports

When designing an external circuit for I/O ports, be sure to set the following items within the standard value range.

- Total average output current
- Peak output current
- Average output current

When performing multiple key-input operations by forming a key matrix, design in consideration of the port input current due to multiple key-input operations.

Figure 2.1.5 shows the example of external circuit design for I/O port.

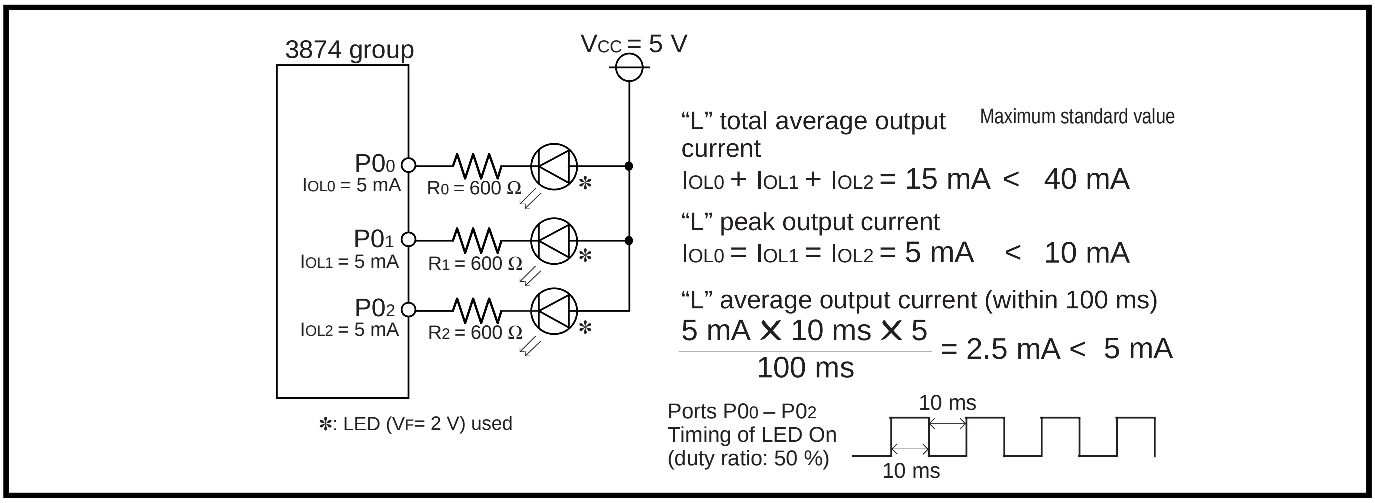


Fig. 2.1.5 Example of external circuit design for I/O port

(6) A-D converter power source pin

Pins AVcc and AVss are A-D converter power source pins. Regardless of using the A-D conversion function or not, connect them as following:

•AVss: Connect the Vss line

(7) Terminate unused pins

Table 2.1.2 shows the termination of unused pins.

Table 2.1.2 Termination of unused pins

Pins	Termination
P0, P1, P3, $P4_0$, $P4_1$, $P4_5$, $P4_7$, $P5_0$, $P5_6$, $P5_7$, P6, $P7_1$, $P7_3$ to $P7_5$, $P8_0$ to $P8_2$, $P8_7$ **(Note 1)**	•Set to the input mode, and then pull down ports through a resistor. **(Note 1)** •Set to the output mode and open at “L” or “H” output state.
P2, $P4_2$ to $P4_4$, $P4_6$, $P5_1$ to $P5_5$, $P7_0$, $P7_2$, $P7_6$, $P7_7$, $P8_3$ to $P8_6$ **(Note 1) (Note 2)**	
$P9_7$ **(Note 3)**	•Pull down ports through a resistor.
V_{REF}	•Connect to Vss (GND).
AVss	•Connect to Vss (GND).
X_{OUT}	•Open (only when using external clock)

Notes 1: For programmable I/O ports, do not connect two or more ports together through a resistor to Vss. For the pin which can control its pull up, make the pull-up invalid.

2: When setting the schmidt input pin such as the $P4_4$/serial I/O1 input pin etc. to the output mode and leave them open, this pin remains set to the input mode, initial state, until this pin is switched to the output mode by software after reset. Consequently, voltage level of the pin is unstable, and a power source current may increase while the pin is set to the input mode. For the effect to system, the user must evaluate the system sufficiently .

3: In One Time PROM version and EPROM version, the $P9_7$ pin also functions as the V_{PP} pin (for EPROM write).

•Make the length of wiring to the V_{PP} pin as short as possible.

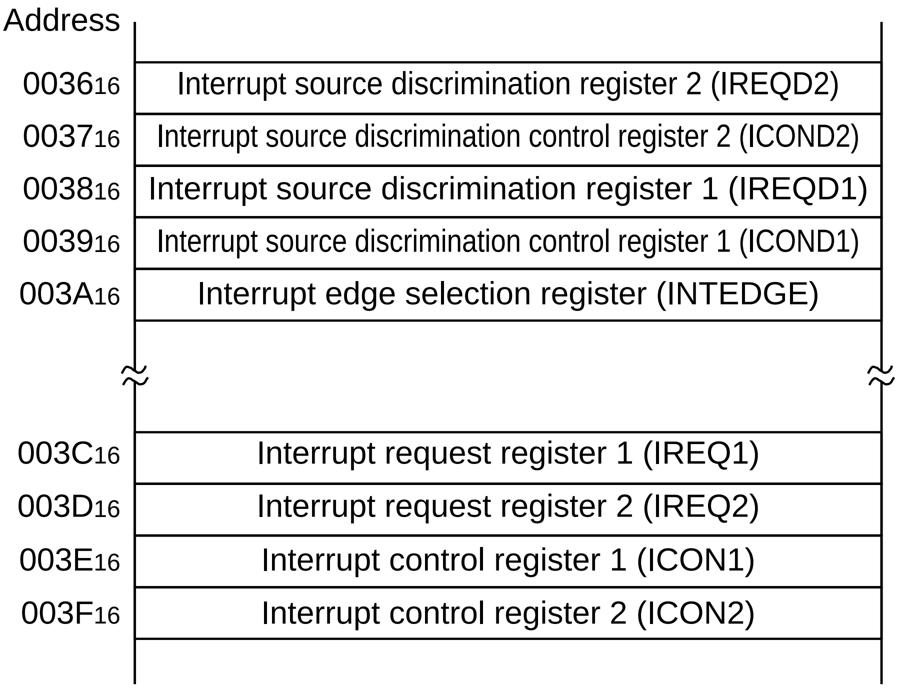
•Connect an approximately 5 kΩ resistor to the V_{PP} pin as close as possible and connect the V_{PP} pin and the Vss pin via that resistor.

4: In programmable I/O ports, the value of the port direction register may be changed and become the output mode by noise or program runaway. Then, a port and GND might cause short circuit as the worst case. Therefore, we recommend to set the value into the port direction register by software regularly.

2.2 Interrupts

2.2.1 Memory assignment

Figure 2.2.1 shows the memory assignment of interrupt relevent registers. Each of these registers is described below. For interrupt vector addresses, refer to **“Table 9”** of **“CHAPTER 1. HARDWARE”**.



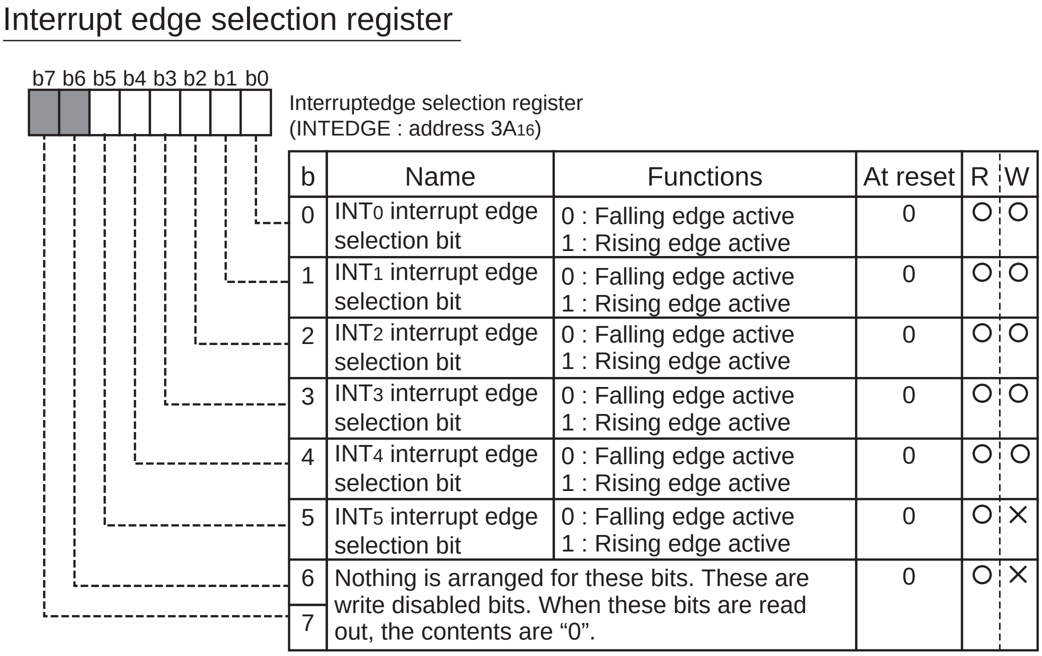
Address	
0036_{16}	Interrupt source discrimination register 2 (IREQD2)
0037_{16}	Interrupt source discrimination control register 2 (ICOND2)
0038_{16}	Interrupt source discrimination register 1 (IREQD1)
0039_{16}	Interrupt source discrimination control register 1 (ICOND1)
$003A_{16}$	Interrupt edge selection register (INTEDGE)
≈	≈
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003D_{16}$	Interrupt request register 2 (IREQ2)
$003E_{16}$	Interrupt control register 1 (ICON1)
$003F_{16}$	Interrupt control register 2 (ICON2)

Fig. 2.2.1 Memory assignment of interrupt relevant registers

2.2.2 Relevant registers

(1) Interrupt edge selection register

The interrupt edge selection register selects an active edge of each INT interrupt. Figure 2.2.2 shows the structure of the interrupt edge selection register.



Interrupt edge selection register

Interruptedge selection register
(INTEDGE : address $3A_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT_1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	INT_2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	INT_3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4	INT_4 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
5	INT_5 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	×
6	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
7					

Fig. 2.2.2 Structure of interrupt edge selection register

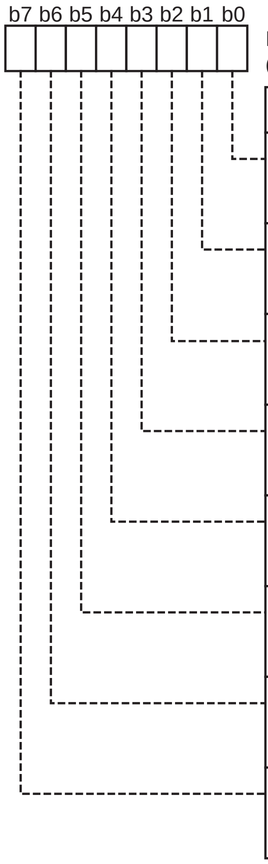
(2) Interrupt request register 1 and interrupt request register 2

The interrupt request register 1 (address 3C$_{16}$) and the interrupt request register 2 (address 3D$_{16}$) indicate whether an interrupt request for each vector has occurred or not.

The interrupt control circuit consists of two types of interrupts: “multiple factors/one vector interrupt (two types or more interrupt sources are assigned to the same interrupt vector)” and “one factor/one vector interrupt”. The interrupt request bit for each vector of “multiple factors/one vector interrupt” is set to “1” when the interrupt disable flag (I) is “0” and the corresponding factor interrupt enable bit is “1” and one of the requests of the interrupt sources assigned to its vector occurs.

Figure 2.2.3 shows the structure of the interrupt request register 1, Figure 2.2.4 shows the structure of the interrupt request register 2.

Interrupt request register 1



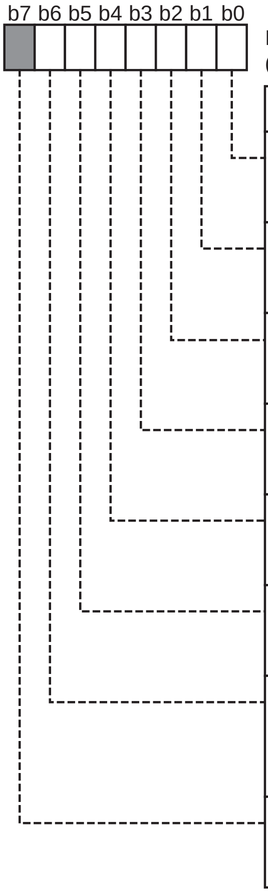
Interrupt request register 1
(IREQ1 : address 3C$_{16}$)

b	Name	Functions	At reset	R	W
0	INT$_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT$_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Receive bus interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Transmit bus interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: “0” can be set by software, but “1” cannot be set.

Fig. 2.2.3 Structure of interrupt request register 1

Interrupt request register 2



Interrupt request register 2
(IREQ2 : address $3D_{16}$)

b	Name	Functions	At reset	R	W
0	INT_2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	$CNTR_0$, serial I/O3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	$CNTR_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3, INT_4, INT_5 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	ADT/A-D converter, serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Key input, serial I/O1 receive, serial I/O1 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is write disabled bit. When this bit is read out, the contents is “0”.		0	○	×

✻: “0” can be set by software, but “1” cannot be set.

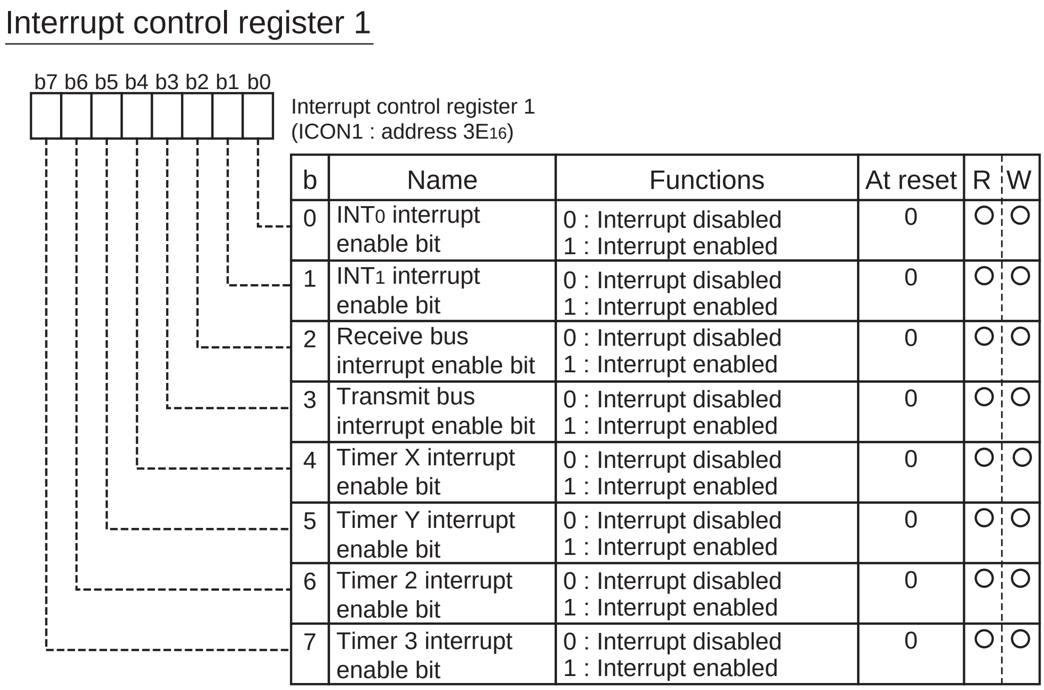
Fig. 2.2.4 Structure of interrupt request register 2

(3) Interrupt control register 1 and interrupt control register 2

The interrupt control register 1 (address $3E_{16}$) and the interrupt control register 2 (address $3F_{16}$) control the interrupt request for each vector.

When the interrupt request bit for each vector and the interrupt enable bit for each vector are both "1", the interrupt is accepted.

Figure 2.2.5 shows the structure of the interrupt control register 1 and Figure 2.2.6 shows the structure of the interrupt control register 2.

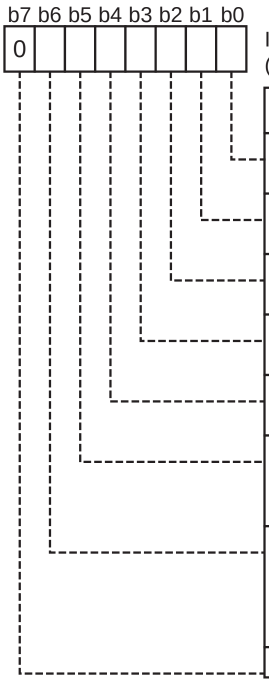


Interrupt control register 1
(ICON1 : address $3E_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Receive bus interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Transmit bus interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.2.5 Structure of interrupt request register 1

Interrupt control register 2



Interrupt control register 2
(ICON2 : address 3F$_{16}$)

b	Name	Functions	At reset	R	W
0	INT$_2$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
1	CNTR$_0$, serial I/O3 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
2	CNTR$_1$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer 1 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT$_3$, INT$_4$, INT$_5$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
5	ADT/A-D converter, serial I/O2 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
6	Key input, serial I/O1 receive, serial I/O1 transmit interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix “0” to this bit.		0	○	○

Fig. 2.2.6 Structure of interrupt control register 2

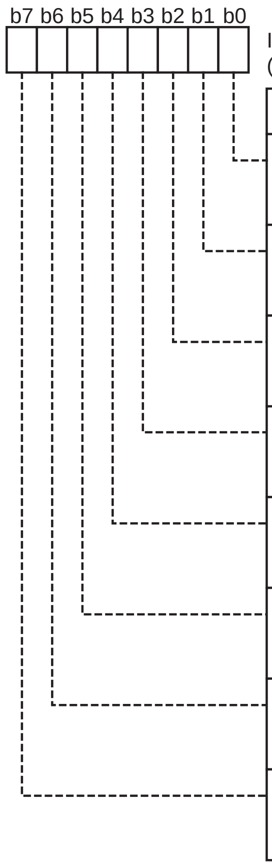
(4) Interrupt source discrimination register 1 and interrupt source discrimination register 2

The interrupt source discrimination register 1 (address 38$_{16}$) and the interrupt source discrimination register 2 (address 36$_{16}$) indicate whether an interrupt request for each interrupt factor of “multiple factors/one vector interrupt” has occurred or not.

The priority of two or more interrupt sources of “multiple factors/one vector interrupt” are same, so that confirm each bit of the interrupt source discrimination register 1 and the interrupt source discrimination register 2, and process according to the priority by software.

Figure 2.2.7 shows the structure of the interrupt source discrimination register 1 and Figure 2.2.8 shows the structure of the interrupt source discrimination register 2.

Interrupt source discrimination register 1



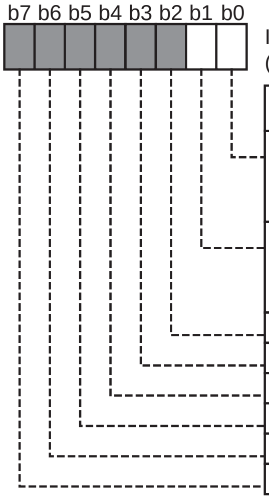
Interrupt source discrimination register 1
(IREQD1 : address 38$_{16}$)

b	Name	Functions	At reset	R	W
0	INT$_3$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT$_4$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	INT$_5$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Serial I/O1 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key input interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	ADT/A-D converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻ : “0” can be set by software, but “1” cannot be set.

Fig. 2.2.7 Structure of interrupt source discrimination register 1

Interrupt source discrimination register 2



Interrupt source discrimination register 2
(IREQD2 : address 36_{16})

b	Name	Functions	At reset	R	W
0	CNTR$_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Serial I/O3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

✻: “0” can be set by software, but “1” cannot be set.

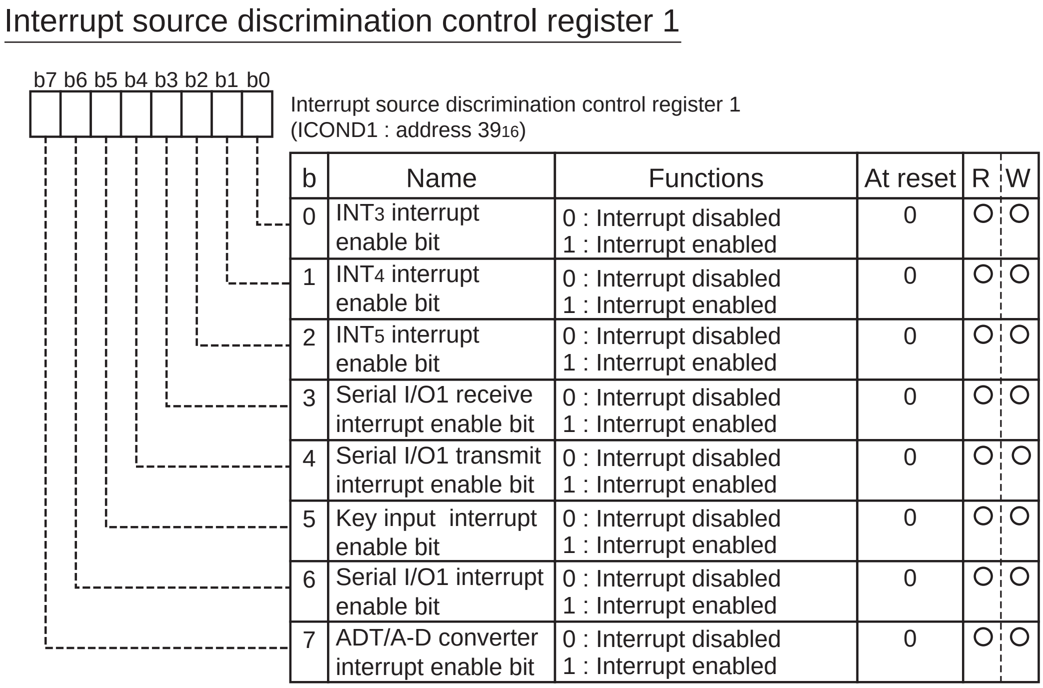
Fig. 2.2.8 Structure of interrupt source discrimination register 2

(5) Interrupt source discrimination control register 1 and interrupt source discrimination control register 2

The interrupt source discrimination control register 1 and the interrupt source discrimination control register 2 control acceptance of interrupts for each interrupt source of “multiple factors/one vector interrupt”.

When the interrupt enable bit for each vector and the interrupt request bit for each vector are both “1”, the interrupt is accepted.

Figure 2.2.9 shows the structure of the interrupt source discrimination control register 1 and Figure 2.2.10 shows the structure of the interrupt source discrimination control register 2.

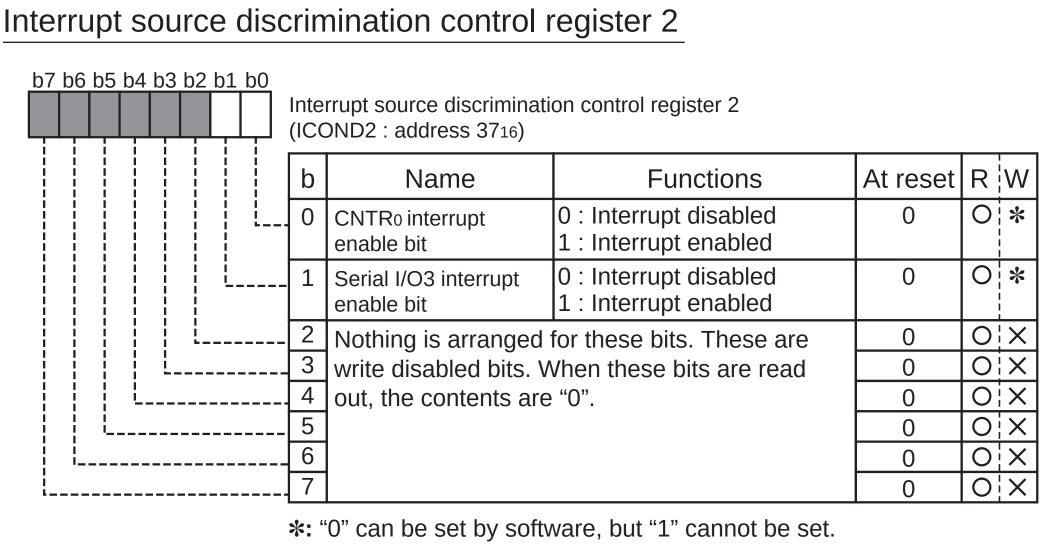


Interrupt source discrimination control register 1

Interrupt source discrimination control register 1
(ICOND1 : address 39_{16})

b	Name	Functions	At reset	R	W
0	INT3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT5 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Serial I/O1 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key input interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Serial I/O1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	ADT/A-D converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.2.9 Structure of interrupt source discrimination control register 1



Interrupt source discrimination control register 2

Interrupt source discrimination control register 2
(ICOND2 : address 37_{16})

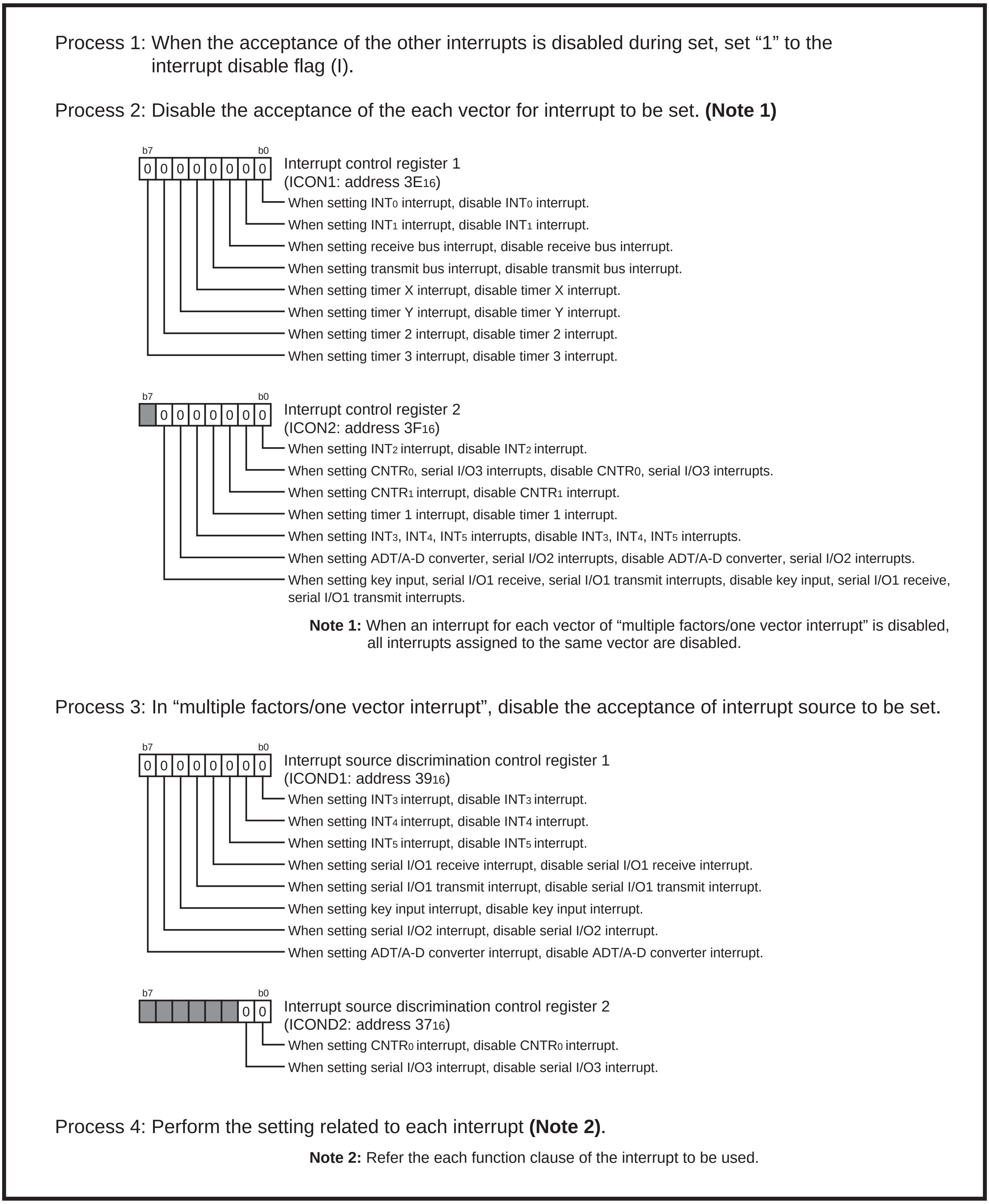
b	Name	Functions	At reset	R	W
0	CNTR$_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	✻
1	Serial I/O3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	✻
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

✻: “0” can be set by software, but “1” cannot be set.

Fig. 2.2.10 Structure of interrupt source discrimination control register 2

2.2.3 Interrupt setting method

Figures 2.2.11 to 2.2.13 show the interrupt setting method.



Process 1: When the acceptance of the other interrupts is disabled during set, set “1” to the interrupt disable flag (I).

Process 2: Disable the acceptance of the each vector for interrupt to be set. **(Note 1)**

Interrupt control register 1
(ICON1: address 3E$_{16}$)

- When setting INT$_0$ interrupt, disable INT$_0$ interrupt.
- When setting INT$_1$ interrupt, disable INT$_1$ interrupt.
- When setting receive bus interrupt, disable receive bus interrupt.
- When setting transmit bus interrupt, disable transmit bus interrupt.
- When setting timer X interrupt, disable timer X interrupt.
- When setting timer Y interrupt, disable timer Y interrupt.
- When setting timer 2 interrupt, disable timer 2 interrupt.
- When setting timer 3 interrupt, disable timer 3 interrupt.

Interrupt control register 2
(ICON2: address 3F$_{16}$)

- When setting INT$_2$ interrupt, disable INT$_2$ interrupt.
- When setting CNTR$_0$, serial I/O3 interrupts, disable CNTR$_0$, serial I/O3 interrupts.
- When setting CNTR$_1$ interrupt, disable CNTR$_1$ interrupt.
- When setting timer 1 interrupt, disable timer 1 interrupt.
- When setting INT$_3$, INT$_4$, INT$_5$ interrupts, disable INT$_3$, INT$_4$, INT$_5$ interrupts.
- When setting ADT/A-D converter, serial I/O2 interrupts, disable ADT/A-D converter, serial I/O2 interrupts.
- When setting key input, serial I/O1 receive, serial I/O1 transmit interrupts, disable key input, serial I/O1 receive, serial I/O1 transmit interrupts.

Note 1: When an interrupt for each vector of “multiple factors/one vector interrupt” is disabled, all interrupts assigned to the same vector are disabled.

Process 3: In “multiple factors/one vector interrupt”, disable the acceptance of interrupt source to be set.

Interrupt source discrimination control register 1
(ICOND1: address 39$_{16}$)

- When setting INT$_3$ interrupt, disable INT$_3$ interrupt.
- When setting INT$_4$ interrupt, disable INT$_4$ interrupt.
- When setting INT$_5$ interrupt, disable INT$_5$ interrupt.
- When setting serial I/O1 receive interrupt, disable serial I/O1 receive interrupt.
- When setting serial I/O1 transmit interrupt, disable serial I/O1 transmit interrupt.
- When setting key input interrupt, disable key input interrupt.
- When setting serial I/O2 interrupt, disable serial I/O2 interrupt.
- When setting ADT/A-D converter interrupt, disable ADT/A-D converter interrupt.

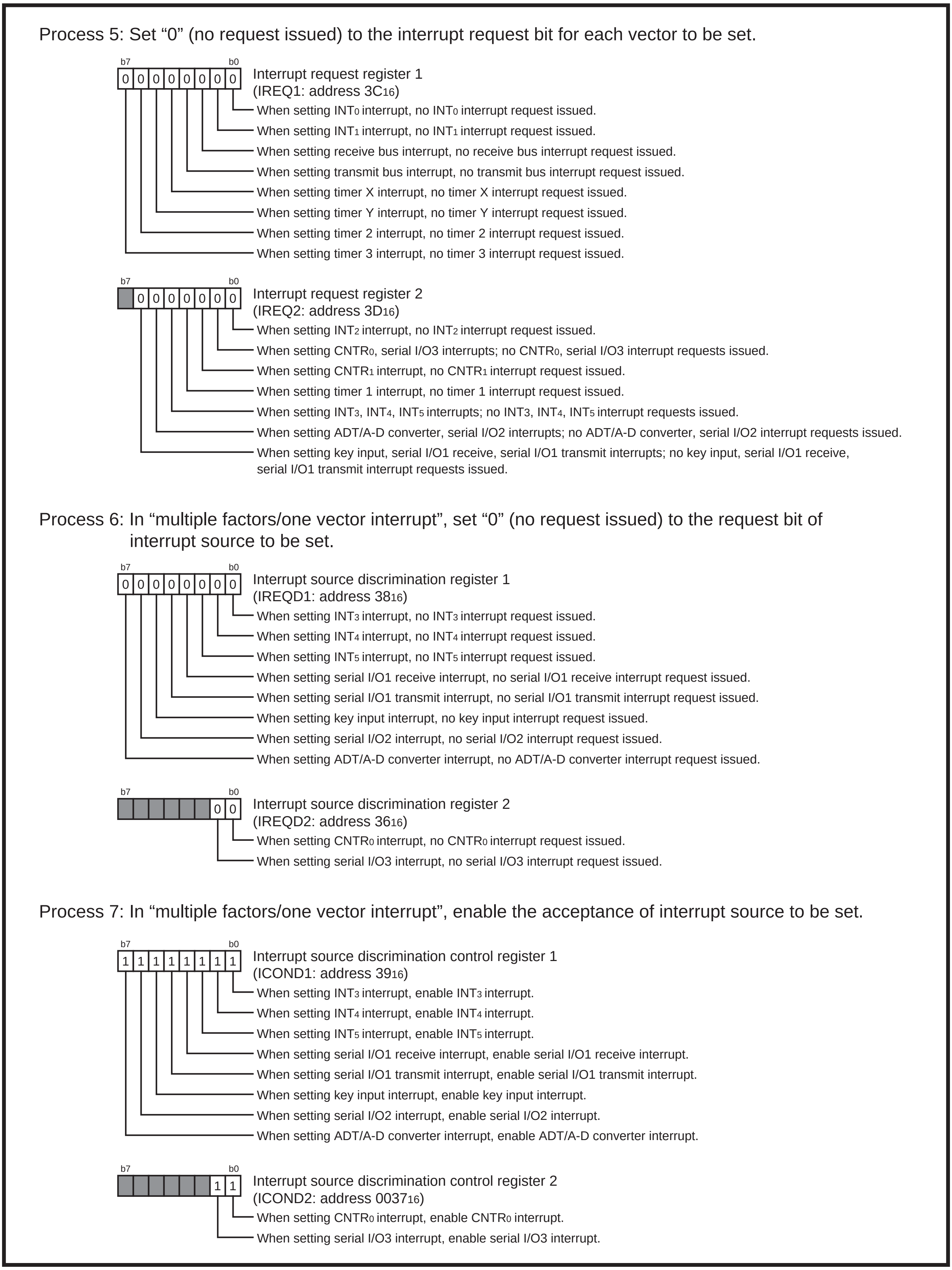
Interrupt source discrimination control register 2
(ICOND2: address 37$_{16}$)

- When setting CNTR$_0$ interrupt, disable CNTR$_0$ interrupt.
- When setting serial I/O3 interrupt, disable serial I/O3 interrupt.

Process 4: Perform the setting related to each interrupt **(Note 2)**.

Note 2: Refer the each function clause of the interrupt to be used.

Fig. 2.2.11 Interrupt setting method (1)



Process 5: Set “0” (no request issued) to the interrupt request bit for each vector to be set.

Interrupt request register 1
($IREQ1$: address $3C_{16}$)

- When setting INT_0 interrupt, no INT_0 interrupt request issued.
- When setting INT_1 interrupt, no INT_1 interrupt request issued.
- When setting receive bus interrupt, no receive bus interrupt request issued.
- When setting transmit bus interrupt, no transmit bus interrupt request issued.
- When setting timer X interrupt, no timer X interrupt request issued.
- When setting timer Y interrupt, no timer Y interrupt request issued.
- When setting timer 2 interrupt, no timer 2 interrupt request issued.
- When setting timer 3 interrupt, no timer 3 interrupt request issued.

Interrupt request register 2
($IREQ2$: address $3D_{16}$)

- When setting INT_2 interrupt, no INT_2 interrupt request issued.
- When setting $CNTR_0$, serial I/O3 interrupts; no $CNTR_0$, serial I/O3 interrupt requests issued.
- When setting $CNTR_1$ interrupt, no $CNTR_1$ interrupt request issued.
- When setting timer 1 interrupt, no timer 1 interrupt request issued.
- When setting INT_3, INT_4, INT_5 interrupts; no INT_3, INT_4, INT_5 interrupt requests issued.
- When setting ADT/A-D converter, serial I/O2 interrupts; no ADT/A-D converter, serial I/O2 interrupt requests issued.
- When setting key input, serial I/O1 receive, serial I/O1 transmit interrupts; no key input, serial I/O1 receive, serial I/O1 transmit interrupt requests issued.

Process 6: In “multiple factors/one vector interrupt”, set “0” (no request issued) to the request bit of interrupt source to be set.

Interrupt source discrimination register 1
($IREQD1$: address 38_{16})

- When setting INT_3 interrupt, no INT_3 interrupt request issued.
- When setting INT_4 interrupt, no INT_4 interrupt request issued.
- When setting INT_5 interrupt, no INT_5 interrupt request issued.
- When setting serial I/O1 receive interrupt, no serial I/O1 receive interrupt request issued.
- When setting serial I/O1 transmit interrupt, no serial I/O1 transmit interrupt request issued.
- When setting key input interrupt, no key input interrupt request issued.
- When setting serial I/O2 interrupt, no serial I/O2 interrupt request issued.
- When setting ADT/A-D converter interrupt, no ADT/A-D converter interrupt request issued.

Interrupt source discrimination register 2
($IREQD2$: address 36_{16})

- When setting $CNTR_0$ interrupt, no $CNTR_0$ interrupt request issued.
- When setting serial I/O3 interrupt, no serial I/O3 interrupt request issued.

Process 7: In “multiple factors/one vector interrupt”, enable the acceptance of interrupt source to be set.

Interrupt source discrimination control register 1
($ICOND1$: address 39_{16})

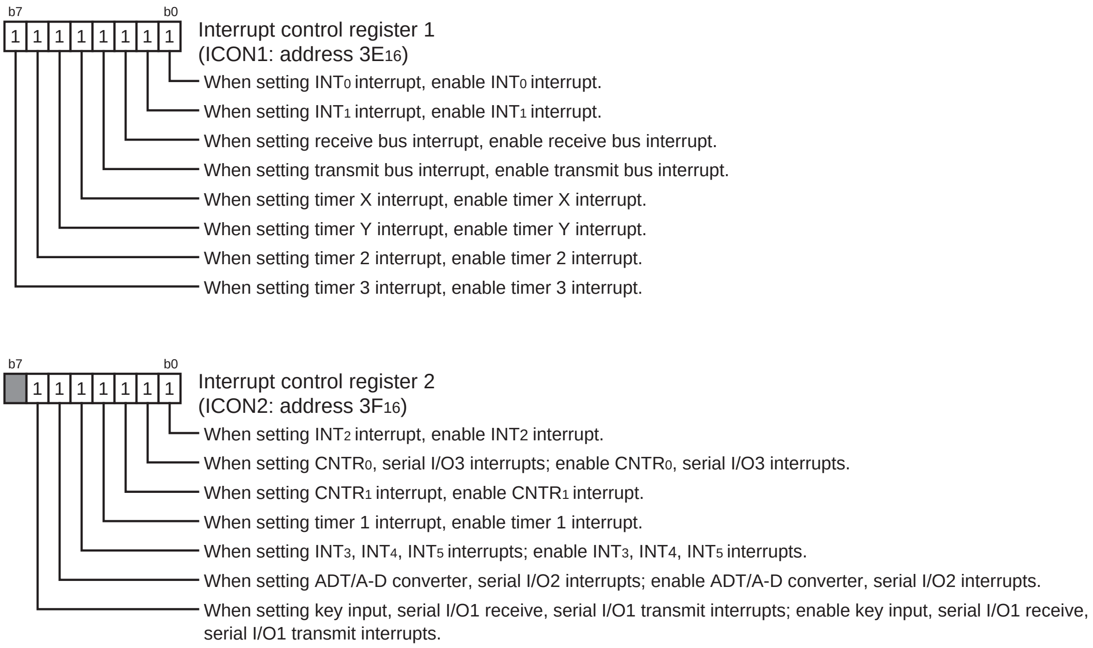
- When setting INT_3 interrupt, enable INT_3 interrupt.
- When setting INT_4 interrupt, enable INT_4 interrupt.
- When setting INT_5 interrupt, enable INT_5 interrupt.
- When setting serial I/O1 receive interrupt, enable serial I/O1 receive interrupt.
- When setting serial I/O1 transmit interrupt, enable serial I/O1 transmit interrupt.
- When setting key input interrupt, enable key input interrupt.
- When setting serial I/O2 interrupt, enable serial I/O2 interrupt.
- When setting ADT/A-D converter interrupt, enable ADT/A-D converter interrupt.

Interrupt source discrimination control register 2
($ICOND2$: address 0037_{16})

- When setting $CNTR_0$ interrupt, enable $CNTR_0$ interrupt.
- When setting serial I/O3 interrupt, enable serial I/O3 interrupt.

Fig. 2.2.12 Interrupt setting method (2)

Process 8: Enable the acceptance of each vector for interrupt to be set.


b7
b0
1 1 1 1 1 1 1 1
Interrupt control register 1
(ICON1: address 3E$_{16}$)
When setting INT$_0$ interrupt, enable INT$_0$ interrupt.
When setting INT$_1$ interrupt, enable INT$_1$ interrupt.
When setting receive bus interrupt, enable receive bus interrupt.
When setting transmit bus interrupt, enable transmit bus interrupt.
When setting timer X interrupt, enable timer X interrupt.
When setting timer Y interrupt, enable timer Y interrupt.
When setting timer 2 interrupt, enable timer 2 interrupt.
When setting timer 3 interrupt, enable timer 3 interrupt.
b7
b0
1 1 1 1 1 1 1
Interrupt control register 2
(ICON2: address 3F$_{16}$)
When setting INT$_2$ interrupt, enable INT$_2$ interrupt.
When setting CNTR$_0$, serial I/O3 interrupts; enable CNTR$_0$, serial I/O3 interrupts.
When setting CNTR$_1$ interrupt, enable CNTR$_1$ interrupt.
When setting timer 1 interrupt, enable timer 1 interrupt.
When setting INT$_3$, INT$_4$, INT$_5$ interrupts; enable INT$_3$, INT$_4$, INT$_5$ interrupts.
When setting ADT/A-D converter, serial I/O2 interrupts; enable ADT/A-D converter, serial I/O2 interrupts.
When setting key input, serial I/O1 receive, serial I/O1 transmit interrupts; enable key input, serial I/O1 receive, serial I/O1 transmit interrupts.

Process 9: When the interrupt disable flag (I) is set to “1” in “process 1”, set “0” to the interrupt disable flag (I).

Process 10: Operate the functions related to each interrupt. **(Note 3)**

Note 3: Refer the each function clause of the interrupt to be used.

Fig. 2.2.13 Interrupt setting method (3)

2.2.4 Key input interrupt

The key input interrupt request occurs when an “L” level voltage is applied to the pin of port P2 set to the input mode.
For interrupt sources except the key input interrupt, refer to **“CHAPTER 1. HARDWARE”**.

(1) Connection example using key input interrupt

When using the key input interrupt, compose a key-matrix where “L” level of port P2 input is valid. For a connection example when using the key input interrupt and port P2 block diagram, refer to “**Figure 21**” of “**CHAPTER 1. HARDWARE**”. In the connection example of Figure 21, a key input interrupt request occurs by pressing the key corresponding to one of ports $P2_0$ to $P2_4$.

(2) Key input interrupt setting method

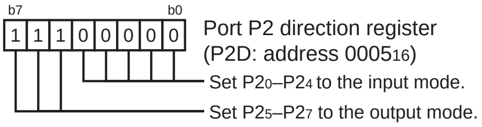
Figure 2.2.14 shows the setting example of key input interrupt relevant registers (corresponding to “**Figure 21**” of “**CHAPTER 1. HARDWARE**”).

Process 1: When the acceptance of the other interrupts is disabled, set “1” to the interrupt disable flag (I).

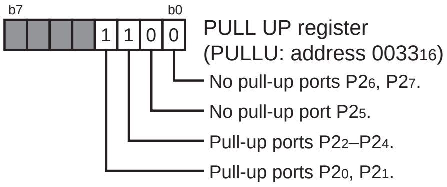
Process 2: Disable the acceptance of the key input interrupt. **(Note)**

Process 3: Perform the setting relevant to the key input interrupt.

- Set the input/output mode of the port P2 direction register.


b7
b0
1 1 1 0 0 0 0 0
Port P2 direction register
(P2D: address 0005_{16})
Set $P2_0$–$P2_4$ to the input mode.
Set $P2_5$–$P2_7$ to the output mode.

- Pull-up the pin to be used as key input.


b7
b0
1 1 0 0
PULL UP register
(PULLU: address 0033_{16})
No pull-up ports $P2_6$, $P2_7$.
No pull-up port $P2_5$.
Pull-up ports $P2_2$–$P2_4$.
Pull-up ports $P2_0$, $P2_1$.

Process 4: Set “0” (no requested) to the key input interrupt request bit. **(Note)**

Process 5: Enable the acceptance of the key input interrupt. **(Note)**

Process 6: When the interrupt disable flag (I) is set to “1” in “process 1”, set “0” to the interrupt disable flag (I).

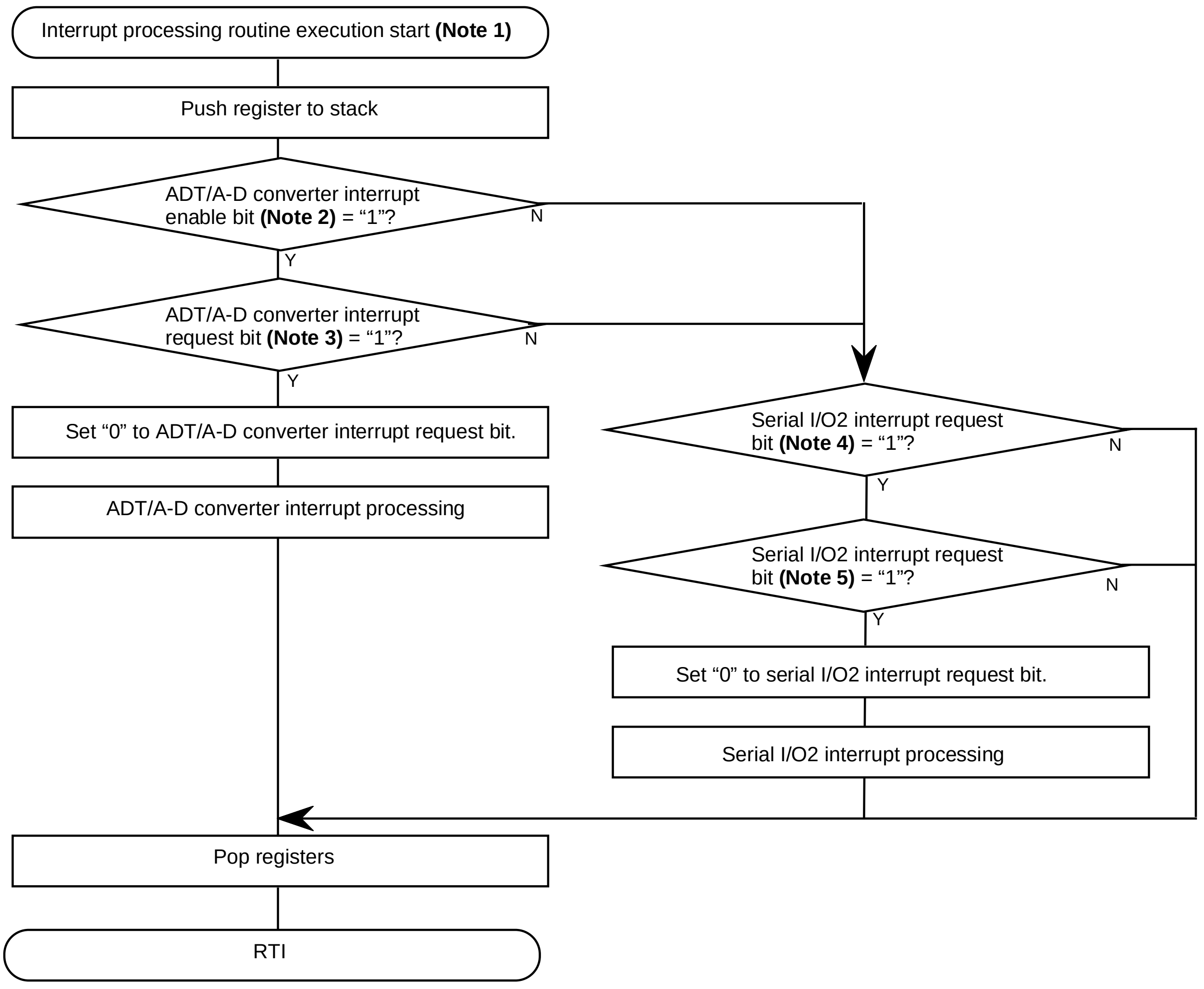
Note: Refer to section “**2.2.3 Interrupt setting method**”.

Fig. 2.2.14 Setting example of key input interrupt relevant registers (corresponding to “Figure 21” of “CHAPTER 1. HARDWARE”)

2.2.5 Management example of multiple factors/one vector interrupt

In “multiple factors/one vector interrupt”, two types or more interrupts have the same interrupt vector address, so that the priority on hardware is the same. While two types or more interrupts are enabled, judge the generated interrupt source by confirming the interrupt source discrimination registers 1 and 2 in the interrupt processing routine.

Figure 2.2.15 shows the interrupt request bit management example when the ADT/A-D converter interrupt and the serial I/O2 interrupt is used.


Interrupt processing routine execution start (Note 1)
Push register to stack
ADT/A-D converter interrupt enable bit (Note 2) = “1”?
N
Y
ADT/A-D converter interrupt request bit (Note 3) = “1”?
N
Y
Set “0” to ADT/A-D converter interrupt request bit.
ADT/A-D converter interrupt processing
Serial I/O2 interrupt request bit (Note 4) = “1”?
N
Y
Serial I/O2 interrupt request bit (Note 5) = “1”?
N
Y
Set “0” to serial I/O2 interrupt request bit.
Serial I/O2 interrupt processing
Pop registers
RTI

Notes 1: The processing of ADT/A-D converter interrupt has more high priority than the serial I/O2 interrupt’s.
2: Bit 7 of interrupt source discrimination control register 1 (address 39_{16})
3: Bit 7 of interrupt source discrimination register 1 (address 38_{16})
4: Bit 6 of interrupt source discrimination control register 1 (address 39_{16})
5: Bit 6 of interrupt source discrimination register 1 (address 38_{16})

Fig. 2.2.15 Interrupt request bit management example

2.2.6 Notes on use

(1) Switching external interrupt detection edge

For the products able to switch the external interrupt detection edge, switch it as the following sequence.

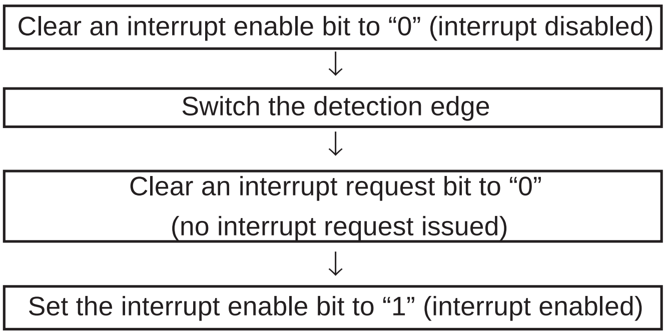

Clear an interrupt enable bit to “0” (interrupt disabled)
Switch the detection edge
Clear an interrupt request bit to “0” (no interrupt request issued)
Set the interrupt enable bit to “1” (interrupt enabled)

Fig. 2.2.16 Sequence of switch detection edge

■ **Reason**

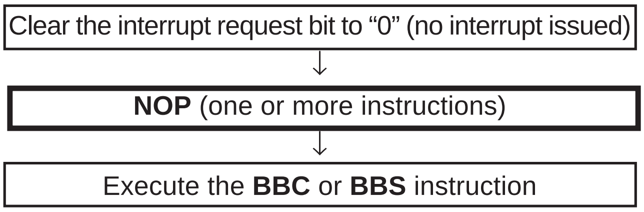
The interrupt circuit recognizes the switching of the detection edge as the change of external input signals. This may cause an unnecessary interrupt.

(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” by using a data transfer instruction, execute one or more instructions before executing the **BBC** or **BBS** instruction.

■ **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.


Clear the interrupt request bit to “0” (no interrupt issued)
NOP (one or more instructions)
Execute the **BBC** or **BBS** instruction

Data transfer instruction:
LDM, LDA, STA, STX, and STY instructions

Fig. 2.2.17 Sequence of check of interrupt request bit

● The request bits of interrupt source discrimination registers are not automatically cleared when an interrupt occurs. After an interrupt source has been discriminated, and before execution of the RTI or CLI instruction; set “0” to the request bits of interrupt source discrimination registers by the user’s program.

(3) Setting of interrupt source discrimination registers

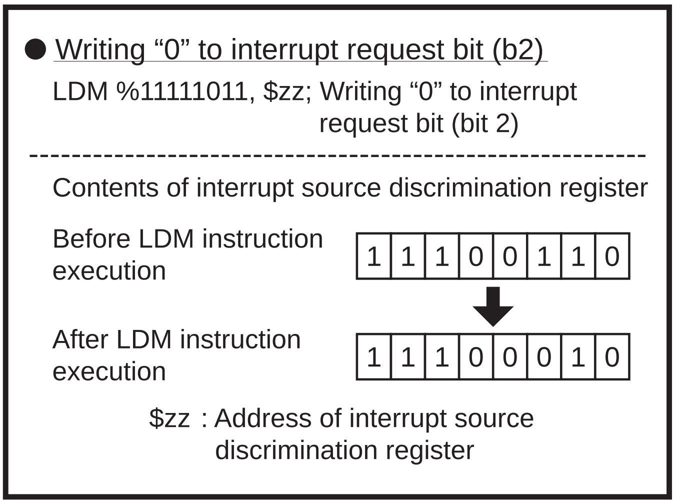
When “1” is written (or transferred) to an interrupt request bit, previous data is kept. Use the data transfer instructions[✻1] to set “0” to the interrupt request bit of the interrupt source discrimination register without using the bit managing instructions[✻2].

✻1 Data transfer instructions: LDM, LDA, STA, STX, and STY instructions
✻2 Bit managing instruction: CLB instruction

● **Reason**

If an interrupt request occurs at the same timing that the contents of the interrupt source discrimination register is modified by a bit managing instruction, the interrupt request bits of this register may not be set correctly.

Figure 2.2.18 shows an example of writing data to interrupt source discrimination register.


● Writing “0” to interrupt request bit (b2)

LDM %11111011, $zz; Writing “0” to interrupt request bit (bit 2)

Contents of interrupt source discrimination register

Before LDM instruction execution	1	1	1	0	0	1	1	0
After LDM instruction execution	1	1	1	0	0	0	1	0

$zz : Address of interrupt source discrimination register

Fig. 2.2.18 Example of writing data to interrupt source discrimination register

(4) Structure of interrupt control register 2

Fix the bit 7 of the interrupt control register 2 to “0”. Figure 2.2.19 shows the structure of the interrupt control register 2.

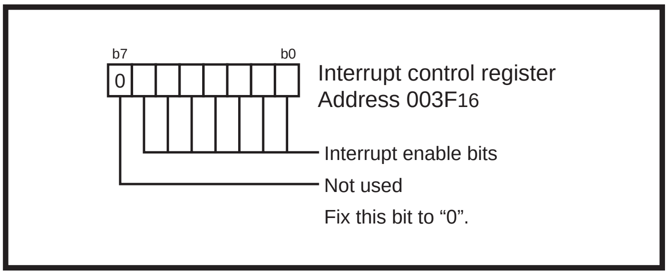


Fig. 2.2.19 Structure of interrupt control register 2

(5) Interrupt occurrence timing assigned to interrupt source discrimination registers

The interrupt (multiple factors / one vector interrupt) assigned to the interrupt source discrimination registers requires one instruction execution cycle (2 to 16 cycles of internal system clock) until starting the interrupt sequence more than that of “one factor / one vector interrupt”.

2.3 Timer X and timer Y

2.3.1 Timer X memory assignment

Figure 2.3.1 shows the memory assignment of timer X relevant registers. Each of these registers is described below.

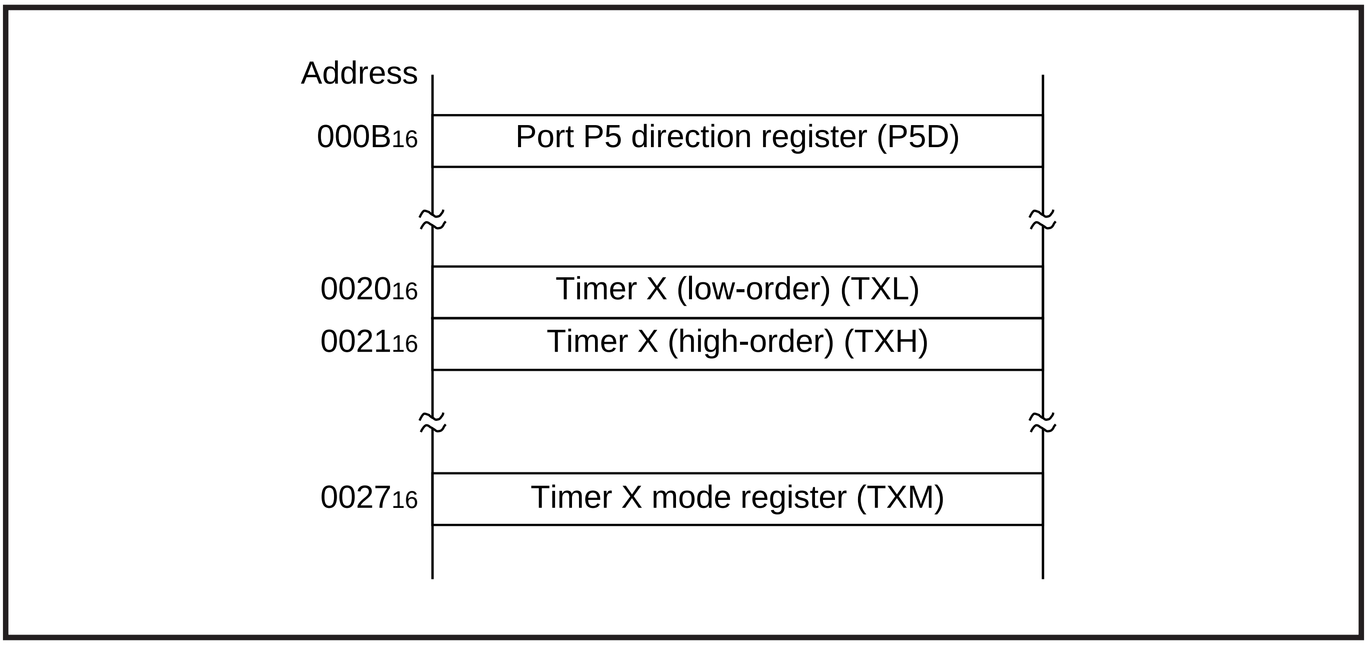

Address
$000B_{16}$
Port P5 direction register (P5D)
0020_{16}
Timer X (low-order) (TXL)
0021_{16}
Timer X (high-order) (TXH)
0027_{16}
Timer X mode register (TXM)

Fig. 2.3.1 Memory assignment of timer X relevant registers

2.3.2 Timer X relevant register

(1) Port P5 direction register (P5D)

■ **Pulse output mode**

When selecting the pulse output mode, set “1” to bit 4 of the port P5 direction register for the output mode of port P5$_4$.

■ **Event counter mode and pulse width measurement mode**

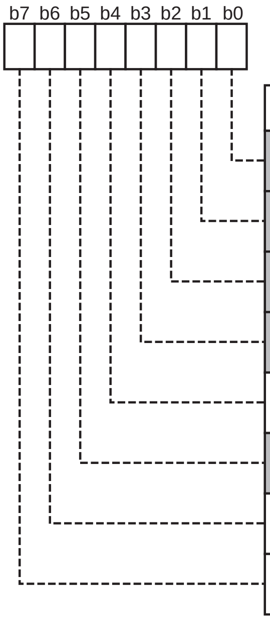
When selecting the event counter mode or the pulse width measurement mode, set “0” to bit 4 of the port P5 direction register for the input mode of port P5$_4$.

■ **Real time port function**

When using the real time port function, set “1” to bits 6 and 7 of the port P5 direction register for the output mode of ports P5$_6$ and P5$_7$.

Figure 2.3.2 shows the structure of the port P5 direction register.

Port P5 direction register



Port P5 direction register
(P5D: address 0B$_{16}$)

b	Name	Functions	At reset	R	W
0	Port P5 direction register	0 : Port P5$_0$ input mode 1 : Port P5$_0$ output mode	0	×	○
1		0 : Port P5$_1$ input mode 1 : Port P5$_1$ output mode	0	×	○
2		0 : Port P5$_2$ input mode 1 : Port P5$_2$ output mode	0	×	○
3		0 : Port P5$_3$ input mode 1 : Port P5$_3$ output mode	0	×	○
4		0 : Port P5$_4$ input mode 1 : Port P5$_4$ output mode	0	×	○
5		0 : Port P5$_5$ input mode 1 : Port P5$_5$ output mode	0	×	○
6		0 : Port P5$_6$ input mode 1 : Port P5$_6$ output mode	0	×	○
7		0 : Port P5$_7$ input mode 1 : Port P5$_7$ output mode	0	×	○

Note : The port P5 direction register cannot be read out.

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	P5$_7$	P5$_6$	P5$_5$	P5$_4$	P5$_3$	P5$_2$	P5$_1$	P5$_0$
	RTP$_1$	RTP$_0$	CNTR$_1$	CNTR$_0$	INT$_5$	INT$_4$	INT$_3$	T$_{OUT}$

Fig. 2.3.2 Structure of port P5 direction register

(2) Timer X latch and timer X (timer X (low-order) and timer X (high-order))

The timer X latch and timer X consist of 16 bits owing to a combination of high-order and low-order. Timer X and the latch are assigned to the same address. To access timer X and the latch, access both timer X (low-order) and timer X (high-order).

■ **Read**

When timer X (low-order) and timer X (high-order) are read, the contents (count value) of timer X are read. Read both registers in order of timer X (high-order) and the timer X (low-order) following. Do not write to timer X (high-order) and timer X (low-order) during a read operation. In this case, timer X will not operate normally.

■ **Write**

When a value is written to timer X (low-order) and timer X (high-order), the value is set in timer X and the latch at one time. Write both registers in order of timer X (low-order) and timer X (high-order) following.

Writing to the timer X latch only can be selected by setting “1” to the timer X write control bit of the timer X mode register. In this case, when writing to the timer X latch (high-order) and the underflow of timer X are performed at the almost same timing, unexpected value may be set in timer X (high-order).

Do not read timer X (low-order) and timer X (high-order) during a write operation. In this case, timer X will not operate normally.

■ **Timer X latch**

The timer X latch is a register which holds the value to be transferred automatically to timer X as the initial value of timer X at timer X underflow. Transfer is performed by next factors.

•Timer X underflow

•Valid edge input from CNTR$_0$ pin (in pulse width measurement mode)

The contents of the timer X latch cannot be read out.

■ **Timer X**

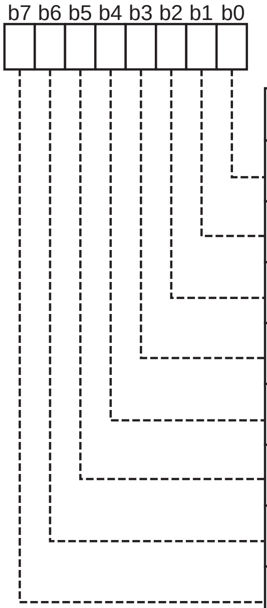
Timer X counts the count source.

The contents of timer X are decremented by “1” each time a count source is input. The division ratio of timer X is expressed by the following expression:

$$\text{Division ratio of timer X} = \frac{1}{\text{Timer X counter initial value} + 1}$$

Figure 2.3.3 shows the structure of timer X (low-order, high-order).

Timer X (low-order, high-order)



Timer X (low-order, high-order)
(TXL, TXH: addresses 20_{16}, 21_{16})

b	Functions	At reset	R	W
0	• Set "0000_{16} to $FFFF_{16}$" as timer X count value. • When the timer X write control bit of the timer X mode register is "0", the value is written to timer X and the latch at one time. When the timer X write control bit of the timer X mode register is "1", the value is written only to the latch. • Count-down method Timer X underflows at the first count pulse after the value of timer X reaches "0". At this time, the value of the timer X latch is transferred to timer X automatically. • The timer X count value is read out by reading this register.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

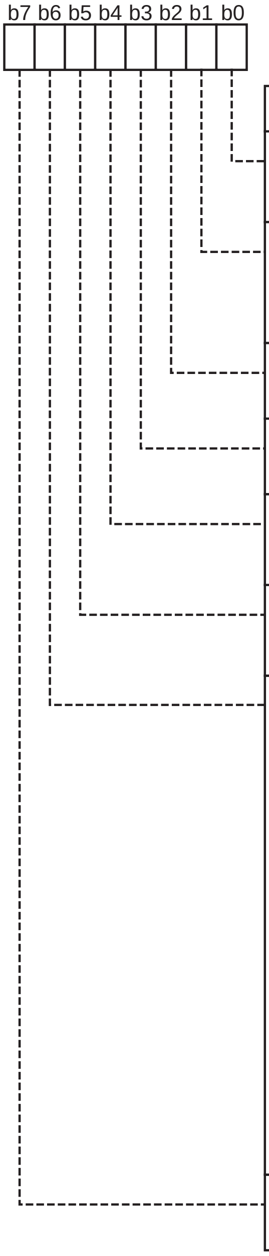
Notes 1: When reading and writing, perform them to both the high-order and low-order bytes.
2: Read both registers in order of TXH and TXL following.
3: Write both registers in order of TXL and TXH following.
4: Do not read both registers during a write, and do not write to both registers during a read.

Fig. 2.3.3 Structure of timer X (low-order, high-order)

(3) Timer X mode register

The timer X mode register consists of operation select bits, count control bit and etc. Figure 2.3.4 shows the structure of the timer X mode register.

Timer X mode register



Timer X mode register
(TXM: address 27_{16})

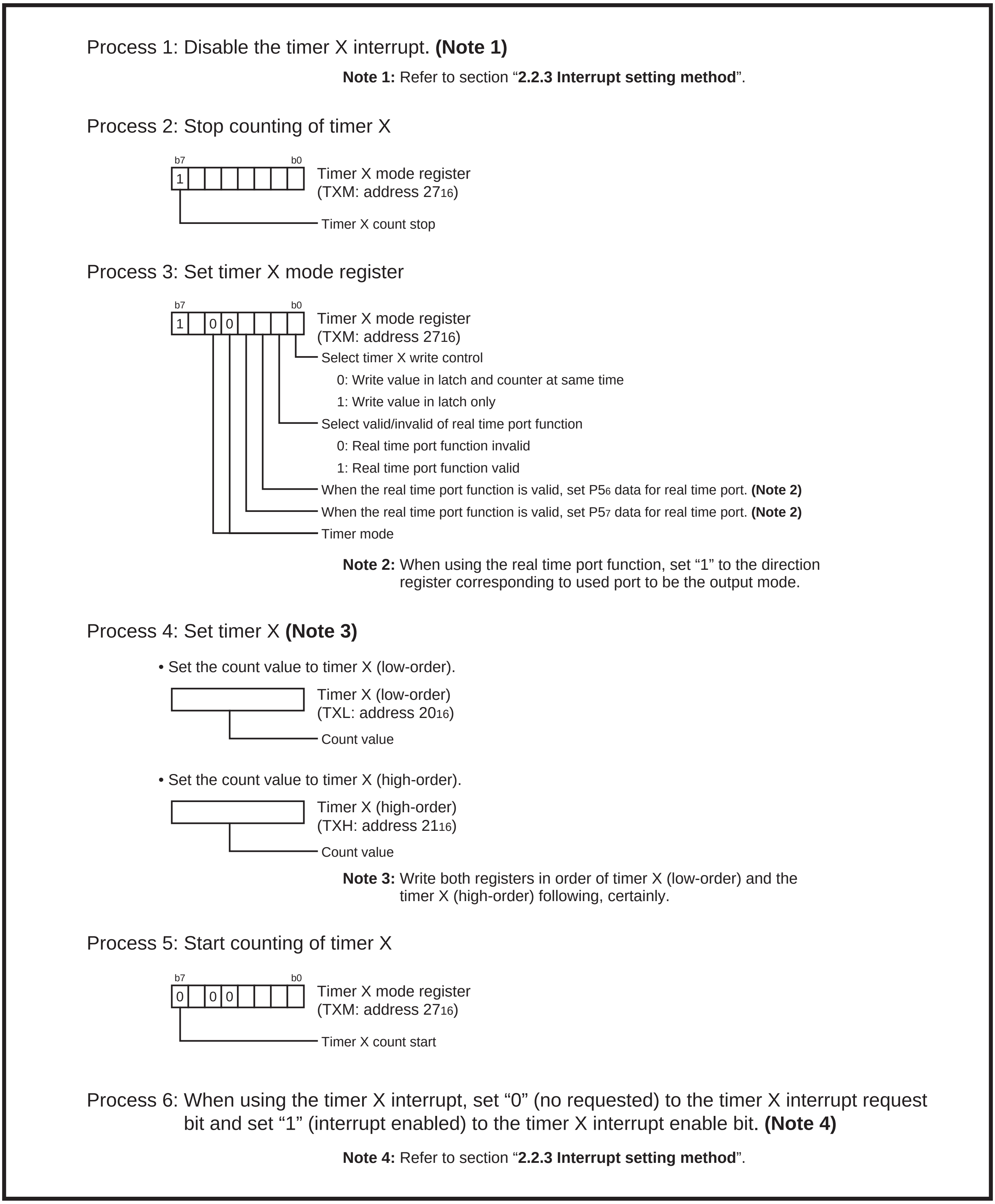
b	Name	Functions	At reset	R	W
0	Timer X write control bit	0 : Write value in latch and counter 1 : Write value in latch only	0	○	○
1	Real time port control bit	0 : Real time port function invalid 1 : Real time port function valid	0	○	○
2	$P5_6$ data for real time port	0 : “0” output 1 : “1” output	0	○	○
3	$P5_7$ data for real time port	0 : “0” output 1 : “1” output	0	○	○
4	Timer X operating mode bits	b5 b4 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
5			0	○	○
6	$CNTR_0$ active edge switch bit	0 : • Count at rising edge in event counter mode • Start from “H” output in pulse output mode • Measure “H” pulse width in pulse width measurement mode • Falling edge active for $CNTR_0$ interrupt 1 : • Count at falling edge in event counter mode • Start from “L” output in pulse output mode • Measure “L” pulse width in pulse width measurement mode • Rising edge active for $CNTR_0$ interrupt	0	○	○
7	Timer X stop control bit	0 : Count start 1 : Count stop	0	○	○

Fig. 2.3.4 Structure of timer X mode register

2.3.3 Timer mode of timer X

(1) Timer mode setting method

Figure 2.3.5 shows the setting method for timer mode of timer X.



Process 1: Disable the timer X interrupt. **(Note 1)**

Note 1: Refer to section “**2.2.3 Interrupt setting method**”.

Process 2: Stop counting of timer X

b7
b0
1
Timer X mode register
(TXM: address 27_{16})
Timer X count stop

Process 3: Set timer X mode register

b7
b0
1
0
0
Timer X mode register
(TXM: address 27_{16})
Select timer X write control
0: Write value in latch and counter at same time
1: Write value in latch only
Select valid/invalid of real time port function
0: Real time port function invalid
1: Real time port function valid
When the real time port function is valid, set $P5_6$ data for real time port. (Note 2)
When the real time port function is valid, set $P5_7$ data for real time port. (Note 2)
Timer mode

Note 2: When using the real time port function, set “1” to the direction register corresponding to used port to be the output mode.

Process 4: Set timer X **(Note 3)**

- Set the count value to timer X (low-order).

Timer X (low-order)
(TXL: address 20_{16})
Count value

- Set the count value to timer X (high-order).

Note 3: Write both registers in order of timer X (low-order) and the timer X (high-order) following, certainly.

Process 5: Start counting of timer X

Process 6: When using the timer X interrupt, set “0” (no requested) to the timer X interrupt request bit and set “1” (interrupt enabled) to the timer X interrupt enable bit. **(Note 4)**

Note 4: Refer to section “**2.2.3 Interrupt setting method**”.

Fig. 2.3.5 Setting method for timer mode of timer X

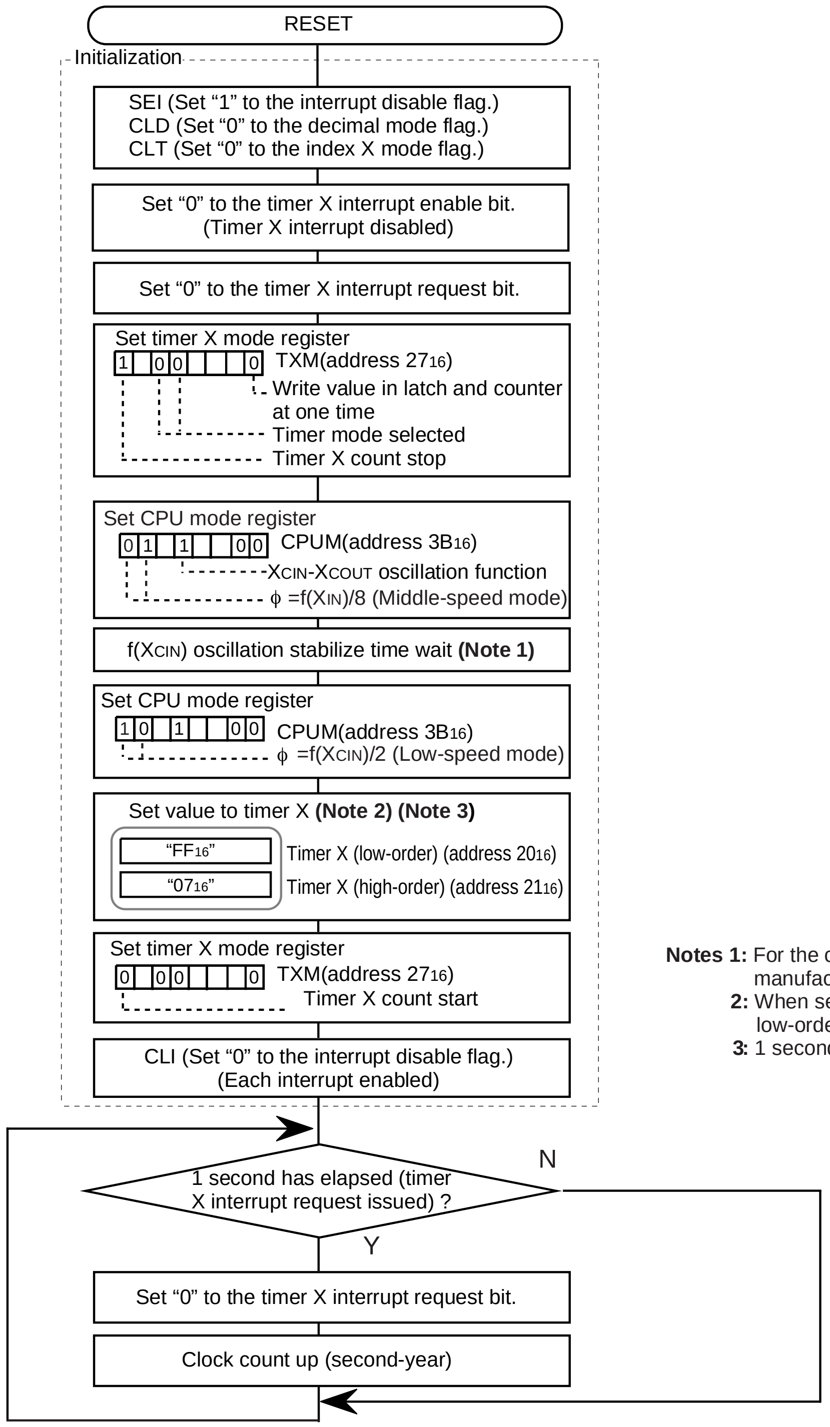
(2) Timer mode application example

■ Outline

The clock is divided by timer X. The clock is counted up at one-second intervals.

■ Specifications

The f(X_{CIN}) = 32.768 kHz is divided by timer X to detect one second. The timer X interrupt regist is confirmed in the main routine. When one second has elapsed, the clock is counted up.
Figure 2.3.6 shows an example of control procedure.


RESET
Initialization
SEI (Set “1” to the interrupt disable flag.)
CLD (Set “0” to the decimal mode flag.)
CLT (Set “0” to the index X mode flag.)
Set “0” to the timer X interrupt enable bit.
(Timer X interrupt disabled)
Set “0” to the timer X interrupt request bit.
Set timer X mode register
1 0 0 0
TXM(address 27₁₆)
Write value in latch and counter at one time
Timer mode selected
Timer X count stop
Set CPU mode register
0 1 1 0 0
CPUM(address 3B₁₆)
XCIN-XCOUT oscillation function
φ =f(XIN)/8 (Middle-speed mode)
f(XCIN) oscillation stabilize time wait (Note 1)
Set CPU mode register
1 0 1 0 0
CPUM(address 3B₁₆)
φ =f(XCIN)/2 (Low-speed mode)
Set value to timer X (Note 2) (Note 3)
“FF₁₆”
“07₁₆”
Timer X (low-order) (address 20₁₆)
Timer X (high-order) (address 21₁₆)
Set timer X mode register
0 0 0 0
TXM(address 27₁₆)
Timer X count start
CLI (Set “0” to the interrupt disable flag.)
(Each interrupt enabled)
1 second has elapsed (timer X interrupt request issued) ?
N
Y
Set “0” to the timer X interrupt request bit.
Clock count up (second-year)

Notes 1: For the concrete time, ask the resonator manufacture.
2: When setting the value of timer X, set in order of low-order byte and high-order byte following.
3: 1 second = 1/32.768 kHz × $\underbrace{16}_{\text{Fixed division ratio}}$ × $\underbrace{(2047 + 1)}_{7FF_{16}}$

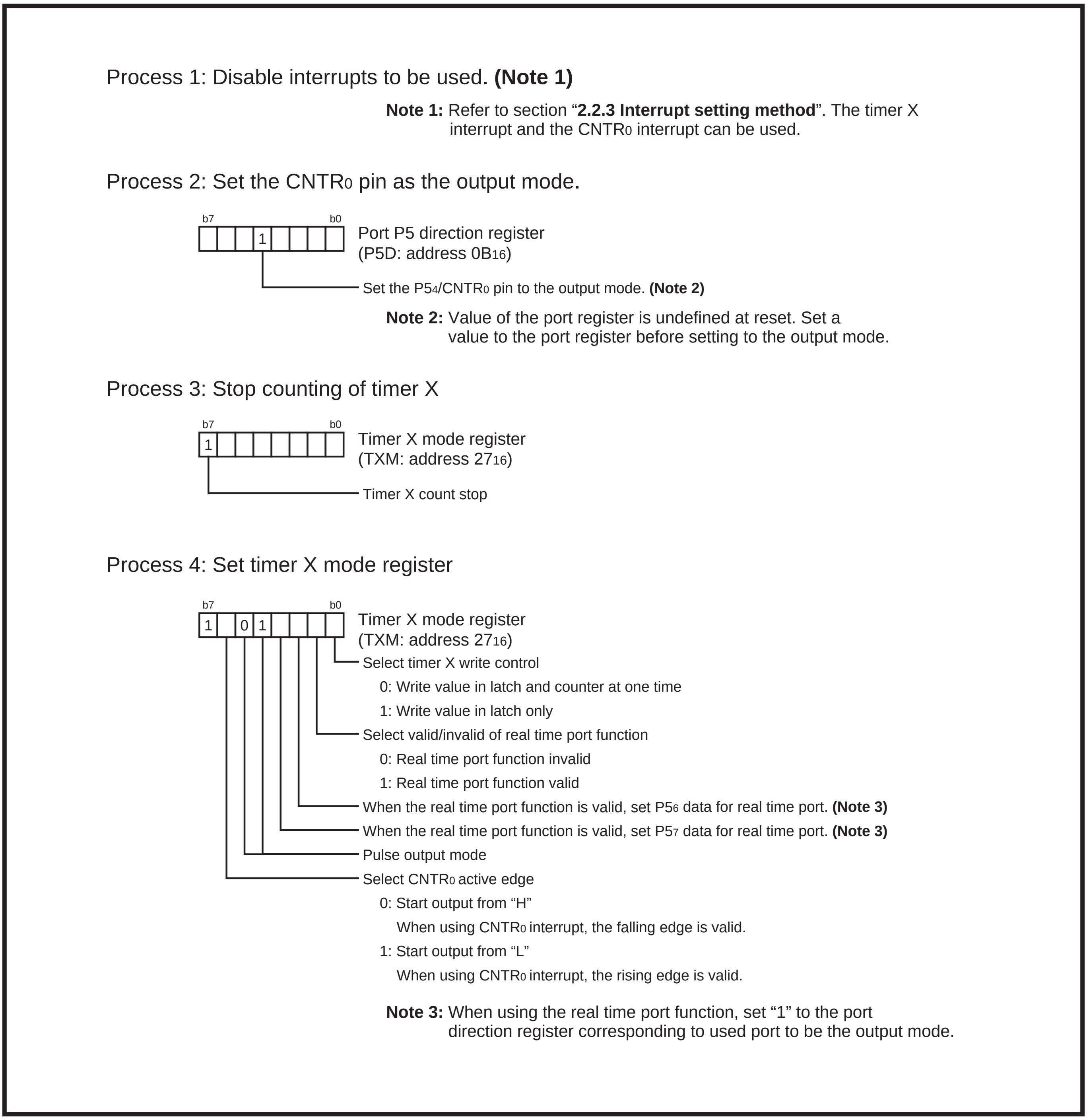
Fig. 2.3.6 Example of control procedure

2.3.4 Pulse output mode of timer X

The pulse output mode operates just as in the timer mode, and also has a pulse output operation. This mode outputs a pulse whose polarity is reversed each time timer X counter underflows from the P5$_4$/CNTR$_0$ pin.

(1) Pulse output mode setting method

Figures 2.3.7 and 2.3.8 show the setting method for the pulse output mode of timer X.



Process 1: Disable interrupts to be used. **(Note 1)**

Note 1: Refer to section "**2.2.3 Interrupt setting method**". The timer X interrupt and the CNTR$_0$ interrupt can be used.

Process 2: Set the CNTR$_0$ pin as the output mode.

Port P5 direction register (P5D: address 0B$_{16}$)

Set the P5$_4$/CNTR$_0$ pin to the output mode. **(Note 2)**

Note 2: Value of the port register is undefined at reset. Set a value to the port register before setting to the output mode.

Process 3: Stop counting of timer X

Timer X mode register (TXM: address 27$_{16}$)

Timer X count stop

Process 4: Set timer X mode register

Timer X mode register (TXM: address 27$_{16}$)

Select timer X write control
0: Write value in latch and counter at one time
1: Write value in latch only

Select valid/invalid of real time port function
0: Real time port function invalid
1: Real time port function valid

When the real time port function is valid, set P5$_6$ data for real time port. **(Note 3)**

When the real time port function is valid, set P5$_7$ data for real time port. **(Note 3)**

Pulse output mode

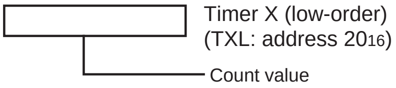
Select CNTR$_0$ active edge
0: Start output from "H"
When using CNTR$_0$ interrupt, the falling edge is valid.
1: Start output from "L"
When using CNTR$_0$ interrupt, the rising edge is valid.

Note 3: When using the real time port function, set "1" to the port direction register corresponding to used port to be the output mode.

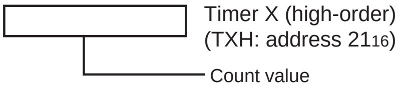
Fig. 2.3.7 Setting method for pulse output mode of timer X (1)

Process 5: Set timer X **(Note 4)**

- Set the count value to timer X (low-order).

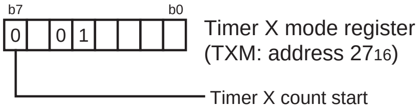


- Set the count value to timer X (high-order).



Note 4: Write both registers in order of timer X (low-order) and timer X (high-order) following, certainly.

Process 6: Start counting of timer X



Process 7: When using the interrupt, set “0” (no requested) to the interrupt request bit of the interrupt to be used, and set “1” (interrupt enabled) to the interrupt enable bit to be used. **(Note 5)**

Note 5: Refer to section “**2.2.3 Interrupt setting method**”. The timer X interrupt and the $CNTR_0$ interrupt can be used.

Fig. 2.3.8 Setting method for pulse output mode of timer X (2)

(2) Application examples of pulse output mode

■ **Outline**

The rectangular waveform output function of timer is applied for a piezoelectric buzzer output.

■ **Specifications**

The rectangular waveform which is clock $f(X_{IN})$ = 4.19 MHz divided up to about 2 kHz (2048 Hz) is output from the $P5_4/CNTR_0$ pin.

The level of the $P5_4/CNTR_0$ pin is fixed to "H" while a piezoelectric buzzer output is stopped.

Figure 2.3.9 shows an example of peripheral circuit; Figure 2.3.10 shows the connection of timer and setting of the division ratio; Figure 2.3.11 shows an example of the control procedure.

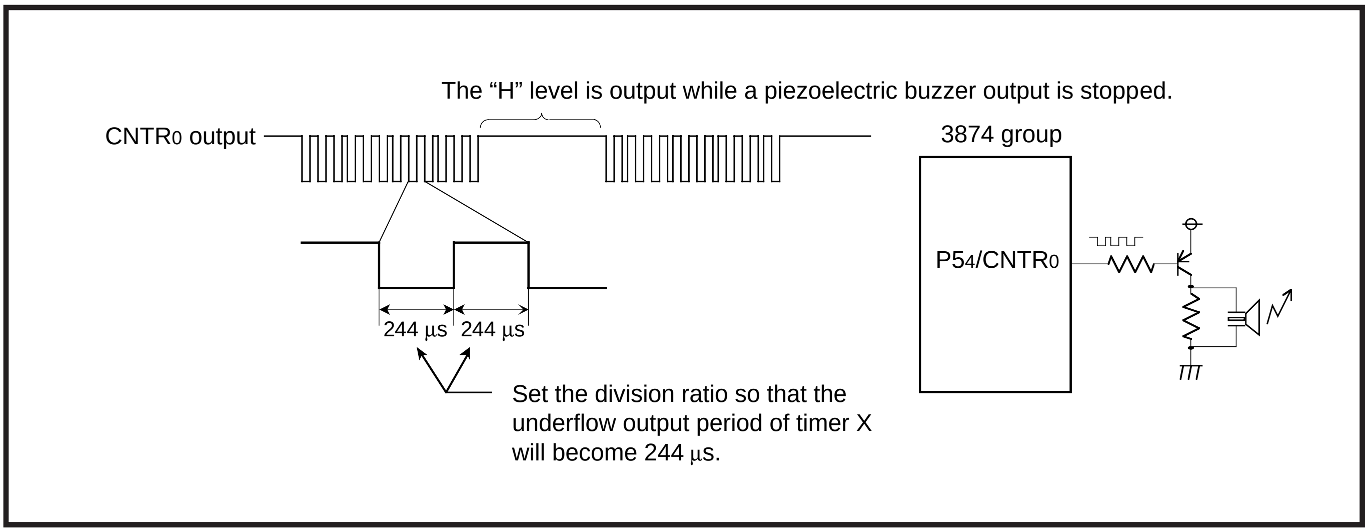

The "H" level is output while a piezoelectric buzzer output is stopped.
CNTR0 output
3874 group
P54/CNTR0
244 µs
244 µs
Set the division ratio so that the underflow output period of timer X will become 244 µs.

Fig. 2.3.9 Example of peripheral circuit

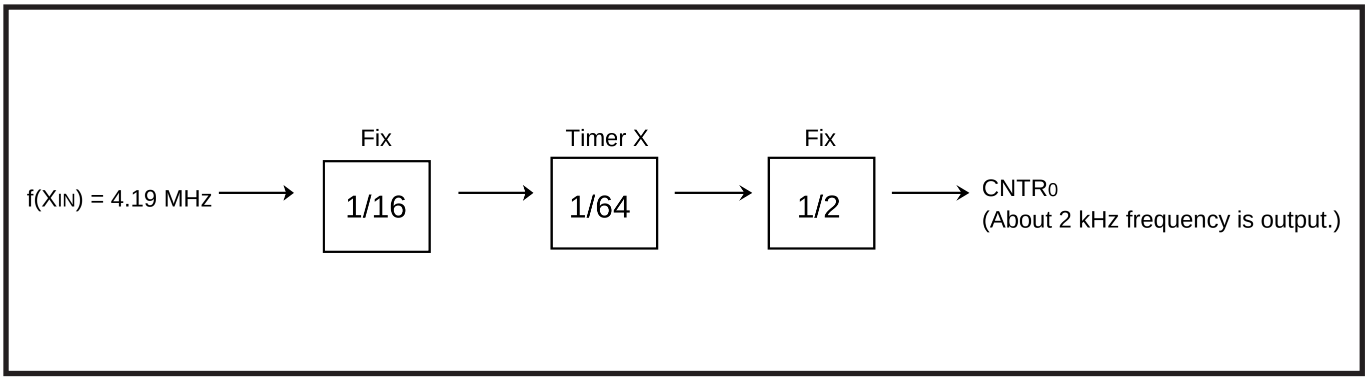
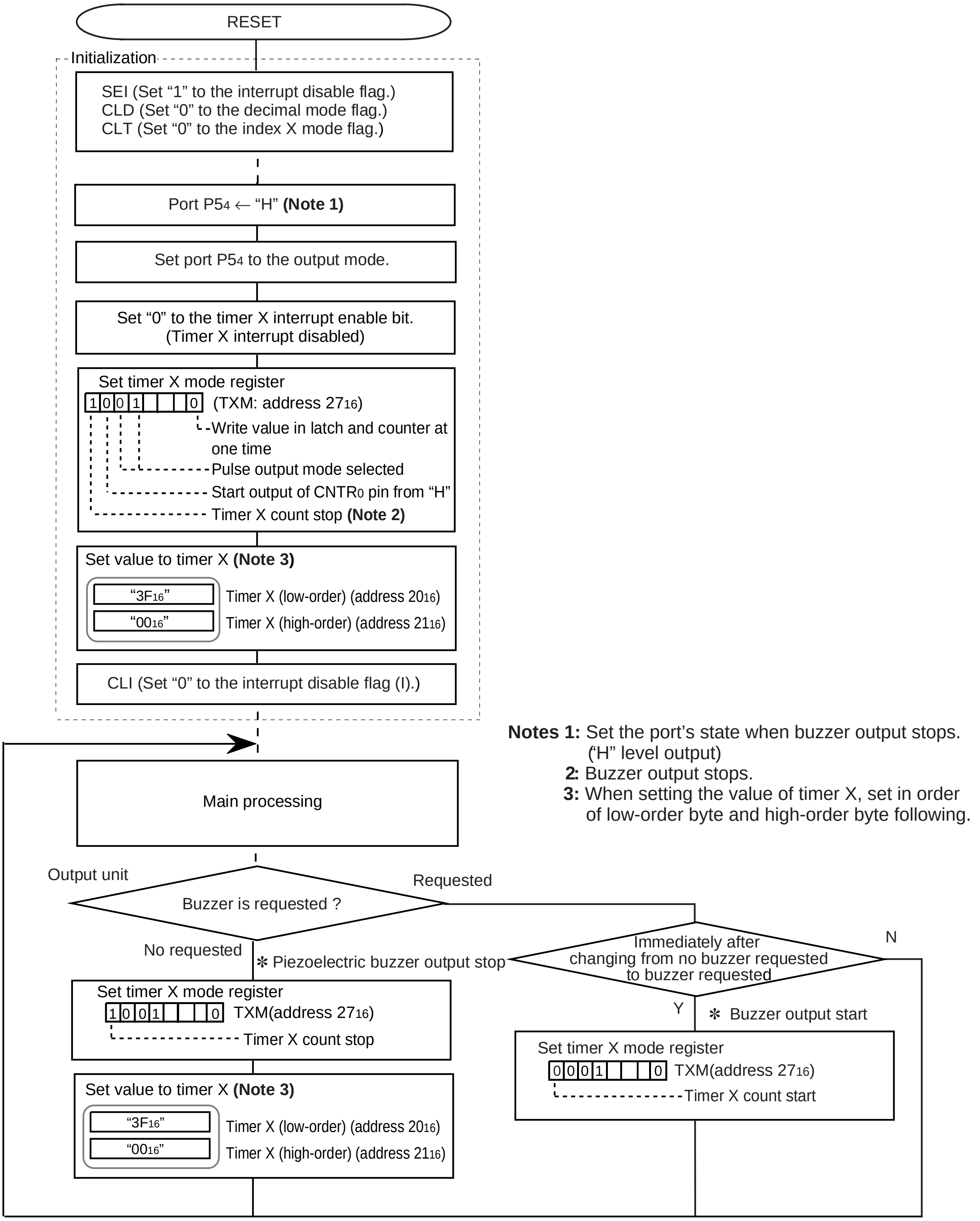


Fig. 2.3.10 Connection of timer and setting of division ratio


RESET
Initialization
SEI (Set “1” to the interrupt disable flag.)
CLD (Set “0” to the decimal mode flag.)
CLT (Set “0” to the index X mode flag.)
Port P54 ← “H” (Note 1)
Set port P54 to the output mode.
Set “0” to the timer X interrupt enable bit.
(Timer X interrupt disabled)
Set timer X mode register
1 0 0 1 0
(TXM: address 2716)
Write value in latch and counter at one time
Pulse output mode selected
Start output of CNTR0 pin from “H”
Timer X count stop (Note 2)
Set value to timer X (Note 3)
“3F16”
“0016”
Timer X (low-order) (address 2016)
Timer X (high-order) (address 2116)
CLI (Set “0” to the interrupt disable flag (I).)
Main processing
Output unit
Buzzer is requested ?
Requested
No requested
✻ Piezoelectric buzzer output stop
Immediately after changing from no buzzer requested to buzzer requested
N
Y
✻ Buzzer output start
Set timer X mode register
1 0 0 1 0
TXM(address 2716)
Timer X count stop
Set value to timer X (Note 3)
“3F16”
“0016”
Timer X (low-order) (address 2016)
Timer X (high-order) (address 2116)
Set timer X mode register
0 0 0 1 0
TXM(address 2716)
Timer X count start

Notes 1: Set the port’s state when buzzer output stops. (“H” level output)

2: Buzzer output stops.

3: When setting the value of timer X, set in order of low-order byte and high-order byte following.

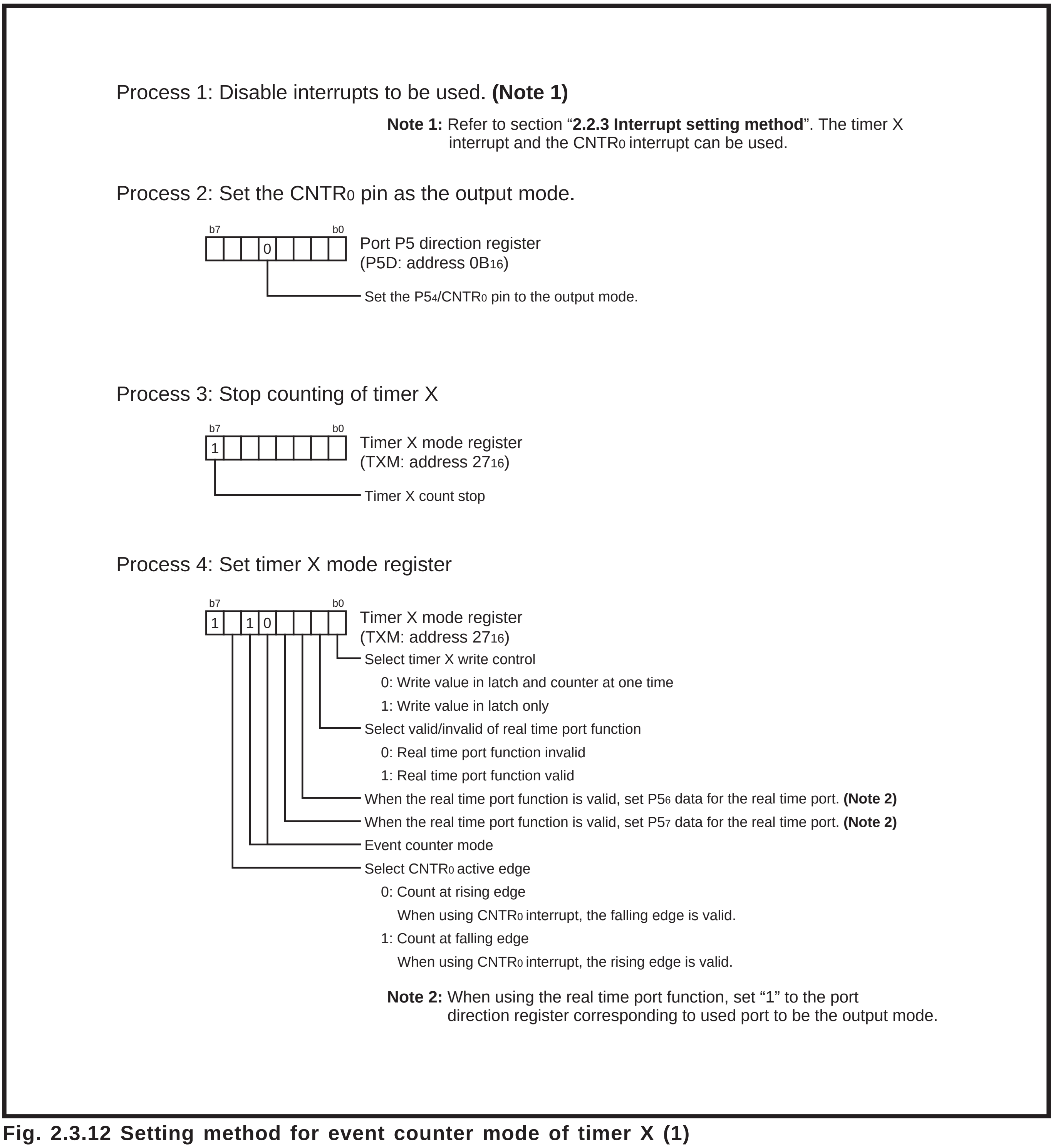
Fig. 2.3.11 Example of control procedure

2.3.5 Event counter mode of timer X

The event counter mode operates just as in the timer mode except that the input signal from the $P5_4/CNTR_0$ pin becomes the count source.

(1) Event counter mode setting method

Figures 2.3.12 and 2.3.13 show the setting method for the event counter mode of timer X.



Process 1: Disable interrupts to be used. **(Note 1)**

Note 1: Refer to section "**2.2.3 Interrupt setting method**". The timer X interrupt and the $CNTR_0$ interrupt can be used.

Process 2: Set the $CNTR_0$ pin as the output mode.

Port P5 direction register
(P5D: address $0B_{16}$)

Set the $P5_4/CNTR_0$ pin to the output mode.

Process 3: Stop counting of timer X

Timer X mode register
(TXM: address 27_{16})

Timer X count stop

Process 4: Set timer X mode register

Timer X mode register
(TXM: address 27_{16})

Select timer X write control
0: Write value in latch and counter at one time
1: Write value in latch only

Select valid/invalid of real time port function
0: Real time port function invalid
1: Real time port function valid

When the real time port function is valid, set $P5_6$ data for the real time port. **(Note 2)**

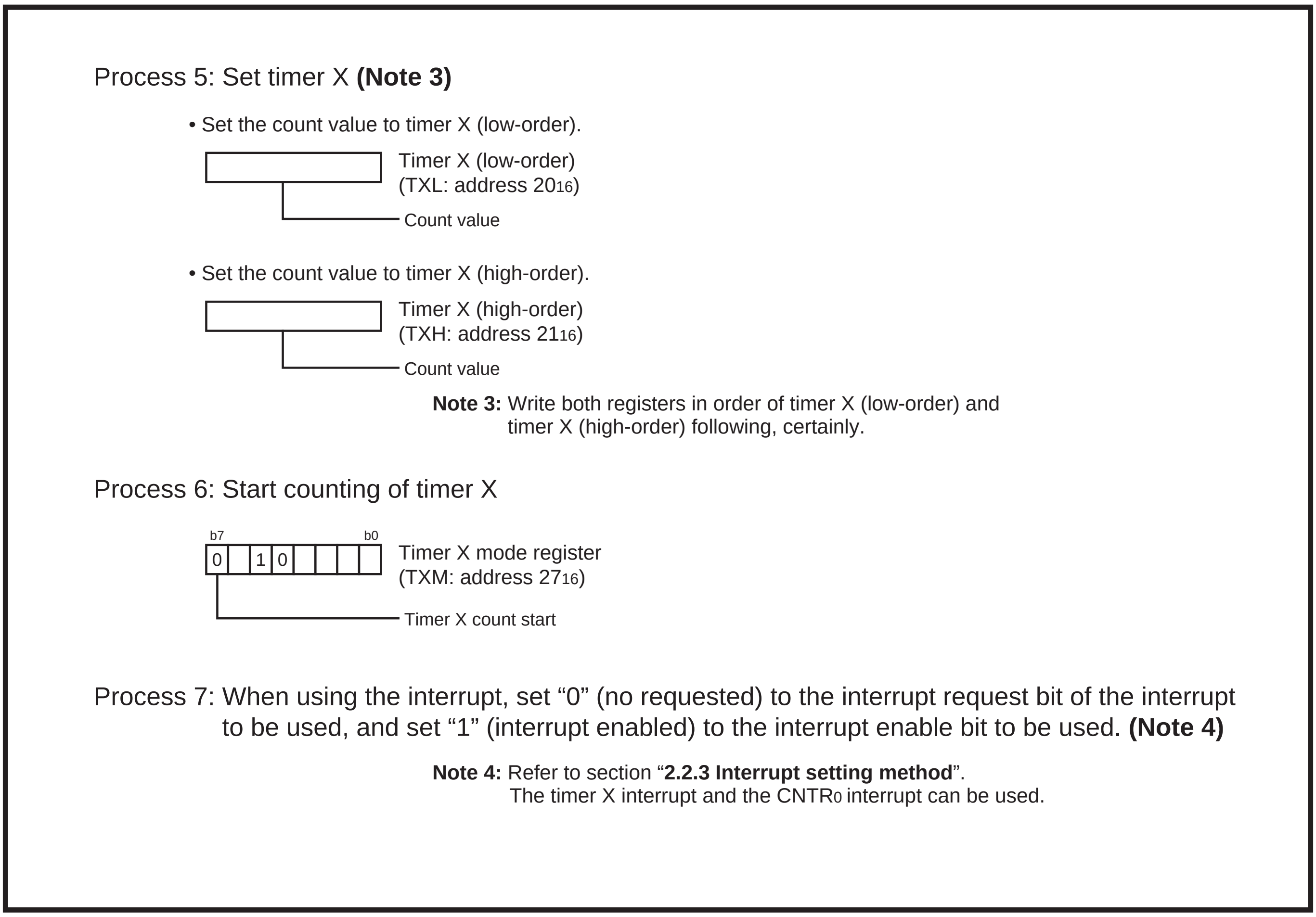
When the real time port function is valid, set $P5_7$ data for the real time port. **(Note 2)**

Event counter mode

Select $CNTR_0$ active edge
0: Count at rising edge
When using $CNTR_0$ interrupt, the falling edge is valid.
1: Count at falling edge
When using $CNTR_0$ interrupt, the rising edge is valid.

Note 2: When using the real time port function, set "1" to the port direction register corresponding to used port to be the output mode.

Fig. 2.3.12 Setting method for event counter mode of timer X (1)



Process 5: Set timer X **(Note 3)**

- Set the count value to timer X (low-order).

- Set the count value to timer X (high-order).

Note 3: Write both registers in order of timer X (low-order) and timer X (high-order) following, certainly.

Process 6: Start counting of timer X

Process 7: When using the interrupt, set “0” (no requested) to the interrupt request bit of the interrupt to be used, and set “1” (interrupt enabled) to the interrupt enable bit to be used. **(Note 4)**

Note 4: Refer to section “**2.2.3 Interrupt setting method**”.
The timer X interrupt and the $CNTR_0$ interrupt can be used.

Fig. 2.3.13 Setting method for event counter mode of timer X (2)

(2) Application examples of event counter mode

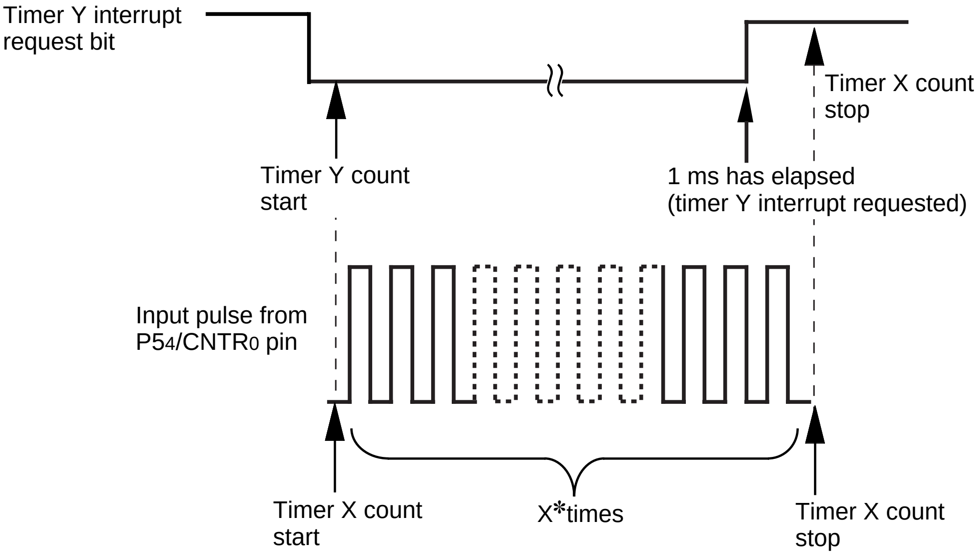
■ **Outline**

The frequency of the pulse which is input to the $P5_4/CNTR_0$ pin (“H” active) is measured by the number of events in a certain period.

■ **Specifications**

The count source of timer X is input from the $P5_4/CNTR_0$ pin, and the timer X starts counting the count source. Clock ($f(X_{IN})$ = 6.4 MHz) is divided by timer Y to detect 1 ms. The frequency of the pulse input to the $P5_4/CNTR_0$ pin is calculated by the number of events counted within 1 ms. (The event number of an input pulse is specified as “$FFFF_{16}$” or less within 1 ms.)

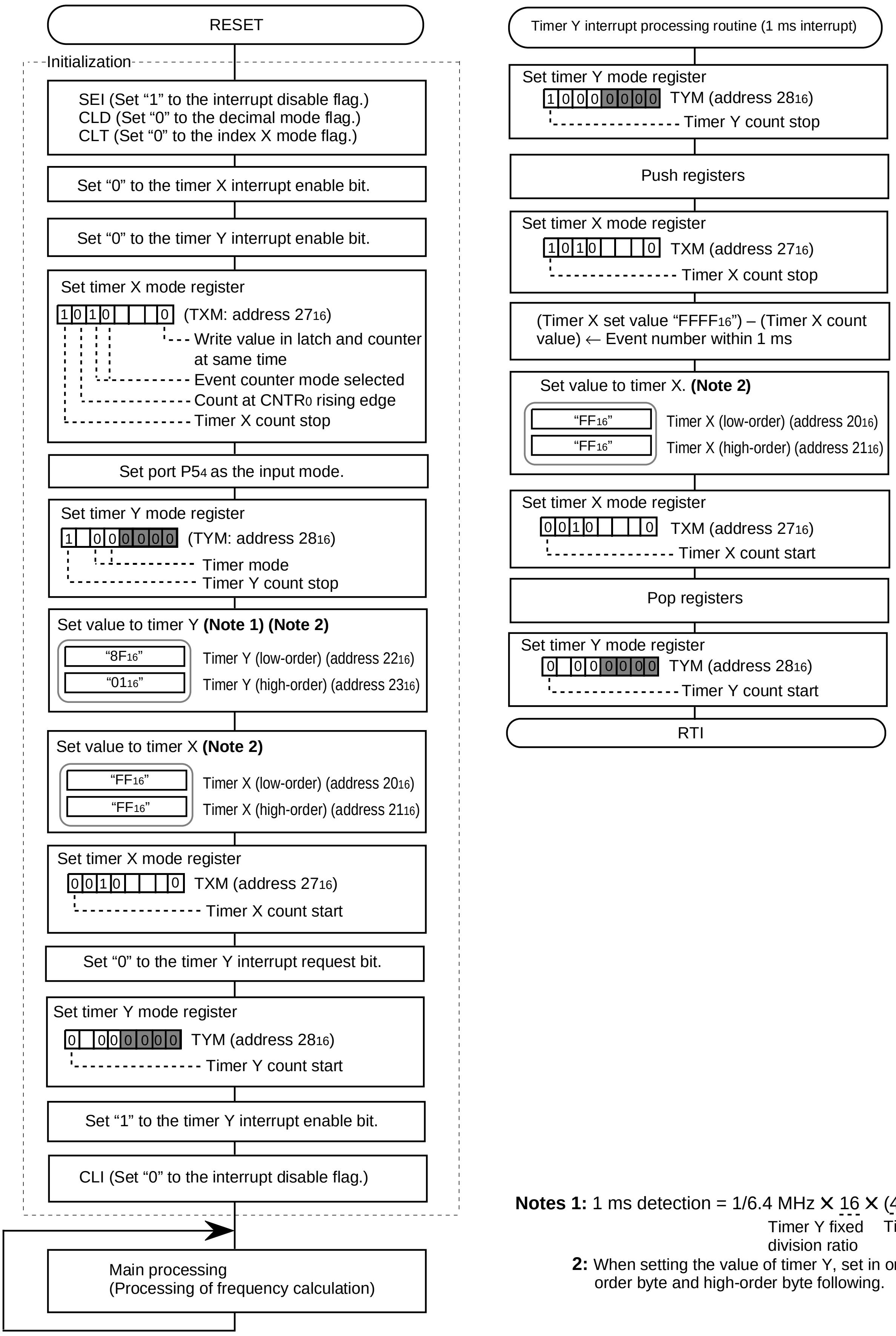
Figure 2.3.14 shows an example of measurement method of frequency and Figure 2.3.15 shows an example of control procedure.


Timer Y interrupt request bit
Timer X count stop
Timer Y count start
1 ms has elapsed (timer Y interrupt requested)
Input pulse from P54/CNTR0 pin
Timer X count start
X*times
Timer X count stop

Note: Pulse frequency input from $P5_4/CNTR_0$ pin = $\frac{X^{*}\text{times}}{1\text{ ms}}$ kHz

✻: Number of pulse input from $CNTR_0$ pin within 1 ms

Fig. 2.3.14 Example of measurement method of frequency


RESET
Initialization
SEI (Set "1" to the interrupt disable flag.)
CLD (Set "0" to the decimal mode flag.)
CLT (Set "0" to the index X mode flag.)
Set "0" to the timer X interrupt enable bit.
Set "0" to the timer Y interrupt enable bit.
Set timer X mode register
1 0 1 0 _ _ _ 0 (TXM: address 27₁₆)
Write value in latch and counter at same time
Event counter mode selected
Count at CNTR₀ rising edge
Timer X count stop
Set port P5₄ as the input mode.
Set timer Y mode register
1 _ 0 0 0 0 0 0 (TYM: address 28₁₆)
Timer mode
Timer Y count stop
Set value to timer Y (Note 1) (Note 2)
"8F₁₆" Timer Y (low-order) (address 22₁₆)
"01₁₆" Timer Y (high-order) (address 23₁₆)
Set value to timer X (Note 2)
"FF₁₆" Timer X (low-order) (address 20₁₆)
"FF₁₆" Timer X (high-order) (address 21₁₆)
Set timer X mode register
0 0 1 0 _ _ _ 0 TXM (address 27₁₆)
Timer X count start
Set "0" to the timer Y interrupt request bit.
Set timer Y mode register
0 _ 0 0 0 0 0 0 TYM (address 28₁₆)
Timer Y count start
Set "1" to the timer Y interrupt enable bit.
CLI (Set "0" to the interrupt disable flag.)
Main processing
(Processing of frequency calculation)
Timer Y interrupt processing routine (1 ms interrupt)
Set timer Y mode register
1 0 0 0 0 0 0 0 TYM (address 28₁₆)
Timer Y count stop
Push registers
Set timer X mode register
1 0 1 0 _ _ _ 0 TXM (address 27₁₆)
Timer X count stop
(Timer X set value "FFFF₁₆") – (Timer X count value) ← Event number within 1 ms
Set value to timer X. (Note 2)
"FF₁₆" Timer X (low-order) (address 20₁₆)
"FF₁₆" Timer X (high-order) (address 21₁₆)
Set timer X mode register
0 0 1 0 _ _ _ 0 TXM (address 27₁₆)
Timer X count start
Pop registers
Set timer Y mode register
0 _ 0 0 0 0 0 0 TYM (address 28₁₆)
Timer Y count start
RTI

Notes 1: 1 ms detection = 1/6.4 MHz × $\underbrace{16}_{\text{Timer Y fixed division ratio}}$ × $(\underbrace{400 - 1}_{\text{Timer Y set}})$

2: When setting the value of timer Y, set in order of low-order byte and high-order byte following.

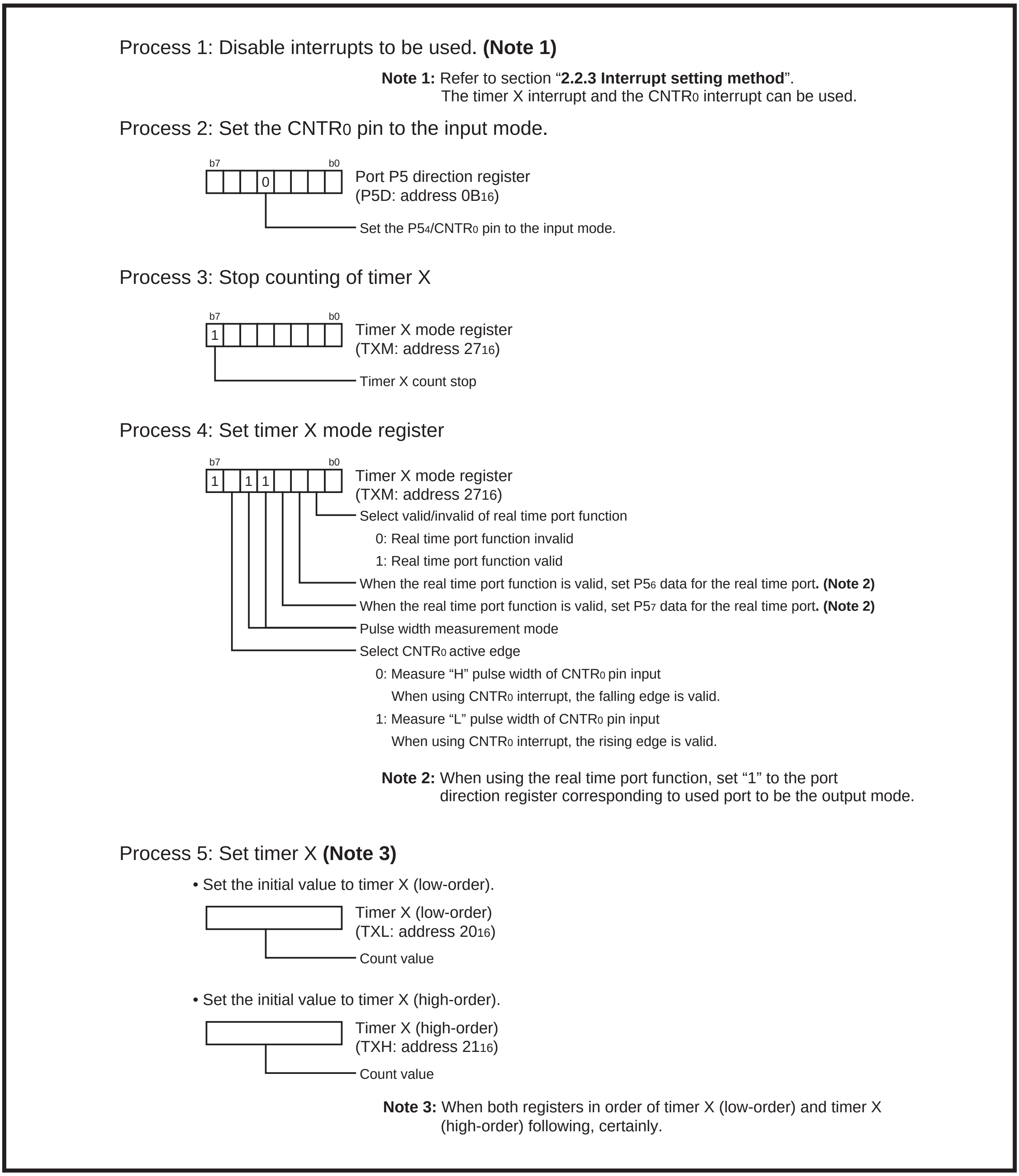
Fig. 2.3.15 Example of control procedure

2.3.6 Pulse width measurement mode of timer X

In the pulse width measurement mode, the timer measures a pulse width ("H" or "L" level) input to the $P5_4/CNTR_0$ pin.

(1) Pulse width measurement mode setting method

Figures 2.3.16 and 2.3.17 show the setting method for the pulse width measurement mode of timer X.



Process 1: Disable interrupts to be used. **(Note 1)**

Note 1: Refer to section "**2.2.3 Interrupt setting method**".
The timer X interrupt and the $CNTR_0$ interrupt can be used.

Process 2: Set the $CNTR_0$ pin to the input mode.

Port P5 direction register
(P5D: address $0B_{16}$)

Set the $P5_4/CNTR_0$ pin to the input mode.

Process 3: Stop counting of timer X

Timer X mode register
(TXM: address 27_{16})

Timer X count stop

Process 4: Set timer X mode register

Timer X mode register
(TXM: address 27_{16})

Select valid/invalid of real time port function
0: Real time port function invalid
1: Real time port function valid

When the real time port function is valid, set $P5_6$ data for the real time port. **(Note 2)**

When the real time port function is valid, set $P5_7$ data for the real time port. **(Note 2)**

Pulse width measurement mode

Select $CNTR_0$ active edge
0: Measure "H" pulse width of $CNTR_0$ pin input
When using $CNTR_0$ interrupt, the falling edge is valid.
1: Measure "L" pulse width of $CNTR_0$ pin input
When using $CNTR_0$ interrupt, the rising edge is valid.

Note 2: When using the real time port function, set "1" to the port direction register corresponding to used port to be the output mode.

Process 5: Set timer X **(Note 3)**

• Set the initial value to timer X (low-order).

Timer X (low-order)
(TXL: address 20_{16})

Count value

• Set the initial value to timer X (high-order).

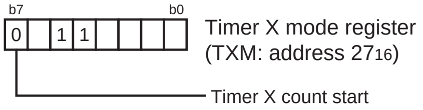
Timer X (high-order)
(TXH: address 21_{16})

Count value

Note 3: When both registers in order of timer X (low-order) and timer X (high-order) following, certainly.

Fig. 2.3.16 Setting method for pulse width measurement mode of timer X (1)

Process 6: Start counting of timer X



Process 7: When using the interrupt, set “0” (no requested) to the interrupt request bit of the interrupt to be used, and set “1” (interrupt enabled) to the interrupt enable bit to be used. **(Note 4)**

Note 4: Refer to section “**2.2.3 Interrupt setting method**”.
The timer X interrupt and the $CNTR_0$ interrupt can be used.

Fig. 2.3.17 Setting method for pulse width measurement mode of timer X (2)

(2) Application examples of pulse width measurement mode

■ **Outline**

The “H” level width of a motor’s FG pulse is measured in the pulse width measurement mode.

■ **Specifications**

The “H” level width of a FG pulse input to the $P5_4$/$CNTR_0$ pin is counted by timer X.
(When $f(X_{IN})$* is 4.19 MHz, the count source becomes 3.8 µs divided by 16. Measurement can be made up to 250 ms in the range of “$FFFF_{16}$” to “0000_{16}”.)

Figure 2.3.18 shows a connection of the timer and setting of the division ratio and Figure 2.3.19 shows an example of control procedure.

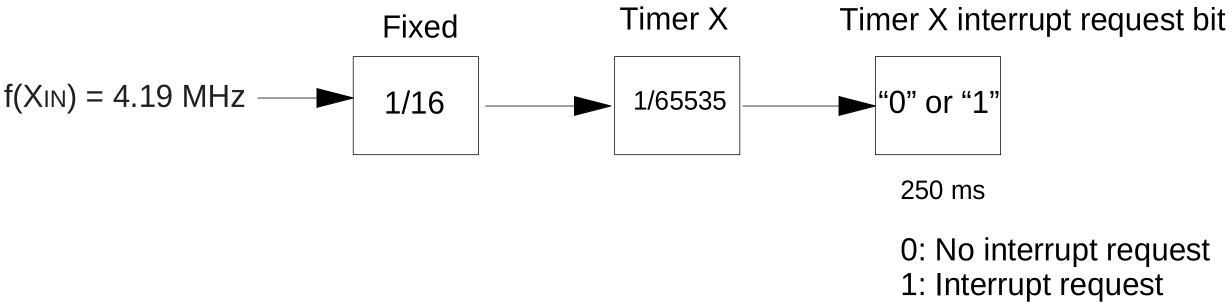
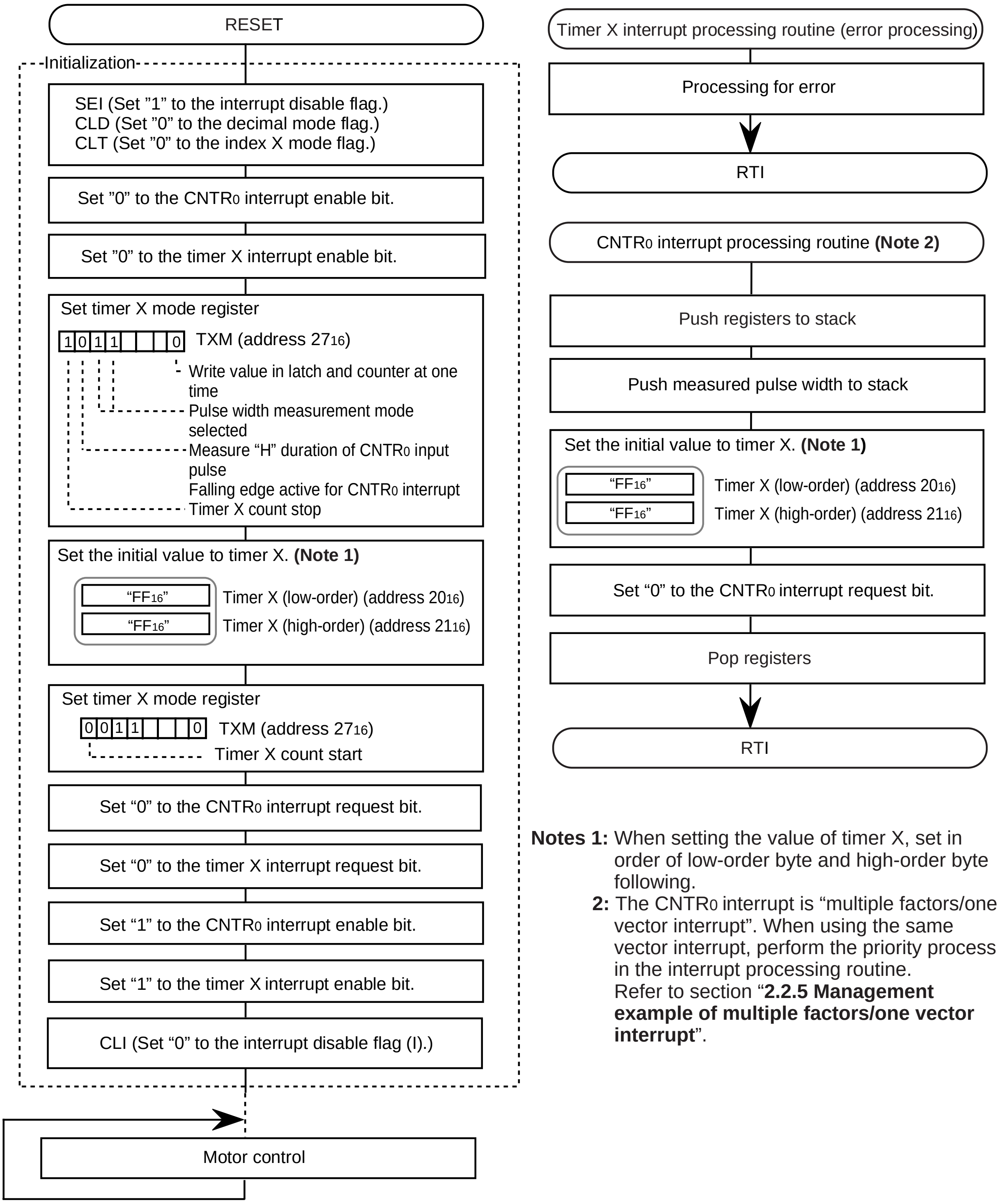


Fig. 2.3.18 Connection of timer and setting of division ratio


RESET
Initialization
SEI (Set ”1” to the interrupt disable flag.)
CLD (Set ”0” to the decimal mode flag.)
CLT (Set ”0” to the index X mode flag.)
Set ”0” to the CNTR0 interrupt enable bit.
Set ”0” to the timer X interrupt enable bit.
Set timer X mode register
1 0 1 1 0
TXM (address 2716)
Write value in latch and counter at one time
Pulse width measurement mode selected
Measure “H” duration of CNTR0 input pulse
Falling edge active for CNTR0 interrupt
Timer X count stop
Set the initial value to timer X. (Note 1)
“FF16”
“FF16”
Timer X (low-order) (address 2016)
Timer X (high-order) (address 2116)
Set timer X mode register
0 0 1 1 0
TXM (address 2716)
Timer X count start
Set “0” to the CNTR0 interrupt request bit.
Set “0” to the timer X interrupt request bit.
Set “1” to the CNTR0 interrupt enable bit.
Set “1” to the timer X interrupt enable bit.
CLI (Set “0” to the interrupt disable flag (I).)
Motor control
Timer X interrupt processing routine (error processing)
Processing for error
RTI
CNTR0 interrupt processing routine (Note 2)
Push registers to stack
Push measured pulse width to stack
Set the initial value to timer X. (Note 1)
“FF16”
“FF16”
Timer X (low-order) (address 2016)
Timer X (high-order) (address 2116)
Set “0” to the CNTR0 interrupt request bit.
Pop registers
RTI

Notes 1: When setting the value of timer X, set in order of low-order byte and high-order byte following.

2: The CNTR0 interrupt is “multiple factors/one vector interrupt”. When using the same vector interrupt, perform the priority process in the interrupt processing routine. Refer to section “**2.2.5 Management example of multiple factors/one vector interrupt**”.

Fig. 2.3.19 Example of control procedure

2.3.7 Real time port output control of timer X

The real time port output control is the function which outputs the set data beforehand synchronizing to the underflow of timer X.

The real time port control can be used in all modes.

The real time port starts to output data when the real time port control bit is set to “1”.

When the contents of the data store bits (bits 2, 3 of the timer X mode register) are re-written, updated data is output at the next underflow of timer X.

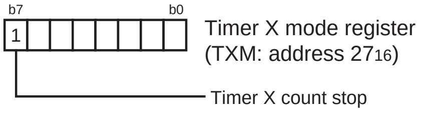
(1) Setting of real time port

Figures 2.3.20 and 2.3.21 show the setting method for the real time port output function of timer X.

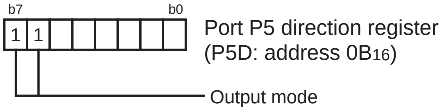
Process 1: Disable interrupts to be used. **(Note 1)**

Note 1: Refer to “2.2.3 Interrupt setting method”.

Process 2: Stop counting of timer X

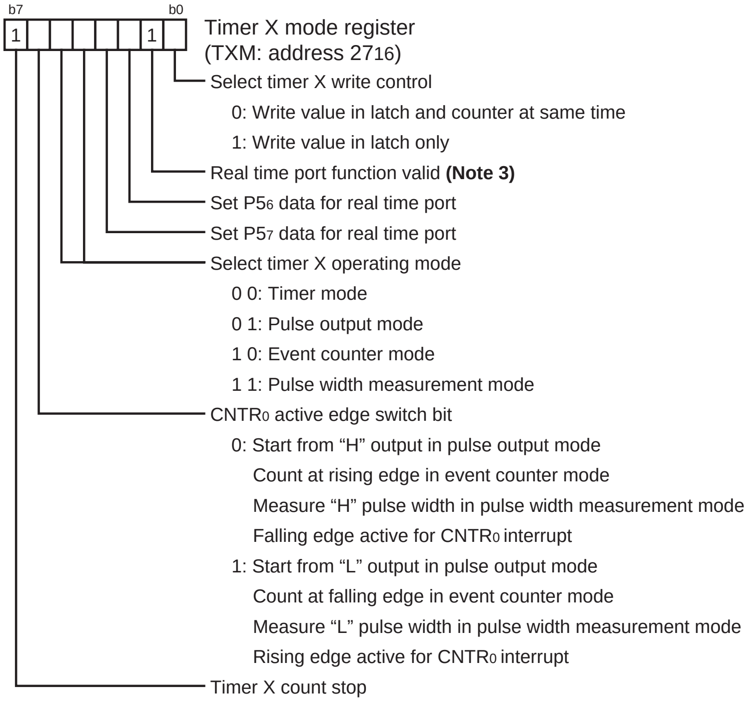


Process 3: Set the real time port as the output mode. **(Note 2)**



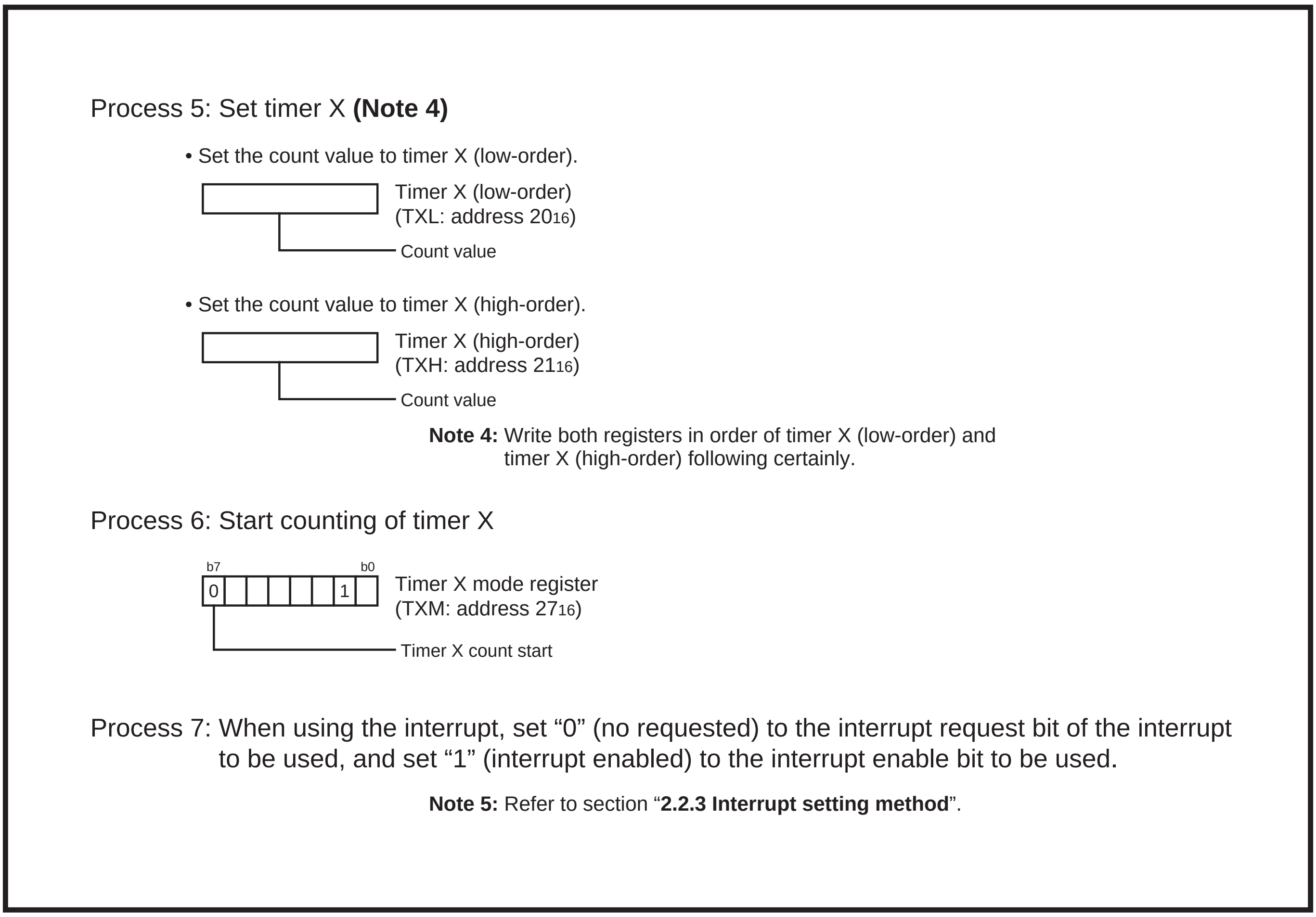
Note 2: The value of the port register is undefined at reset.

Process 4: Set timer X mode register


b7
b0
1
1
Timer X mode register
(TXM: address 27₁₆)
Select timer X write control
0: Write value in latch and counter at same time
1: Write value in latch only
Real time port function valid (Note 3)
Set P5₆ data for real time port
Set P5₇ data for real time port
Select timer X operating mode
0 0: Timer mode
0 1: Pulse output mode
1 0: Event counter mode
1 1: Pulse width measurement mode
CNTR₀ active edge switch bit
0: Start from “H” output in pulse output mode
Count at rising edge in event counter mode
Measure “H” pulse width in pulse width measurement mode
Falling edge active for CNTR₀ interrupt
1: Start from “L” output in pulse output mode
Count at falling edge in event counter mode
Measure “L” pulse width in pulse width measurement mode
Rising edge active for CNTR₀ interrupt
Timer X count stop

Note 3: Data is output from the real time port regardless of timer X operation.

Fig. 2.3.20 Setting method for real time port output function of timer X (1)



Process 5: Set timer X **(Note 4)**

- Set the count value to timer X (low-order).

- Set the count value to timer X (high-order).

Note 4: Write both registers in order of timer X (low-order) and timer X (high-order) following certainly.

Process 6: Start counting of timer X

Process 7: When using the interrupt, set “0” (no requested) to the interrupt request bit of the interrupt to be used, and set “1” (interrupt enabled) to the interrupt enable bit to be used.

Note 5: Refer to section “**2.2.3 Interrupt setting method**”.

Fig. 2.3.21 Setting method for real time port output function of timer X (2)

(2) Application examples of real time port output control

■ **Outline**

An analog output is performed by using the real time port output function.

■ **Specifications**

The pulse which is output from the real time port pin is converted into an analog voltage **(Note)** by the external circuit, and this voltage is output.

Note: The analog voltage to be output varies depending on the duty ratio of the pulse.

Figure 2.3.22 shows an example of peripheral circuit and Figure 2.3.23 shows an example of control procedure.

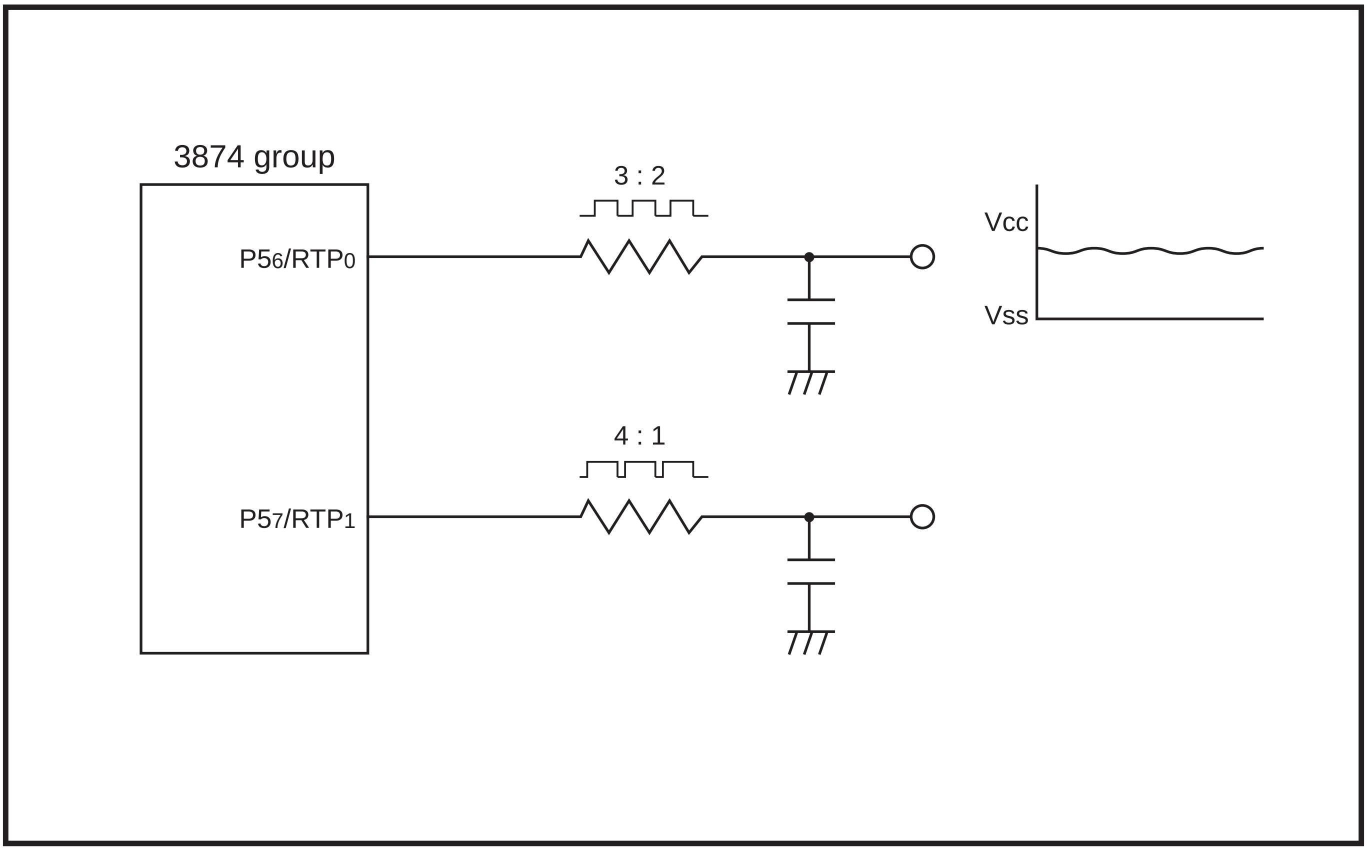
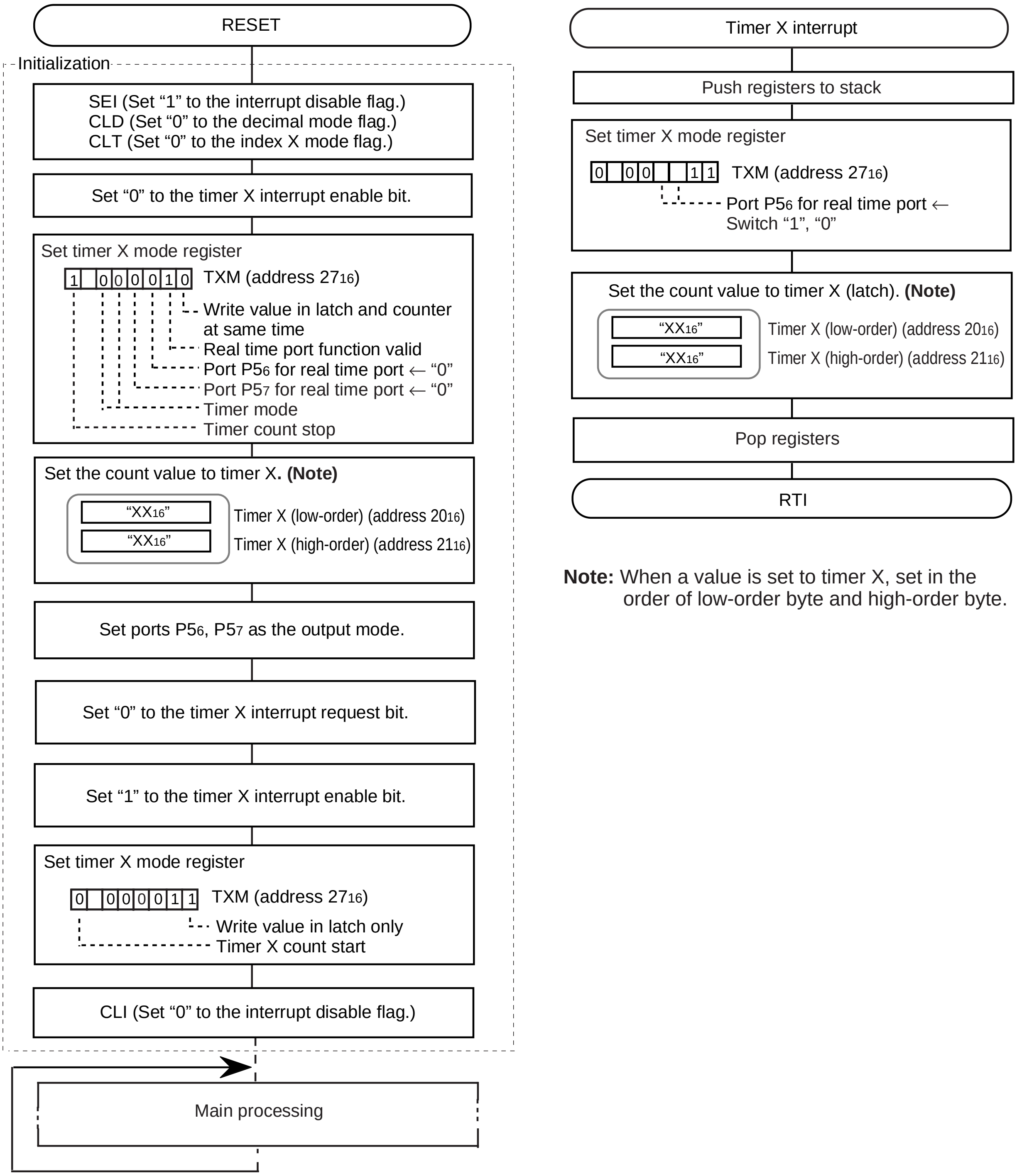


Fig. 2.3.22 Example of peripheral circuit


RESET
Initialization
SEI (Set "1" to the interrupt disable flag.)
CLD (Set "0" to the decimal mode flag.)
CLT (Set "0" to the index X mode flag.)
Set "0" to the timer X interrupt enable bit.
Set timer X mode register
1 0 0 0 0 1 0
TXM (address 27₁₆)
Write value in latch and counter at same time
Real time port function valid
Port P5₆ for real time port ← "0"
Port P5₇ for real time port ← "0"
Timer mode
Timer count stop
Set the count value to timer X. (Note)
"XX₁₆"
"XX₁₆"
Timer X (low-order) (address 20₁₆)
Timer X (high-order) (address 21₁₆)
Set ports P5₆, P5₇ as the output mode.
Set "0" to the timer X interrupt request bit.
Set "1" to the timer X interrupt enable bit.
Set timer X mode register
0 0 0 0 0 1 1
TXM (address 27₁₆)
Write value in latch only
Timer X count start
CLI (Set "0" to the interrupt disable flag.)
Main processing
Timer X interrupt
Push registers to stack
Set timer X mode register
0 0 0 1 1
TXM (address 27₁₆)
Port P5₆ for real time port ← Switch "1", "0"
Set the count value to timer X (latch). (Note)
"XX₁₆"
"XX₁₆"
Timer X (low-order) (address 20₁₆)
Timer X (high-order) (address 21₁₆)
Pop registers
RTI

Note: When a value is set to timer X, set in the order of low-order byte and high-order byte.

Fig. 2.3.23 Example of control procedure

2.3.8 Notes on use

Notes on using for each mode of timer X are described below.

(1) Common to all modes

■ **Reading/Writing of timer X**

● When reading or writing to timer X, be sure to execute to both timer X (high-order) and timer X (low-order). When reading a value from timer X, read it in order of timer X (high-order) and timer X (low-order) following. When writing a value to timer X, write in order of timer X (low-order) and timer X (high-order) following. If the following operations are performed to timer X, unexpected operation may occur.

•Write operation before execution of timer X (low-order) reading

•Read operation before execution of timer X (high-order) writing

•When writing for the latch only, if writing timing for the high-order latch is the almost same as the underflow timing, unexpected value may be set in the high-order counter.

■ **Division ratio of timer X**

When a value n (0 to FFFF$_{16}$) is written to timer X, the division ratio is 1/(n+1).

■ **Select of CNTR$_0$ interrupt active edge**

Setting the CNTR$_0$ active edge switch bit affects on the active edge of an interrupt. Consequently, a CNTR$_0$ interrupt request may occur by setting the CNTR$_0$ active edge switch bit. As a countermeasure against the above, switch the active edge after disabling the CNTR$_0$ interrupt, and set “0” to the CNTR$_0$ interrupt request bit following.

(2) Pulse output mode

● Set “1” to bit 4 (CNTR$_0$ pin) of the port P5 direction register (output mode).

● When bit 4 (CNTR$_0$ pin) of the port P5 register is read, the value of the port register are not read out, but the output value of the pin is read out.

(3) Event counter mode

● Set “0” to bit 4 (CNTR$_0$ pin) of the port P5 direction register (input mode).

(4) Pulse width measurement mode

● Set bit 4 (CNTR$_0$ pin) of the port P5 direction register to “0” (input mode).

● When reading bit 4 of port P5, the value is “1” at “H” level input or “0” at “L” level input regardless of the value of the CNTR$_0$ active edge switch bit of the timer X mode register.

(5) Real time port function

● Port P5$_6$ (RTP$_0$ pin) and port P5$_7$ (RTP$_1$ pin) function as a normal I/O port after reset released. When using ports P5$_6$ and P5$_7$ as real time port function pins, set “1” to the corresponding port direction register to be the output mode.

● Do not switch the pins which is used to the real time port to the input mode during operation.

2.3.9 Timer Y memory assignment

Figure 2.3.24 shows the memory assignment of timer Y relevant registers. Each of these registers is described below.

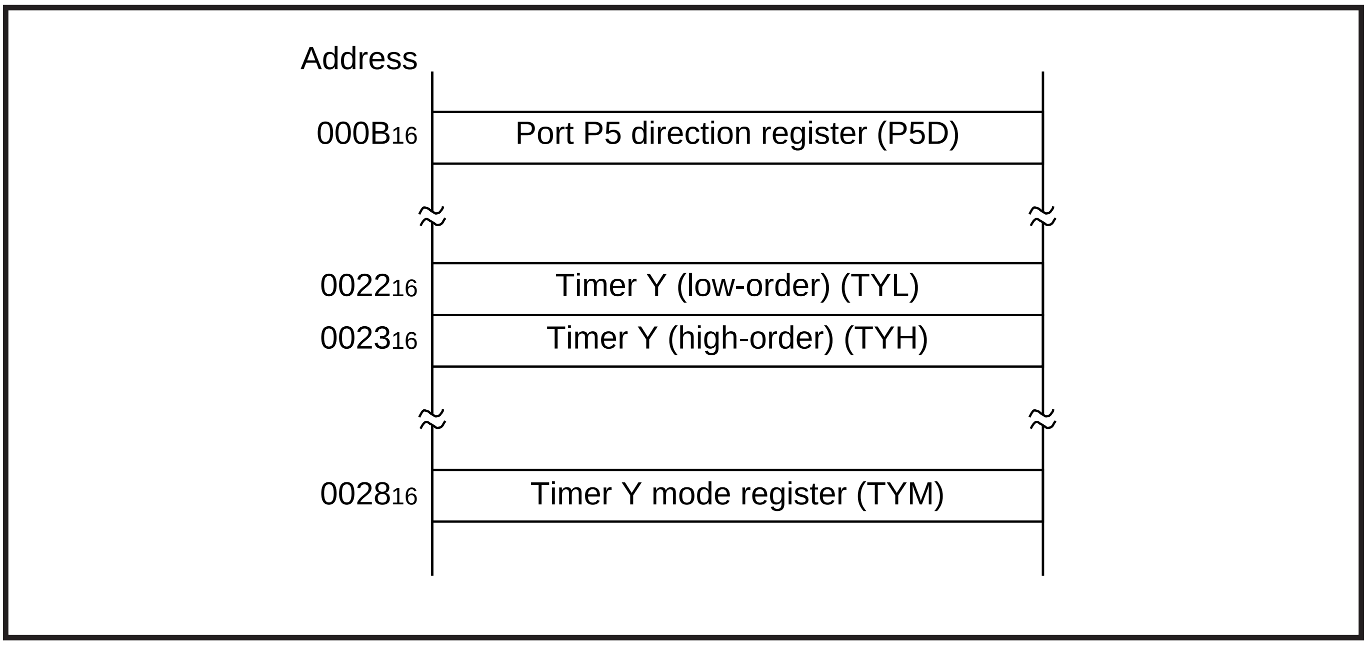

Address
$000B_{16}$
Port P5 direction register (P5D)
0022_{16}
Timer Y (low-order) (TYL)
0023_{16}
Timer Y (high-order) (TYH)
0028_{16}
Timer Y mode register (TYM)

Fig. 2.3.24 Memory assignment of timer Y relevant registers

2.3.10 Timer Y relevant registers

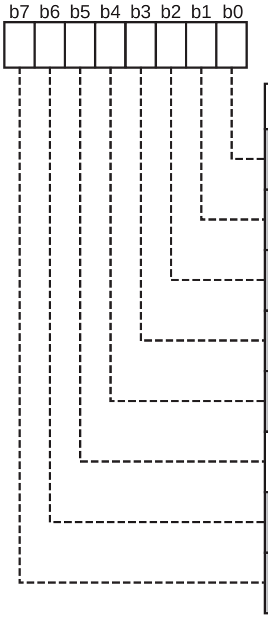
(1) Port P5 direction register

■ **Period measurement mode, event counter mode, pulse width HL continuously measurement mode**

When selecting the period measurement mode, event counter mode, or pulse width HL continuously measurement mode, set “0” to bit 5 of the port P5 direction register to set port $P5_5$ to the input mode.

Figure 2.3.25 shows the structure of the port P5 direction register.

Port P5 direction register



Port P5 direction register
(P5D: address $0B_{16}$)

B	Name	Functions	At reset	R	W
0	Port P5 direction register	0 : Port $P5_0$ input mode 1 : Port $P5_0$ output mode	0	×	○
1		0 : Port $P5_1$ input mode 1 : Port $P5_1$ output mode	0	×	○
2		0 : Port $P5_2$ input mode 1 : Port $P5_2$ output mode	0	×	○
3		0 : Port $P5_3$ input mode 1 : Port $P5_3$ output mode	0	×	○
4		0 : Port $P5_4$ input mode 1 : Port $P5_4$ output mode	0	×	○
5		0 : Port $P5_5$ input mode 1 : Port $P5_5$ output mode	0	×	○
6		0 : Port $P5_6$ input mode 1 : Port $P5_6$ output mode	0	×	○
7		0 : Port $P5_7$ input mode 1 : Port $P5_7$ output mode	0	×	○

Note : The value of the port P5 direction register cannot be read out.

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	$P5_7$	$P5_6$	$P5_5$	$P5_4$	$P5_3$	$P5_2$	$P5_1$	$P5_0$
	RTP_1	RTP_0	$CNTR_1$	$CNTR_0$	INT_5	INT_4	INT_3	T_{OUT}

Fig. 2.3.25 Structure of port P5 direction register

(2) Timer Y latch and timer Y (timer Y (low-order) and timer Y (high-order))

The timer Y latch and timer Y consist of 16 bits owing to a combination of high-order and low-order. Timer Y and the latch are allocated at the same address. To access timer Y and the latch, access both timer Y (low-order) and timer Y (high-order).

■ **Read**

When timer Y (high-order) and timer Y (low-order) are read, the contents of timer Y (count value) are read. Read both registers in order of timer Y (high-order) and timer Y (low-order) following. Do not write to timer Y (high-order) and timer Y (low-order) before timer Y (low-order) has been read. In this case, timer Y will not operate normally.

■ **Write**

When a value is written to timer Y (low-order) and timer Y (high-order), the value is set in timer Y and the latch at one time. Write both registers in order of timer Y (low-order) and timer Y (high-order) following.

Do not read timer Y (low-order) and timer Y (high-order) during a write operation. In this case, timer Y will not operate normally.

● **Timer Y latch**

The timer Y latch is a register which holds the value to be transferred automatically to timer Y as the initial value of timer Y at timer Y underflow. Transfer is performed by next factors.

•Timer Y underflow

•Valid edge input from CNTR$_1$ pin (in period measurement mode and pulse width HL continuously measurement mode)

The contents of the timer Y latch cannot be read out.

● **Timer Y**

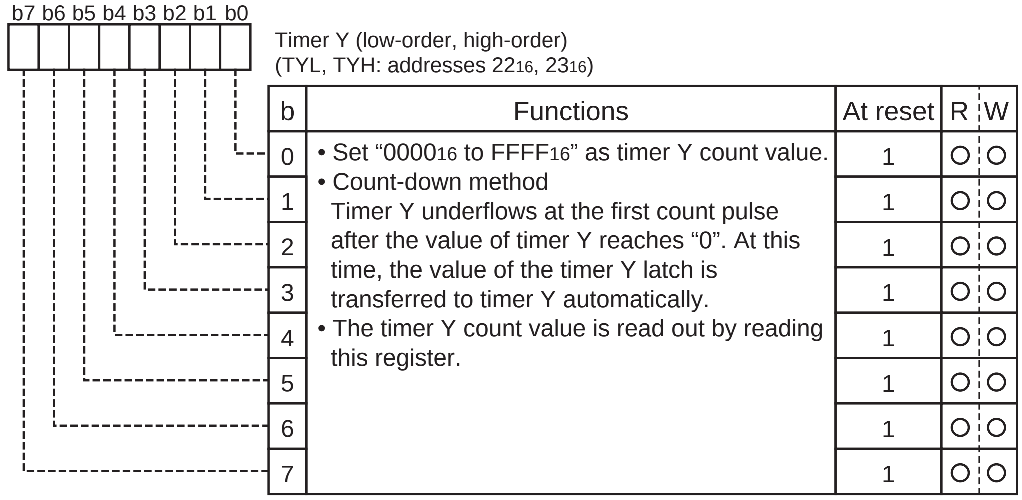
Timer Y counts the count source.

The contents of timer Y are decremented by “1” each time a count source is input. The division ratio of timer Y is expressed by the following expression.

$$\text{Division ratio of timer Y} = \frac{1}{\text{Timer Y counter initial value} + 1}$$

Figure 2.3.26 shows the structure of timer Y (low-order, high-order).

Timer Y (low-order, high-order)


b7 b6 b5 b4 b3 b2 b1 b0
Timer Y (low-order, high-order)
(TYL, TYH: addresses 22$_{16}$, 23$_{16}$)

b	Functions	At reset	R	W
0	• Set "0000$_{16}$ to FFFF$_{16}$" as timer Y count value. • Count-down method Timer Y underflows at the first count pulse after the value of timer Y reaches "0". At this time, the value of the timer Y latch is transferred to timer Y automatically. • The timer Y count value is read out by reading this register.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Notes 1: When reading and writing, perform them to both the high-order and low-order bytes.

2: Read both registers in order of TYH and TYL following.

3: Write both registers in order of TYL and TYH following.

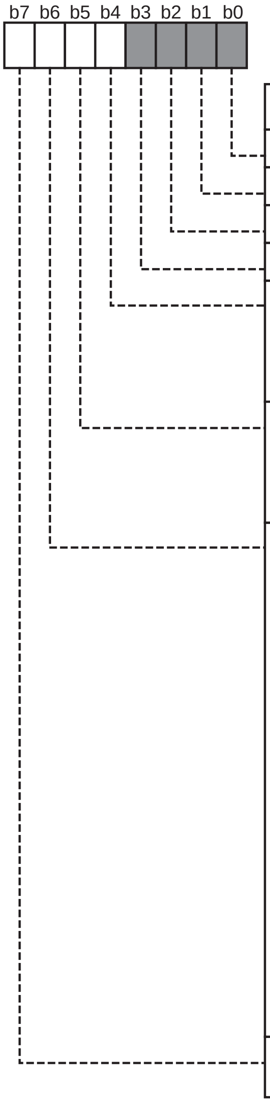
4: Do not read both registers during a write, and do not write to both registers during a read.

Fig. 2.3.26 Structure of timer Y (low-order, high-order)

(3) Timer Y mode register

The timer Y mode register consists of operation select bits and timer control bits etc. Figure 2.3.27 shows the structure of the timer Y mode register.

Timer Y mode register



Timer Y mode register
(TYM: address 28_{16})

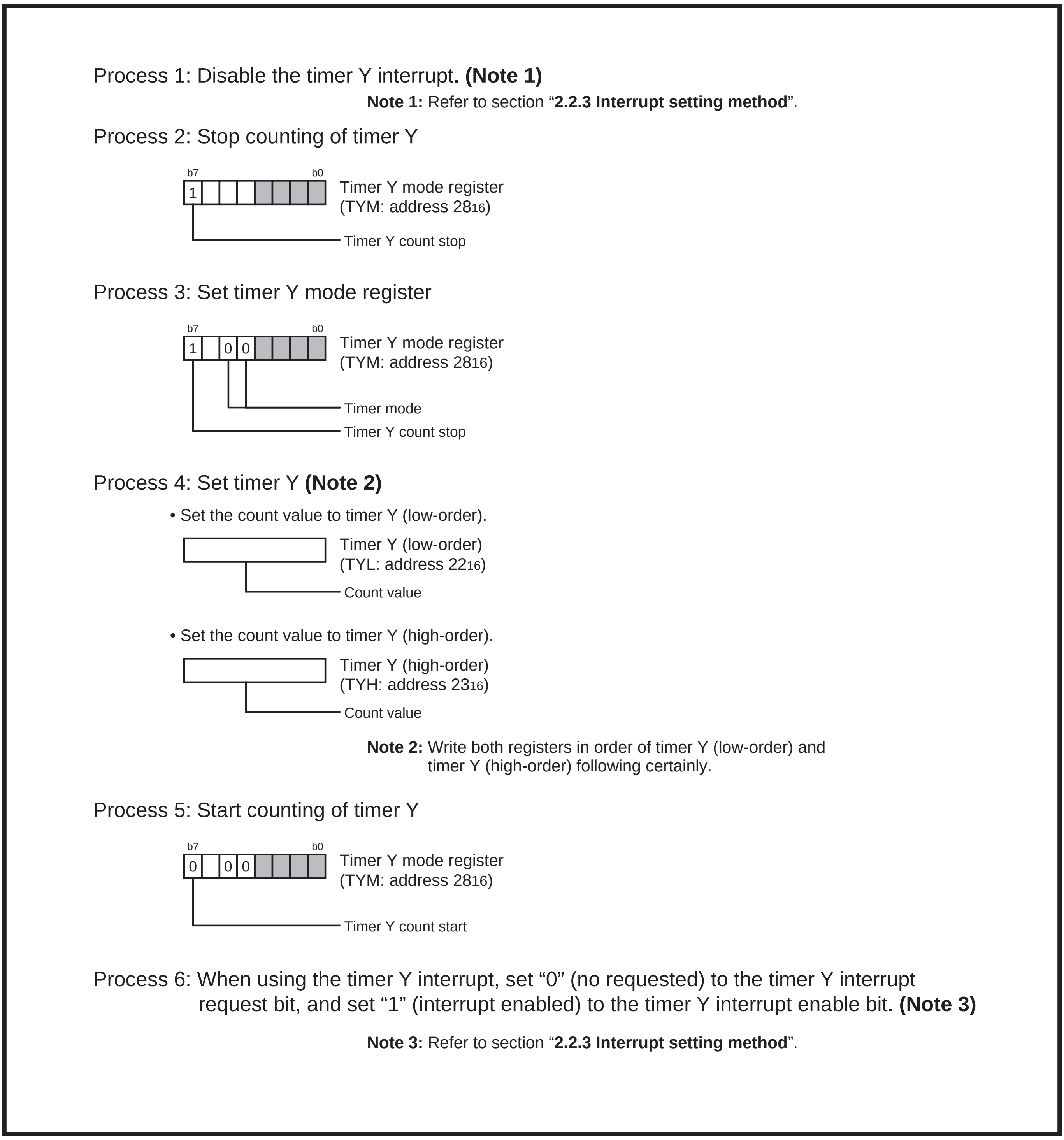
b	Name	Functions	At reset	R	W
0	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
1			0	○	×
2			0	○	×
3			0	○	×
4	Timer Y operating mode bits	b5 b4 0 0 : Timer mode 0 1 : Period measurement mode 1 0 : Event counter mode 1 1 : Pulse width HL continuously measurement mode	0	○	○
5			0	○	○
6	$CNTR_1$ active edge switch bit	0 : • Count at rising edge in event counter mode • Measure the falling edge to falling edge period in period measurement mode • Falling edge active for $CNTR_1$ interrupt 1 : • Count at falling edge in event counter mode • Measure the rising edge to rising edge period in period measurement mode • Rising edge active for $CNTR_1$ interrupt	0	○	○
7	Timer Y stop control bit	0 : Count start 1 : Count stop	0	○	○

Fig. 2.3.27 Structure of timer Y mode register

2.3.11 Timer mode of timer Y

(1) Timer mode setting method

Figure 2.3.28 shows the setting method for timer mode of timer Y.



Process 1: Disable the timer Y interrupt. **(Note 1)**

Note 1: Refer to section "**2.2.3 Interrupt setting method**".

Process 2: Stop counting of timer Y

Process 3: Set timer Y mode register

Process 4: Set timer Y **(Note 2)**

- Set the count value to timer Y (low-order).

- Set the count value to timer Y (high-order).

Note 2: Write both registers in order of timer Y (low-order) and timer Y (high-order) following certainly.

Process 5: Start counting of timer Y

Process 6: When using the timer Y interrupt, set "0" (no requested) to the timer Y interrupt request bit, and set "1" (interrupt enabled) to the timer Y interrupt enable bit. **(Note 3)**

Note 3: Refer to section "**2.2.3 Interrupt setting method**".

Fig. 2.3.28 Setting method for timer mode of timer Y

(2) Timer mode application example

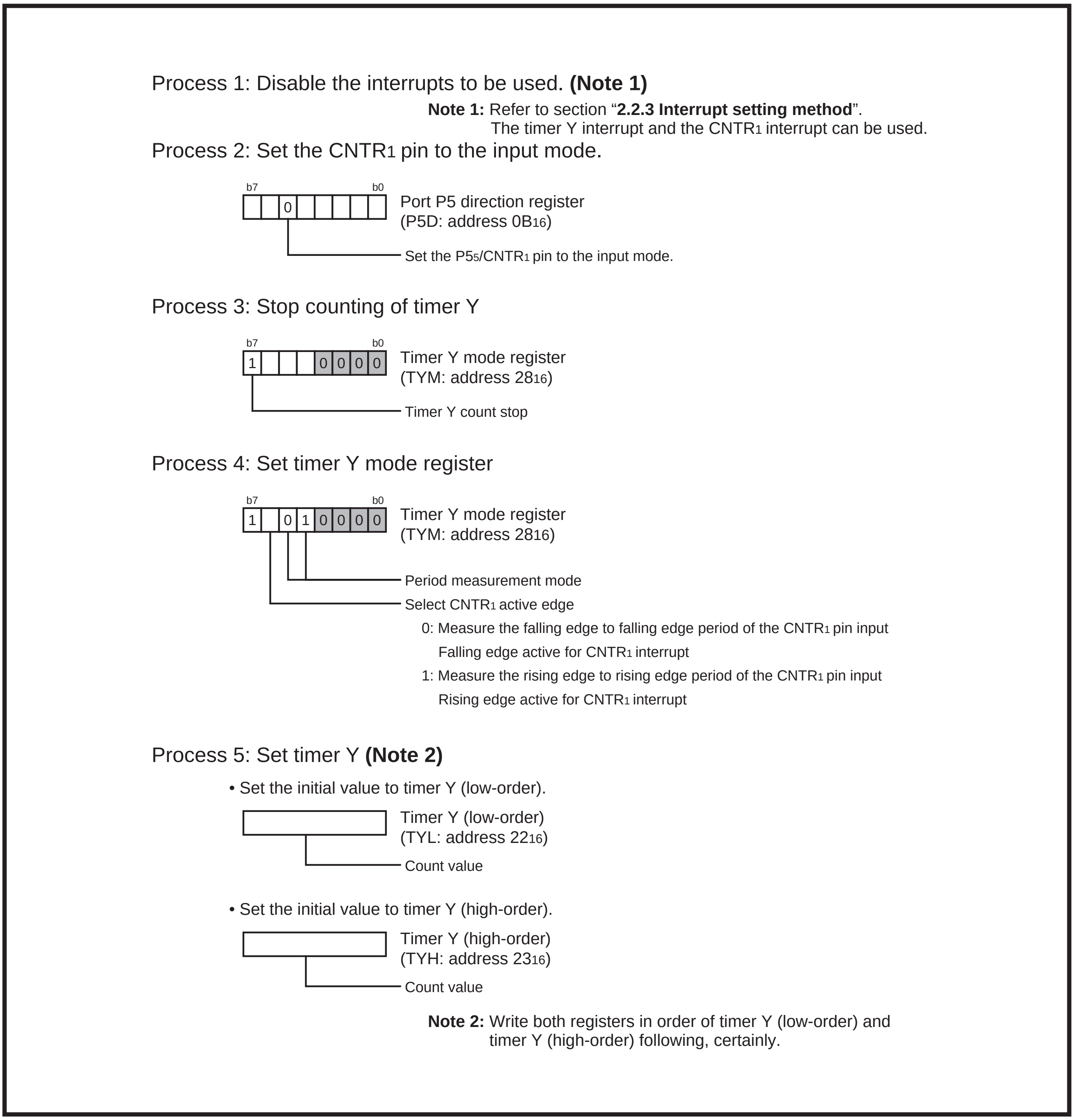
For the timer mode of timer Y application example, refer to application example for the timer mode of timer X ("**2.3.3 Timer mode of timer X (2) Timer mode application example**").

2.3.12 Period measurement mode of timer Y

The period measurement mode measures a period of the pulse which is input from the P5$_5$/CNTR$_1$ pin.

(1) Period measurement mode setting method

Figures 2.3.29 and 2.3.30 show the setting method for the period measurement mode of timer Y.



Process 1: Disable the interrupts to be used. **(Note 1)**

Note 1: Refer to section "**2.2.3 Interrupt setting method**".
The timer Y interrupt and the CNTR$_1$ interrupt can be used.

Process 2: Set the CNTR$_1$ pin to the input mode.

b7
b0
0
Port P5 direction register
(P5D: address 0B16)
Set the P55/CNTR1 pin to the input mode.

Process 3: Stop counting of timer Y

b7
b0
1
0 0 0 0
Timer Y mode register
(TYM: address 2816)
Timer Y count stop

Process 4: Set timer Y mode register

b7
b0
1
0 1 0 0 0 0
Timer Y mode register
(TYM: address 2816)
Period measurement mode
Select CNTR1 active edge
0: Measure the falling edge to falling edge period of the CNTR1 pin input
Falling edge active for CNTR1 interrupt
1: Measure the rising edge to rising edge period of the CNTR1 pin input
Rising edge active for CNTR1 interrupt

Process 5: Set timer Y **(Note 2)**

- Set the initial value to timer Y (low-order).

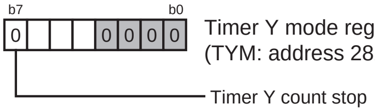
Timer Y (low-order)
(TYL: address 2216)
Count value

- Set the initial value to timer Y (high-order).

Note 2: Write both registers in order of timer Y (low-order) and timer Y (high-order) following, certainly.

Fig. 2.3.29 Setting method for period measurement mode of timer Y (1)

Process 6: Start counting of timer Y



Process 7: When using the interrupt, set “0” (no requested) to the interrupt request bit of the interrupt to be used and set “1” (interrupt enabled) to the interrupt enable bit to be used. **(Note 3)**

Note 3: Refer to section “**2.2.3 Interrupt setting method**”.

Fig. 2.3.30 Setting method for period measurement mode of timer Y (2)

(2) Application examples of period measurement mode

■ **Outline**

The phase control signal is adjusted by using the period measurement mode.

■ **Specifications**

•The phase control signal is output to a load, and that controls the phase of a load.

•The period of the pulse input from the load as a feedback signal is measured. The correct of the phase control signal to the load is corrected using this result.

Figure 2.3.31 shows an example of peripheral circuit and Figure 2.3.32 shows an example of control procedure.

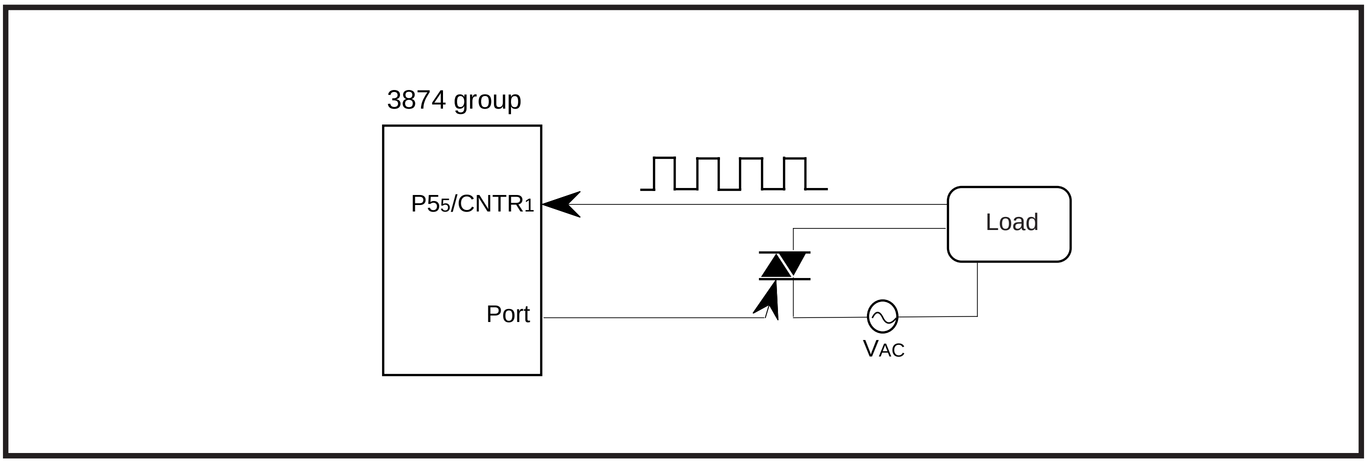
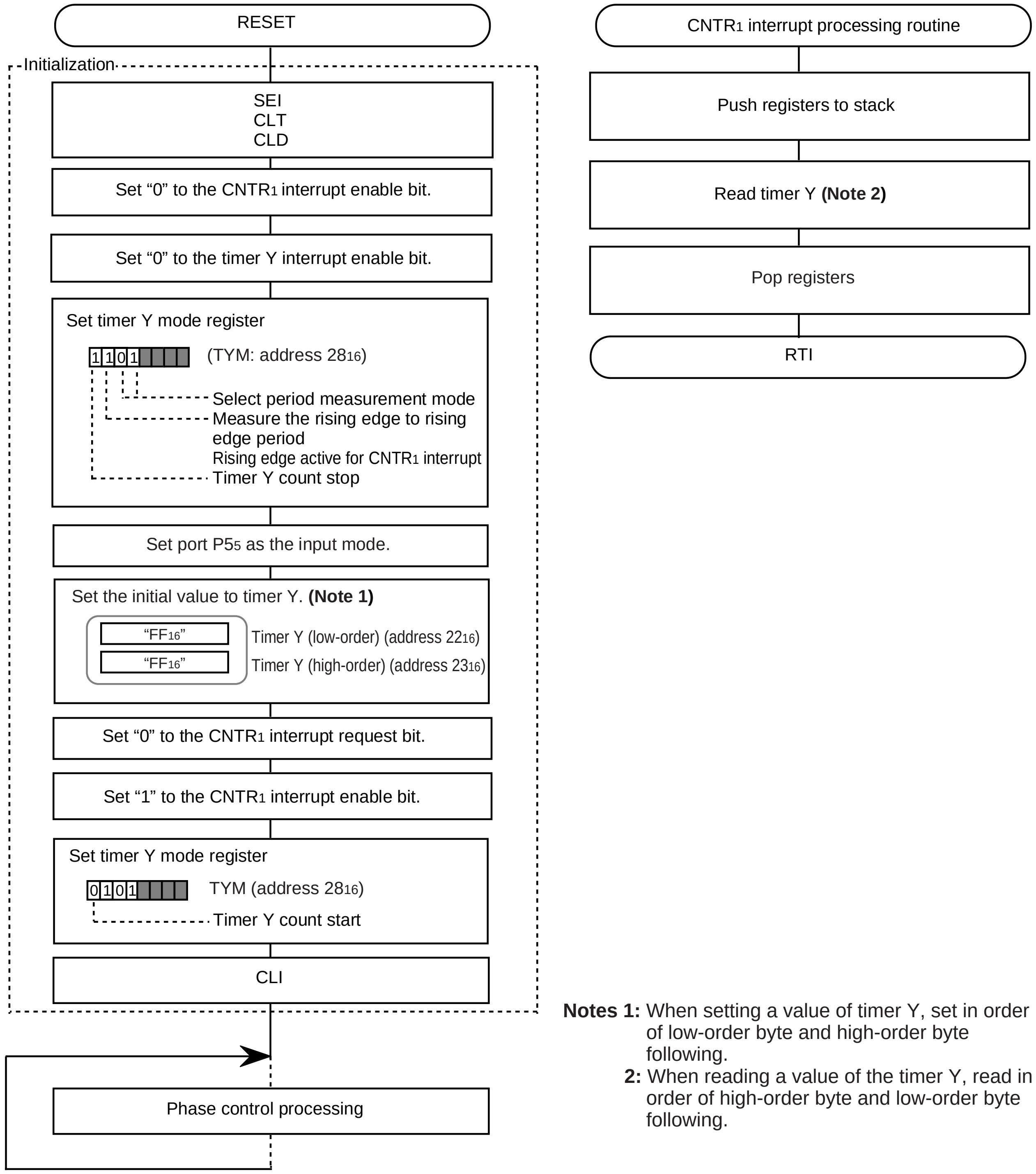


Fig. 2.3.31 Example of peripheral circuit


RESET
Initialization
SEI
CLT
CLD
Set “0” to the CNTR$_1$ interrupt enable bit.
Set “0” to the timer Y interrupt enable bit.
Set timer Y mode register
1 1 0 1
(TYM: address 28$_{16}$)
Select period measurement mode
Measure the rising edge to rising edge period
Rising edge active for CNTR$_1$ interrupt
Timer Y count stop
Set port P5$_5$ as the input mode.
Set the initial value to timer Y. (Note 1)
“FF$_{16}$”
“FF$_{16}$”
Timer Y (low-order) (address 22$_{16}$)
Timer Y (high-order) (address 23$_{16}$)
Set “0” to the CNTR$_1$ interrupt request bit.
Set “1” to the CNTR$_1$ interrupt enable bit.
Set timer Y mode register
0 1 0 1
TYM (address 28$_{16}$)
Timer Y count start
CLI
Phase control processing
CNTR$_1$ interrupt processing routine
Push registers to stack
Read timer Y (Note 2)
Pop registers
RTI

Notes 1: When setting a value of timer Y, set in order of low-order byte and high-order byte following.

2: When reading a value of the timer Y, read in order of high-order byte and low-order byte following.

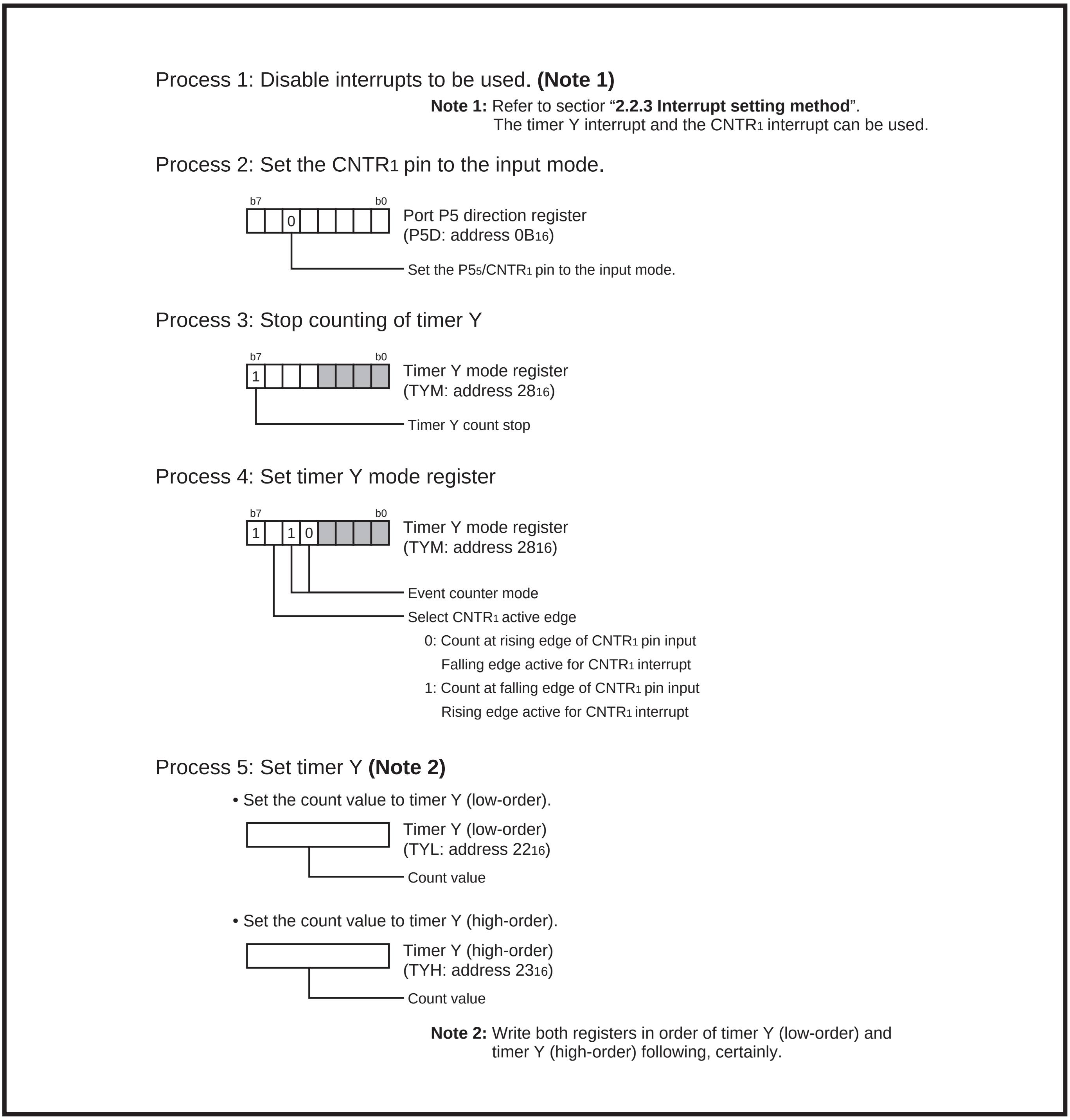
Fig. 2.3.32 Example of control procedure

2.3.13 Event counter mode of timer Y

The operation in the event counter mode is the same as that in the timer mode except input signal from the P5₅/CNTR₁ pin becomes count source.

(1) Event counter mode setting method

Figures 2.3.33 and 2.3.34 show the setting method for the event counter mode of timer Y.



Process 1: Disable interrupts to be used. **(Note 1)**

Note 1: Refer to sectior "**2.2.3 Interrupt setting method**".
The timer Y interrupt and the CNTR1 interrupt can be used.

Process 2: Set the CNTR1 pin to the input mode.

b7
b0
0
Port P5 direction register
(P5D: address 0B16)
Set the P55/CNTR1 pin to the input mode.

Process 3: Stop counting of timer Y

b7
b0
1
Timer Y mode register
(TYM: address 2816)
Timer Y count stop

Process 4: Set timer Y mode register

b7
b0
1
1
0
Timer Y mode register
(TYM: address 2816)
Event counter mode
Select CNTR1 active edge
0: Count at rising edge of CNTR1 pin input
Falling edge active for CNTR1 interrupt
1: Count at falling edge of CNTR1 pin input
Rising edge active for CNTR1 interrupt

Process 5: Set timer Y **(Note 2)**

- Set the count value to timer Y (low-order).

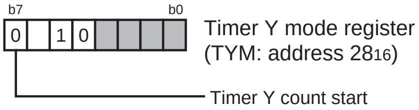
Timer Y (low-order)
(TYL: address 2216)
Count value

- Set the count value to timer Y (high-order).

Note 2: Write both registers in order of timer Y (low-order) and timer Y (high-order) following, certainly.

Fig. 2.3.33 Setting method for event counter mode of timer Y (1)

Process 6: Start counting of timer Y



Process 7: When using the interrupt, set “0” (no requested) to the interrupt request bit of the interrupt to be used and set “1” (interrupt enabled) to the interrupt enable bit to be used. **(Note 3)**

Note 3: Refer to section “**2.2.3 Interrupt setting method**”.

Fig. 2.3.34 Setting method for event counter mode of timer Y (2)

(2) Event counter mode application example

For the event counter mode of timer Y application example, refer to application example for the event counter mode of timer X (“**2.3.5 Event counter mode of timer X (2) Application examples of event counter mode**”).

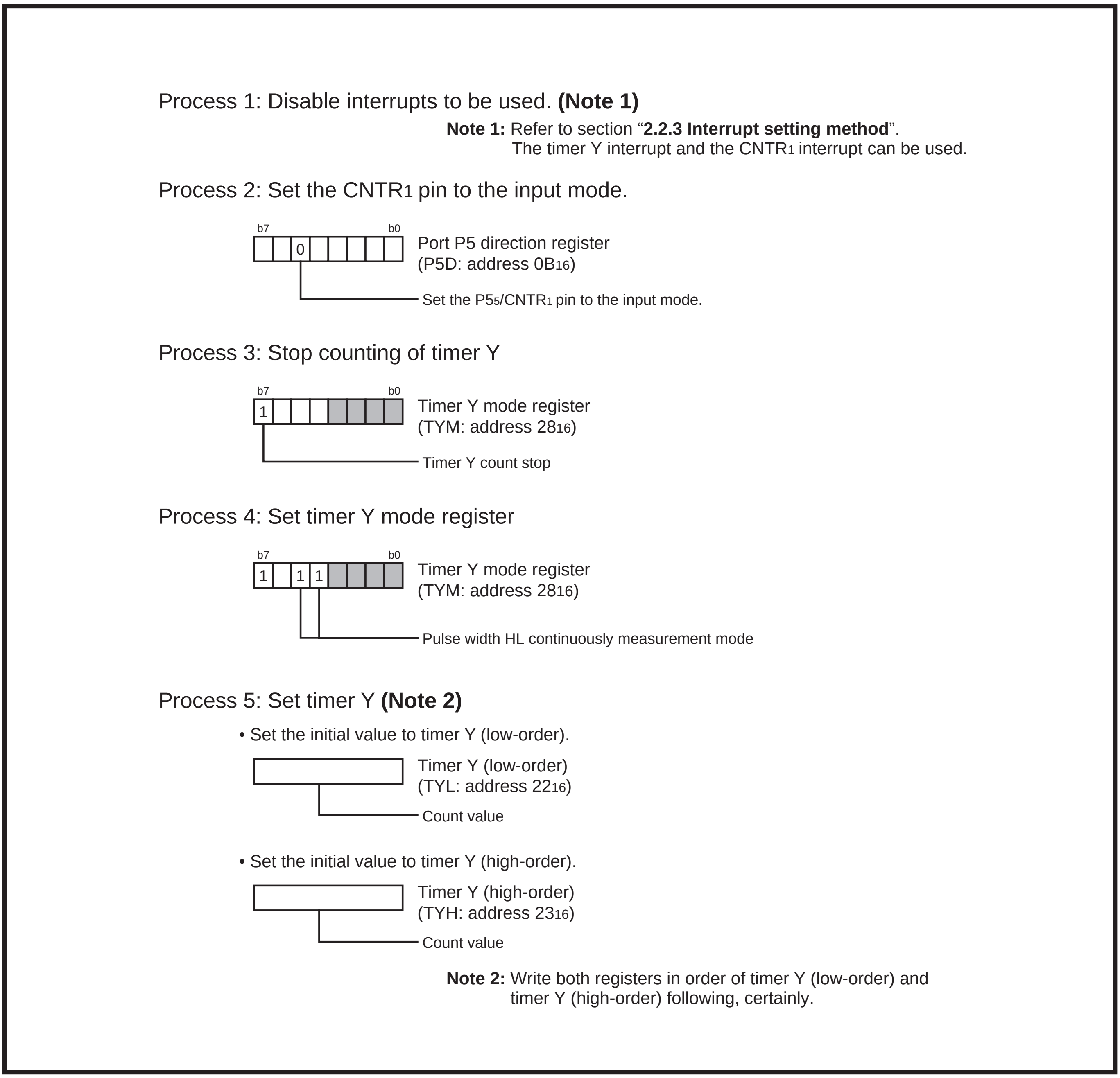
2.3.14 Pulse width HL continuously measurement mode of timer Y

The pulse width HL continuously measurement mode, measures the pulse width (“H” and “L” level) of the P5$_5$/CNTR$_1$ pin input signal continuously.

This mode operates just as in the period measurement mode except that the value in timer Y latch is reloaded in timer Y and the interrupt request occurs at both edges of the pulse which is input from the P5$_5$/CNTR$_1$ pin.

(1) Pulse width HL continuously measurement mode setting method

Figures 2.3.35 and 2.3.36 show the setting method for the pulse width HL continuously measurement mode of timer Y.



Process 1: Disable interrupts to be used. **(Note 1)**

Note 1: Refer to section “**2.2.3 Interrupt setting method**”.
The timer Y interrupt and the CNTR1 interrupt can be used.

Process 2: Set the CNTR1 pin to the input mode.

Port P5 direction register
(P5D: address 0B16)

Set the P55/CNTR1 pin to the input mode.

Process 3: Stop counting of timer Y

Timer Y mode register
(TYM: address 2816)

Timer Y count stop

Process 4: Set timer Y mode register

Timer Y mode register
(TYM: address 2816)

Pulse width HL continuously measurement mode

Process 5: Set timer Y **(Note 2)**

- Set the initial value to timer Y (low-order).

Timer Y (low-order)
(TYL: address 2216)

Count value

- Set the initial value to timer Y (high-order).

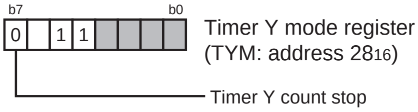
Timer Y (high-order)
(TYH: address 2316)

Count value

Note 2: Write both registers in order of timer Y (low-order) and timer Y (high-order) following, certainly.

Fig. 2.3.35 Setting method for pulse width HL continuously measurement mode of timer Y (1)

Process 6: Start counting of timer Y



Process 7: When using the interrupt, set “0” (no requested) to the interrupt request bit of the interrupt to be used and set “1” (interrupt enabled) to the interrupt enable bit to be used. **(Note 3)**

Note 3: Refer to section **“2.2.3 Interrupt setting method”.** In the pulse width HL continuously measurement mode, the $CNTR_1$ interrupt request occurs at the rising edge and falling edge of the $P5_5/CNTR_1$ pin regardless of the value of the $P5_5/CNTR_1$ active edge switch bit of the timer Y mode register.

Fig. 2.3.36 Setting method for pulse width HL continuously measurement mode of timer Y (2)

(2) Pulse width HL continuously measurement mode application example

■ **Outline**

A telephone ringing (calling) pulse * is detected by using the pulse width HL continuously measurement mode.

✻: Signal which is sent by turning on/off (make/break) the telephone line.
Each country has a different standard. In this case, Japanese domestic standard is adopted as an example.

■ **Specifications**

Whether a telephone call exists or not is judged by measuring a pulse width output from the ringing signal detection circuit.

$f(X_{IN})/16$ ($f(X_{IN}) = 6.4$ MHz) is used as the count source, and “H” and “L” pulse width of the ringing pulse are measured by using the pulse width HL continuously measurement mode. When the following condition is satisfied, it is recognized as a normal value. When the following conditions is not satisfied, it is recognized as an unusual value.

200 ms ≤ “H” pulse width of ringing pulse < 1.2 s
600 ms ≤ “L” pulse width of ringing pulse < 2.2 s
1.0 s ≤ one period (“H” pulse width + “L” pulse width) < 3.0 s

Figure 2.3.37 shows an example of a peripheral circuit, and Figure 2.3.38 shows an operating timing when a ringing pulse is input. Figures 2.3.39 and 2.3.40 show a example of control procedure.

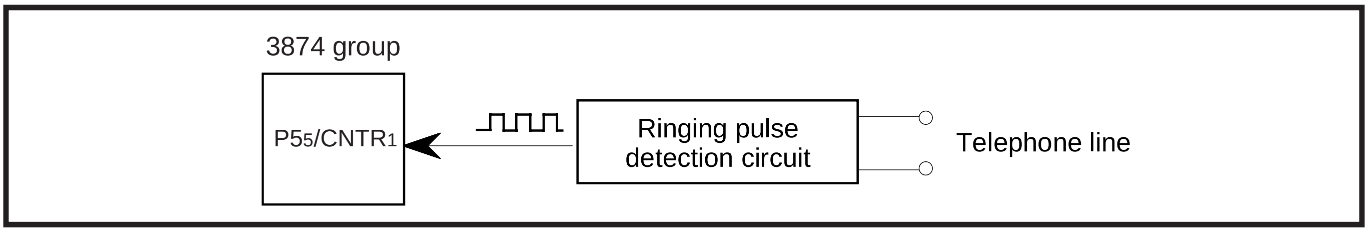


Fig. 2.3.37 Example of peripheral circuit

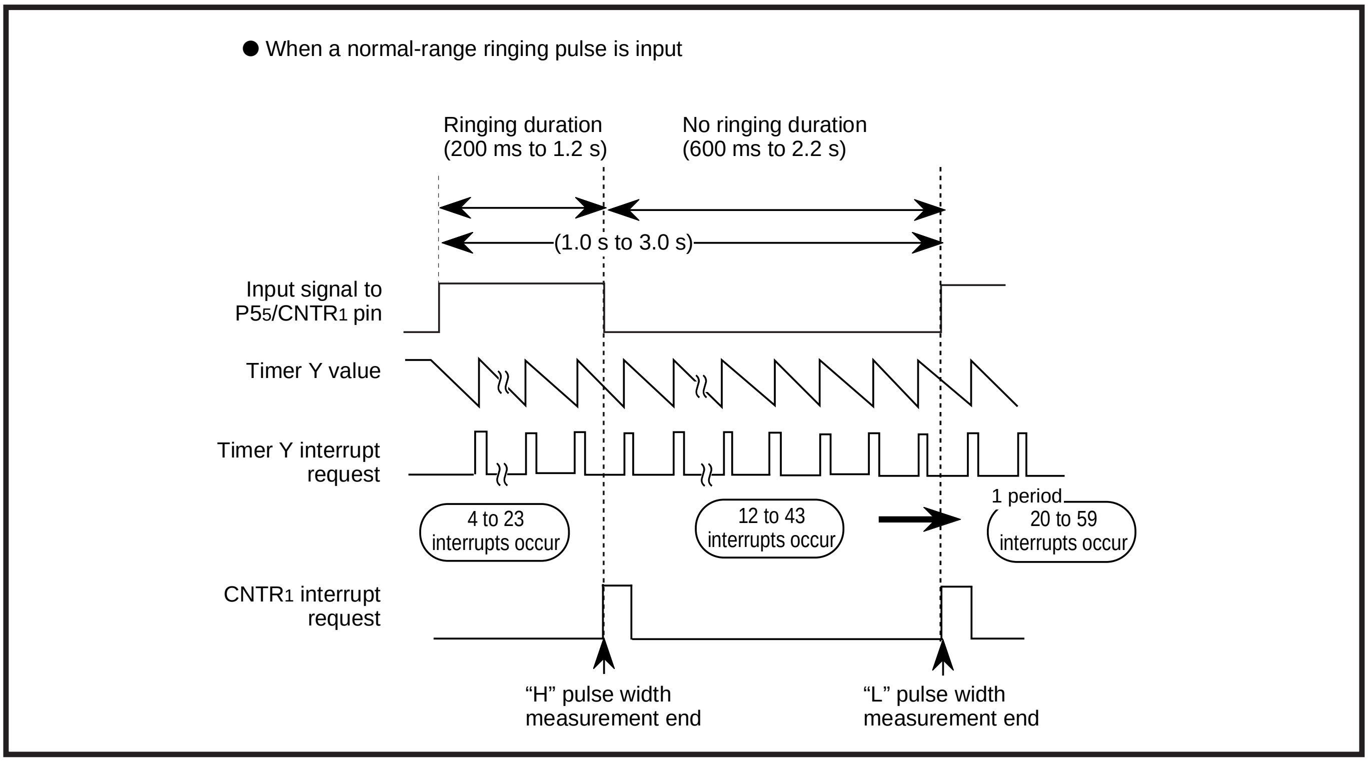

● When a normal-range ringing pulse is input
Ringing duration (200 ms to 1.2 s)
No ringing duration (600 ms to 2.2 s)
(1.0 s to 3.0 s)
Input signal to P55/CNTR1 pin
Timer Y value
Timer Y interrupt request
4 to 23 interrupts occur
12 to 43 interrupts occur
1 period
20 to 59 interrupts occur
CNTR1 interrupt request
“H” pulse width measurement end
“L” pulse width measurement end

Fig. 2.3.38 Operation timing when ringing pulse is input

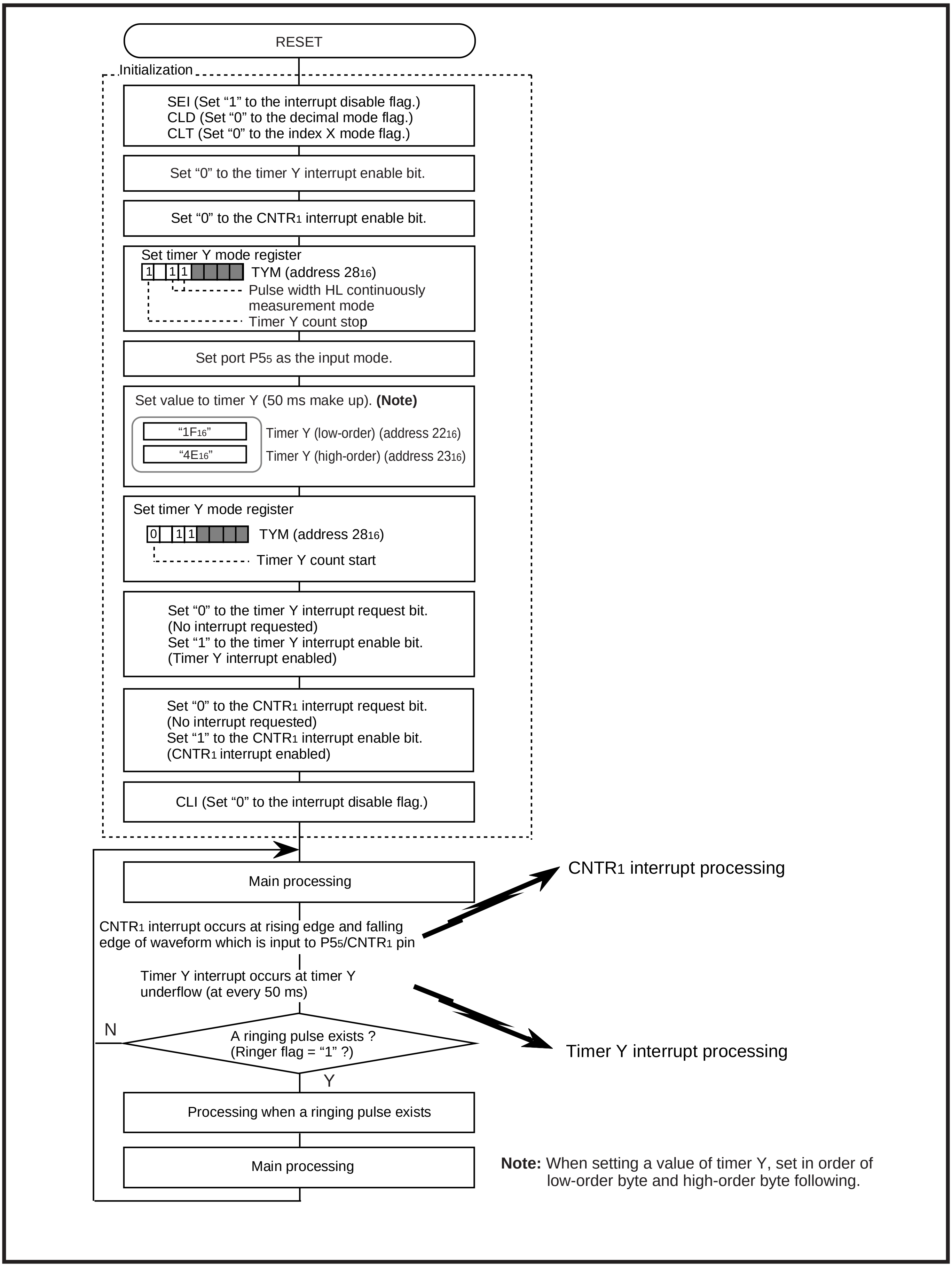

RESET
Initialization
SEI (Set "1" to the interrupt disable flag.)
CLD (Set "0" to the decimal mode flag.)
CLT (Set "0" to the index X mode flag.)
Set "0" to the timer Y interrupt enable bit.
Set "0" to the CNTR1 interrupt enable bit.
Set timer Y mode register
1 1 1
TYM (address 2816)
Pulse width HL continuously measurement mode
Timer Y count stop
Set port P55 as the input mode.
Set value to timer Y (50 ms make up). (Note)
"1F16"
Timer Y (low-order) (address 2216)
"4E16"
Timer Y (high-order) (address 2316)
Set timer Y mode register
0 1 1
TYM (address 2816)
Timer Y count start
Set "0" to the timer Y interrupt request bit. (No interrupt requested)
Set "1" to the timer Y interrupt enable bit. (Timer Y interrupt enabled)
Set "0" to the CNTR1 interrupt request bit. (No interrupt requested)
Set "1" to the CNTR1 interrupt enable bit. (CNTR1 interrupt enabled)
CLI (Set "0" to the interrupt disable flag.)
Main processing
CNTR1 interrupt processing
CNTR1 interrupt occurs at rising edge and falling edge of waveform which is input to P55/CNTR1 pin
Timer Y interrupt occurs at timer Y underflow (at every 50 ms)
N
A ringing pulse exists ? (Ringer flag = "1" ?)
Y
Timer Y interrupt processing
Processing when a ringing pulse exists
Main processing
Note: When setting a value of timer Y, set in order of low-order byte and high-order byte following.

Fig. 2.3.39 Example of control procedure (1)

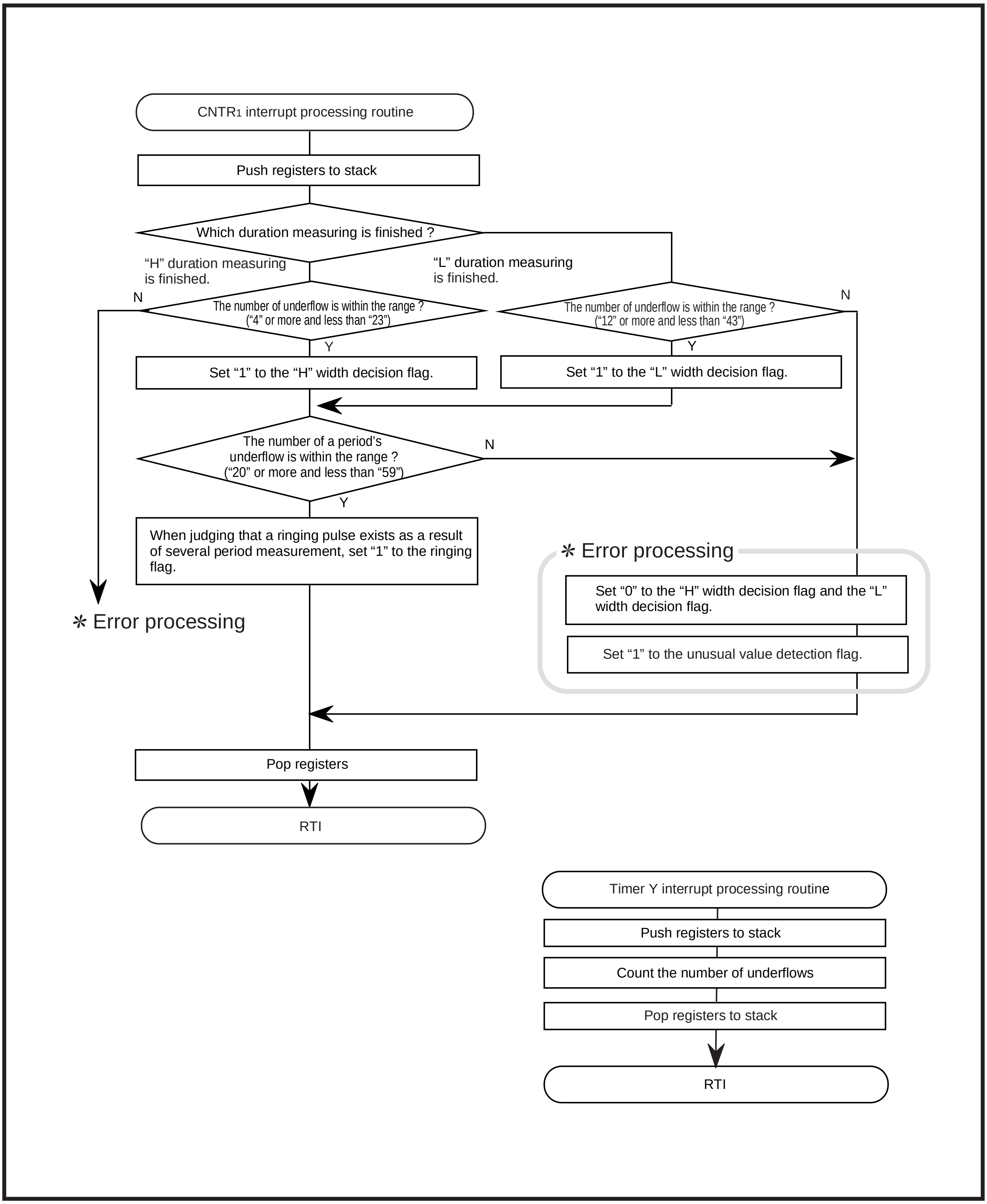

CNTR1 interrupt processing routine
Push registers to stack
Which duration measuring is finished ?
"H" duration measuring is finished.
"L" duration measuring is finished.
N
The number of underflow is within the range ? ("4" or more and less than "23")
Y
The number of underflow is within the range ? ("12" or more and less than "43")
N
Y
Set "1" to the "H" width decision flag.
Set "1" to the "L" width decision flag.
The number of a period's underflow is within the range ? ("20" or more and less than "59")
N
Y
When judging that a ringing pulse exists as a result of several period measurement, set "1" to the ringing flag.
✻ Error processing
✻ Error processing
Set "0" to the "H" width decision flag and the "L" width decision flag.
Set "1" to the unusual value detection flag.
Pop registers
RTI
Timer Y interrupt processing routine
Push registers to stack
Count the number of underflows
Pop registers to stack
RTI

Fig. 2.3.40 Example of control procedure (2)

2.3.15 Notes on use

Notes on using each mode of timer Y are described below.

(1) Common to all modes

- ● When reading or writing to timer Y, be sure to execute to both timer Y (high-order) and timer Y (low-order).
- ● When reading a value from timer Y, read it in order of timer Y (high-order) and timer Y (low-order) following. When writing a value to timer Y, write in order of timer Y (low-order) and timer Y (high-order) following.
- ● If the following operations are performed to timer Y, unexpected operation may occur.
  •Write operation before execution of timer Y (low-order) reading
  •Read operation before execution of timer Y (high-order) writing

■ **Division ratio of timer Y**

When a value n (0 to $FFFF_{16}$) is written to timer Y, the division ratio is 1/(n+1).

■ **Select of $CNTR_1$ interrupt active edge**

Setting the $CNTR_1$ active edge switch bit affects the active edge of an interrupt. Consequently, a $CNTR_1$ interrupt request may occur by setting the $CNTR_1$ active edge switch bit. As a countermeasure against the above, switch the active edge after disabling “0” to the $CNTR_1$ interrupt, and set “0” to the $CNTR_1$ interrupt request bit.

(2) Period measurement mode

- ● Set “0” to bit 5 ($CNTR_1$ pin) of the port P5 direction register (input mode).

(3) Event counter mode

- ● Set “0” to bit 5 ($CNTR_1$ pin) of the port P5 direction register (input mode).

(4) Pulse width HL continuously measurement mode

- ● Set “0” to bit 5 ($CNTR_1$ pin) of the port P5 direction register to be the input mode.
- ● The $CNTR_1$ interrupt request occurs at both edges of a input pulse regardless of the contents of the $CNTR_1$ active edge switch bit of the timer Y mode register.

2.4 Timers 1, 2, 3

2.4.1 Timers 1, 2, 3 memory assignment

Figure 2.4.1 shows the memory assignment of timers 1, 2, 3 relevant registers.

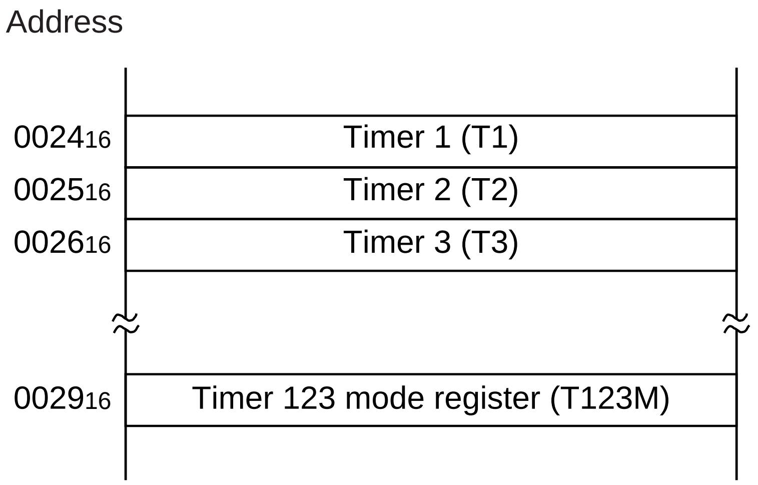


Fig. 2.4.1 Memory assignment of timers 1, 2, 3 relevant registers

2.4.2 Timers 1, 2, 3 relevant registers

(1) Timer i latch and timer i (i = 1 to 3)

Timer i latch and timer i consist of 8 bits each and are assigned at the same address to each timer. To access timer i latch and timer i, access timer i (i = 1 to 3). The contents of timer i (count value) are read out.

■ Timer i latch

The timer i latch holds the value to be transferred (reloaded) automatically to timer i as the initial value of timer i at timer i underflow. The contents of the timer i latch cannot be read out.

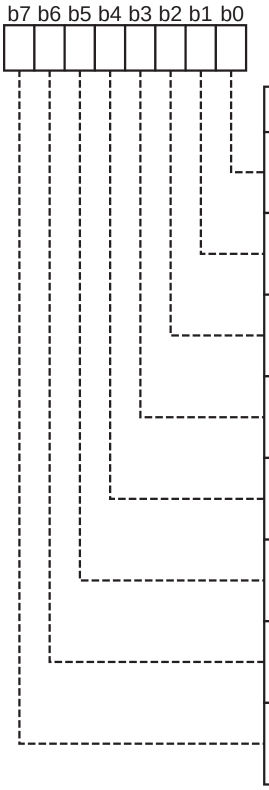
■ Timer i

Timer i counts the selected count source.
The contents of timer i are decremented by “1” each time a count source inputs.

$$\text{Division ratio of timer} = \frac{1}{\text{Timer initial value} + 1}$$

Figure 2.4.2 shows the structure of timer i (i = 1 to 3).

Timer i (i = 1 to 3)



Timer 1 (T1) (address 24_{16})
Timer 2 (T2) (address 25_{16})
Timer 3 (T3) (address 26_{16})

b	Functions	At reset	R	W
0	• Set “00_{16}” to “FF_{16}” as timer i count value. • Timer 1 and timer 3 A set value is written to the timer latch and timer. • Timer 2 When the timer 2 write control bit of the timer 123 mode register is “0”, a set value is written to the latch and timer at one time. When the timer 2 write control bit of the timer 123 mode register is “1”, a set value is written to the latch only. • Count-down method The count value is read out by reading timer i. • Timer i underflows at the first count pulse after the value of timer i reaches “0”. At this time, the value of the timer i latch is transferred to timer i automatically. • When reading this register, the count value of corresponding timer is read out.	**(Note 1)**	○	○
1			○	○
2			○	○
3			○	○
4			○	○
5			○	○
6			○	○
7			○	○

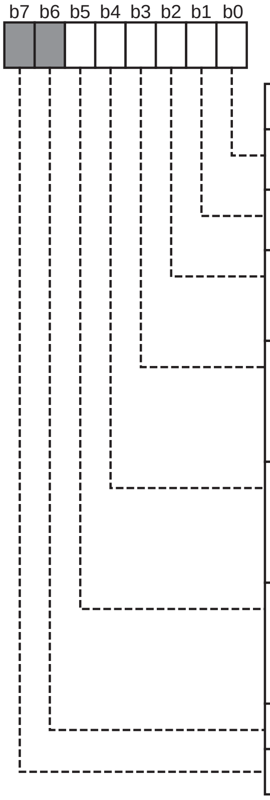
Notes 1: In timer 1 and timer 3, it becomes “FF_{16}”. In timer 2, it becomes “01_{16}”.
2: When switching the count source, a value of timer i may become an inaccurate value.
Set timers again in order of timer 1, timer 2, and timer 3.

Fig. 2.4.2 Structure of timer i (i = 1 to 3)

(2) Timer 123 mode register

The timer 123 mode register consists of the T_{OUT} output control bit and the count source selection bits etc. Figure 2.4.3 shows the structure of the timer 123 mode register.

Timer 123 mode register



Timer 123 mode register
(T123M: address 29_{16})

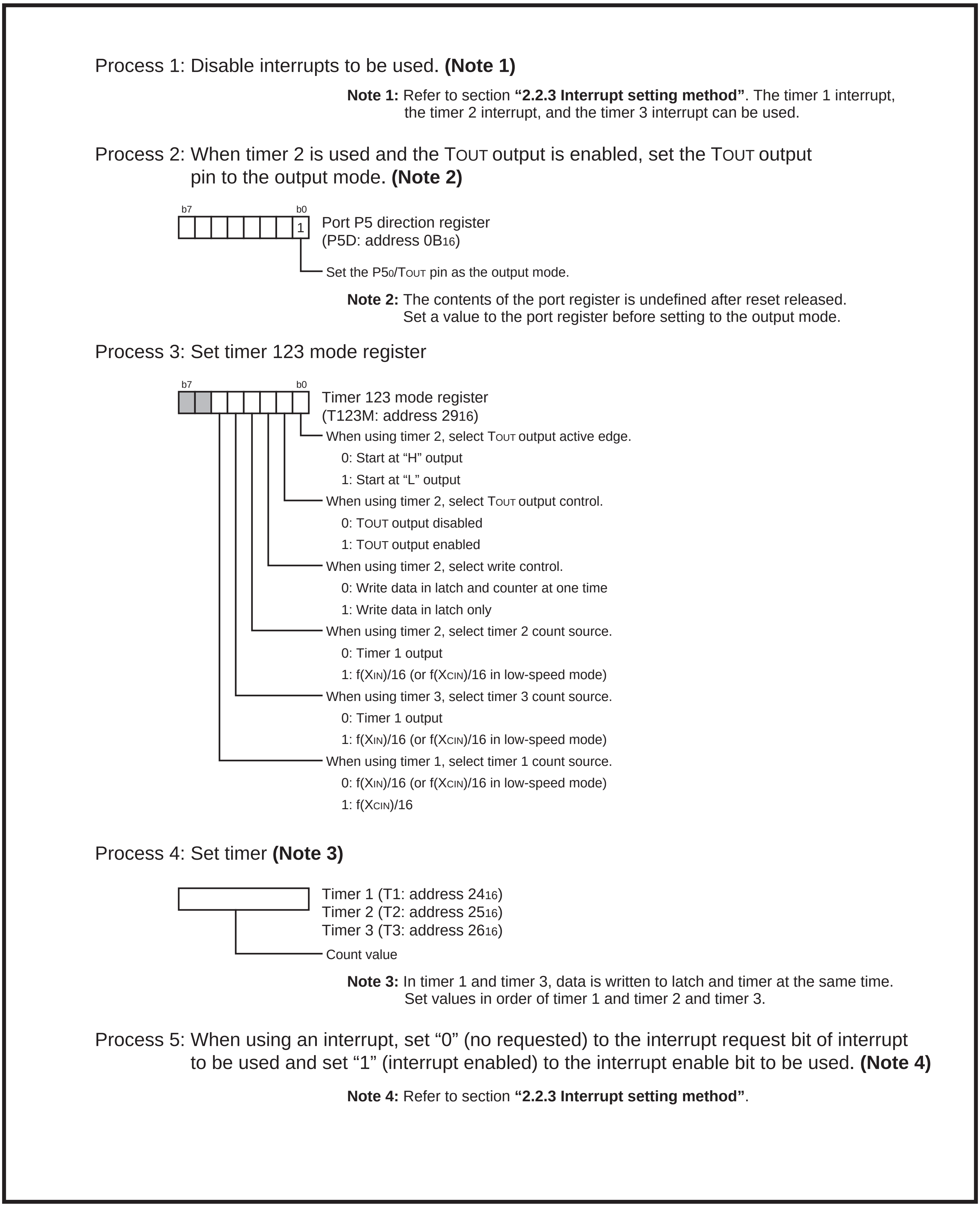
b	Name	Functions	At reset	R	W
0	T_{OUT} output active edge switch bit	0 : Start at “H” output 1 : Start at “L” output	0	○	○
1	T_{OUT} output control bit	0 : T_{OUT} output disabled 1 : T_{OUT} output enabled	0	○	○
2	Timer 2 write control bit	0 : Write data in latch and counter at one time 1 : Write data in latch only	0	○	○
3	Timer 2 count source selection bit	0 : Timer 1 output 1 : f(X_{IN})/16 (or f(X_{CIN})/16 in low-speed mode) **(Note)**	0	○	○
4	Timer 3 count source selection bit	0 : Timer 1 output 1 : f(X_{IN})/16 (or f(X_{CIN})/16 in low-speed mode) **(Note)**	0	○	○
5	Timer 1 count source selection bit	1 : f(X_{IN})/16 (or f(X_{CIN})/16 in low-speed mode) **(Note)** 0 : f(X_{CIN})	0	○	○
6	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
7			0	○	×

Note: In the low-speed mode, the internal clock ϕ is X_{IN} pin input signal divided by 2.

Fig. 2.4.3 Structure of timer 123 mode register

2.4.3 Timers 1, 2, 3 setting method

Figure 2.4.4 shows the setting method of timers 1, 2, 3.



Process 1: Disable interrupts to be used. **(Note 1)**

Note 1: Refer to section **“2.2.3 Interrupt setting method”**. The timer 1 interrupt, the timer 2 interrupt, and the timer 3 interrupt can be used.

Process 2: When timer 2 is used and the T_{OUT} output is enabled, set the T_{OUT} output pin to the output mode. **(Note 2)**

Port P5 direction register
(P5D: address $0B_{16}$)

Set the $P5_0/T_{OUT}$ pin as the output mode.

Note 2: The contents of the port register is undefined after reset released. Set a value to the port register before setting to the output mode.

Process 3: Set timer 123 mode register

Timer 123 mode register
(T123M: address 29_{16})

When using timer 2, select T_{OUT} output active edge.
0: Start at “H” output
1: Start at “L” output

When using timer 2, select T_{OUT} output control.
0: T_{OUT} output disabled
1: T_{OUT} output enabled

When using timer 2, select write control.
0: Write data in latch and counter at one time
1: Write data in latch only

When using timer 2, select timer 2 count source.
0: Timer 1 output
1: $f(X_{IN})/16$ (or $f(X_{CIN})/16$ in low-speed mode)

When using timer 3, select timer 3 count source.
0: Timer 1 output
1: $f(X_{IN})/16$ (or $f(X_{CIN})/16$ in low-speed mode)

When using timer 1, select timer 1 count source.
0: $f(X_{IN})/16$ (or $f(X_{CIN})/16$ in low-speed mode)
1: $f(X_{CIN})/16$

Process 4: Set timer **(Note 3)**

Timer 1 (T1: address 24_{16})
Timer 2 (T2: address 25_{16})
Timer 3 (T3: address 26_{16})

Count value

Note 3: In timer 1 and timer 3, data is written to latch and timer at the same time. Set values in order of timer 1 and timer 2 and timer 3.

Process 5: When using an interrupt, set “0” (no requested) to the interrupt request bit of interrupt to be used and set “1” (interrupt enabled) to the interrupt enable bit to be used. **(Note 4)**

Note 4: Refer to section **“2.2.3 Interrupt setting method”**.

Fig. 2.4.4 Setting method for timers 1, 2, 3

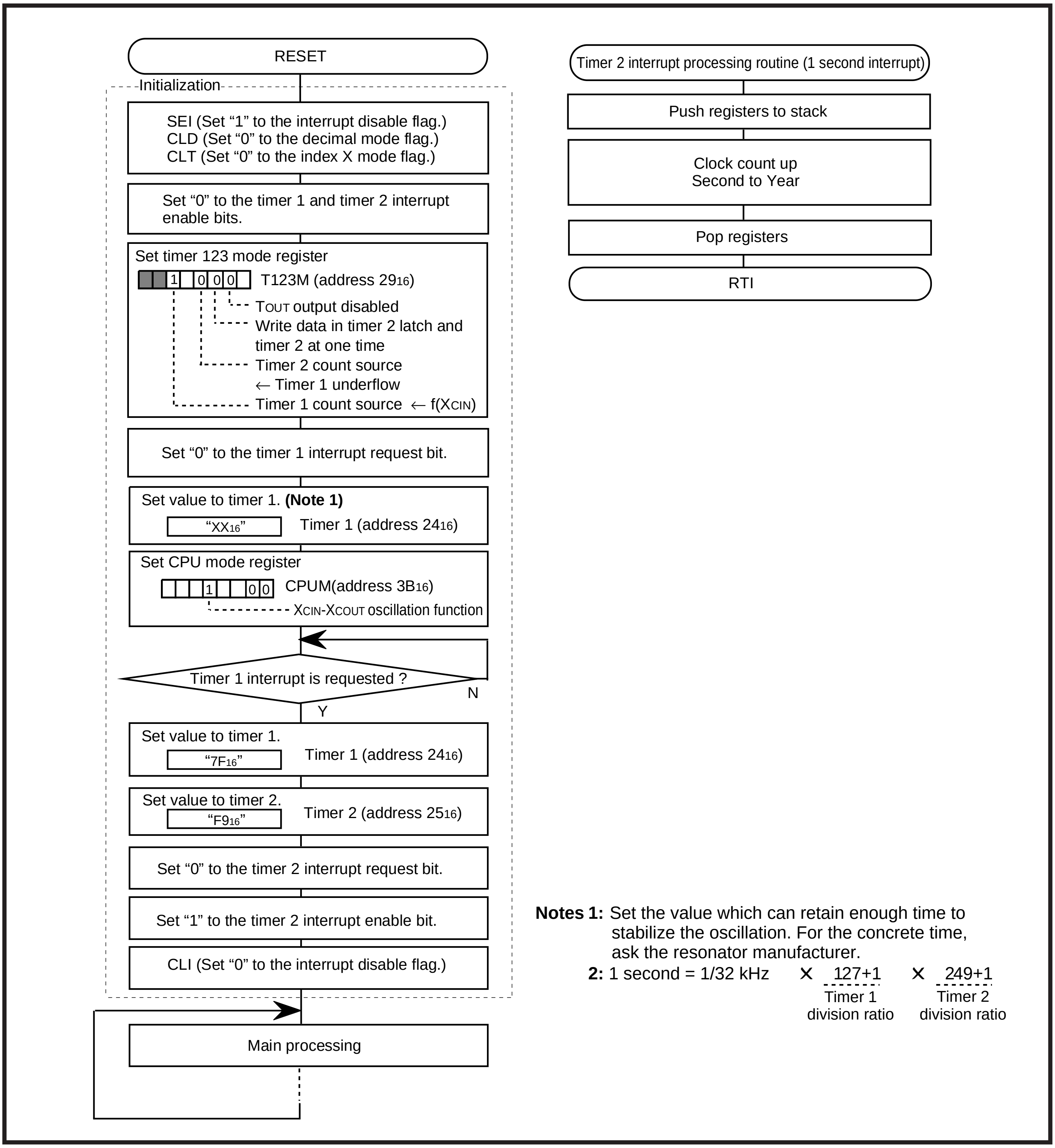
2.4.4 Application example of timers 1, 2, 3

■ **Outline**

The count clock is divided by the timers and the clock is counted up with the timer 2 interrupt at 1-second intervals.

■ **Specifications**

f(X_{CIN}) = 32 kHz is divided by timer 1 and timer 2 so that the timer 2 interrupt occurs at 1-second intervals. The clock is counted up in the timer 2 interrupt processing routine.


RESET
Initialization
SEI (Set "1" to the interrupt disable flag.)
CLD (Set "0" to the decimal mode flag.)
CLT (Set "0" to the index X mode flag.)
Set "0" to the timer 1 and timer 2 interrupt enable bits.
Set timer 123 mode register
1 0 0 0
T123M (address 29₁₆)
TOUT output disabled
Write data in timer 2 latch and timer 2 at one time
Timer 2 count source ← Timer 1 underflow
Timer 1 count source ← f(XCIN)
Set "0" to the timer 1 interrupt request bit.
Set value to timer 1. (Note 1)
"XX₁₆"
Timer 1 (address 24₁₆)
Set CPU mode register
1 0 0
CPUM(address 3B₁₆)
XCIN-XCOUT oscillation function
Timer 1 interrupt is requested ?
N
Y
Set value to timer 1.
"7F₁₆"
Timer 1 (address 24₁₆)
Set value to timer 2.
"F9₁₆"
Timer 2 (address 25₁₆)
Set "0" to the timer 2 interrupt request bit.
Set "1" to the timer 2 interrupt enable bit.
CLI (Set "0" to the interrupt disable flag.)
Main processing
Timer 2 interrupt processing routine (1 second interrupt)
Push registers to stack
Clock count up
Second to Year
Pop registers
RTI

Notes 1: Set the value which can retain enough time to stabilize the oscillation. For the concrete time, ask the resonator manufacturer.

2: 1 second = 1/32 kHz × $\underbrace{127+1}_{\text{Timer 1 division ratio}}$ × $\underbrace{249+1}_{\text{Timer 2 division ratio}}$

Fig. 2.4.5 Control procedure example of timers 1, 2, 3

2.4.5 Notes on use

(1) Notes on timer 1 to timer 3

- When a value n (“0” to “255”) is written to the timer latch, the division ratio is 1/(n+1).
- In case of the following, a short pulse occurs on counted input signals, so that the timer count value may change greatly. Accordingly, after setting their count sources, set values in order of timer 1, timer 2, and timer 3 following:
  - When the count sources of timers 1 to 3 are switched
  - When the timer 1 output signal is selected as a count source of timer 2 or timer 3 and data is written to timer 1

(2) Notes on timer 2

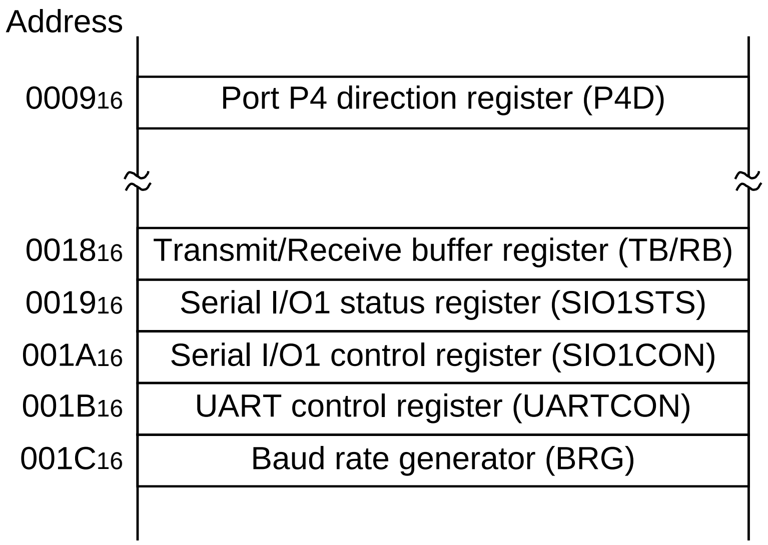
- When writing to the latch only is selected, the value written into timer 2 is written only in the latch for reloading and the contents of timer 2 is not changed. This written value is transferred to timer 2 at the first underflow after writing. Normally, a value is written in both the latch and timer at one time. That is, when a value is written to the timer, it is set in both the latch and the timer.
- In the T_{OUT} output enabled state, a signal whose polarity is reversed each time timer 2 underflows is output from the T_{OUT} pin. In this case, set port $P5_0$ (sharing with the T_{OUT} pin) to the output mode.

2.5 Serial I/O

The 3874 group has serial I/O1, serial I/O2, and serial I/O3. This section describes each serial I/O.

2.5.1 Serial I/O1 memory assignment

Serial I/O1 can be used as either clock synchronous or asynchronous (UART) serial I/O. This clause shows the relevant registers and the register structure; and the setting method and the application examples of clock synchronous (transmit and receive) and UART (transmit and receive) are described separately.
Figure 2.5.1 shows the memory assignment of the serial I/O1 relevant registers. Each of these registers is described below.



Address	
0009_{16}	Port P4 direction register (P4D)
≈	≈
0018_{16}	Transmit/Receive buffer register (TB/RB)
0019_{16}	Serial I/O1 status register (SIO1STS)
$001A_{16}$	Serial I/O1 control register (SIO1CON)
$001B_{16}$	UART control register (UARTCON)
$001C_{16}$	Baud rate generator (BRG)

Fig. 2.5.1 Memory assignment of serial I/O1 relevant registers

2.5.2 Serial I/O1 relevant registers

(1) Port P4 direction register

■ Selecting of port RxD/P4$_4$ direction mode

When performing the reception, set “0” to bit 4 to the input mode.

■ Selecting of port TxD/P4$_5$ direction mode

When performing the transmission, set “1” to bit 5 to the output mode.

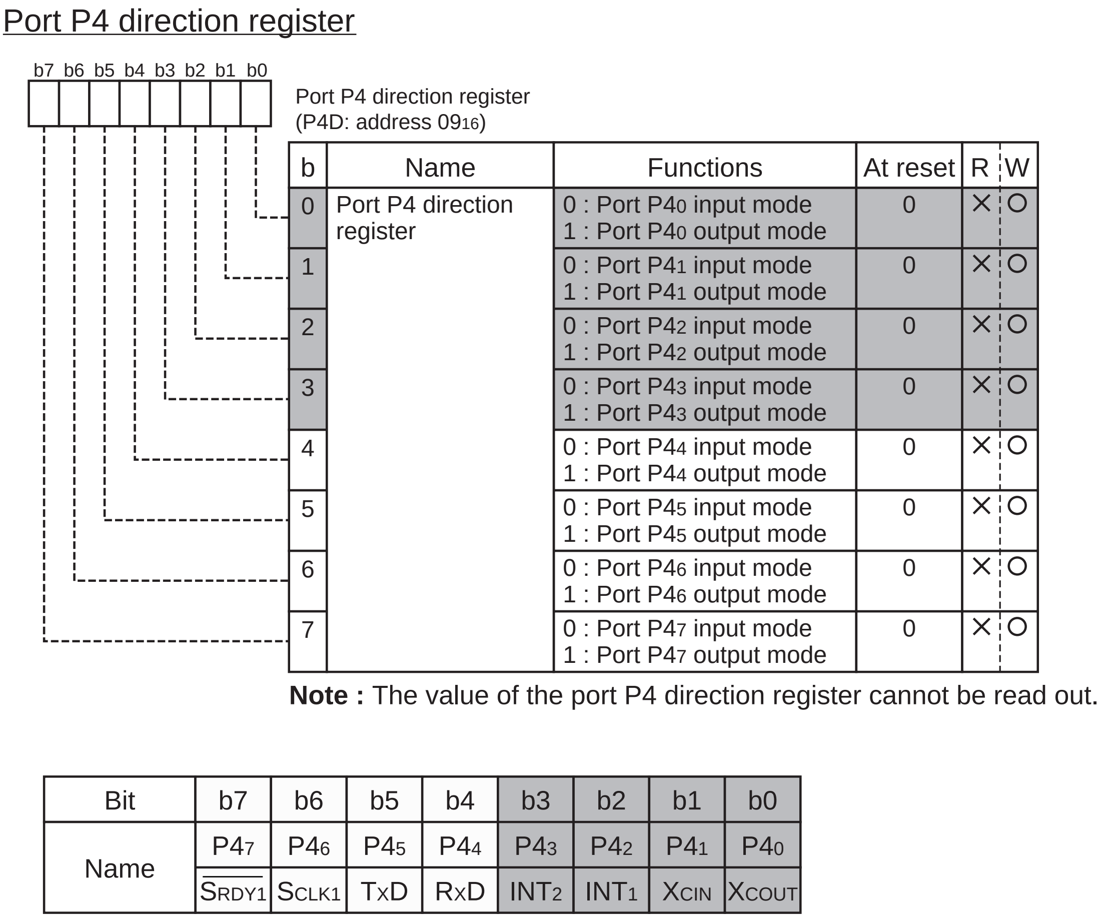
■ Selecting of port S$_{CLK1}$/P4$_6$ direction mode

When using an external clock as the synchronous clock, set “0” to bit 6 to the input mode.
When the synchronous clock is provided to the external, set “1” to bit 6 to the output mode.

■ Selecting of port $\overline{S_{RDY1}}$/P4$_7$ direction mode

When the function of $\overline{S_{RDY1}}$ which is a sharing pin is used, set “1” to bit 7 to the output mode.

The TxD pin and the S$_{CLK1}$ pin can select pull-up by the PULL UP register in the output mode.
Figure 2.5.2 shows the structure of the port P4 direction register.



b	Name	Functions	At reset	R	W
0	Port P4 direction register	0 : Port P4$_0$ input mode 1 : Port P4$_0$ output mode	0	×	○
1		0 : Port P4$_1$ input mode 1 : Port P4$_1$ output mode	0	×	○
2		0 : Port P4$_2$ input mode 1 : Port P4$_2$ output mode	0	×	○
3		0 : Port P4$_3$ input mode 1 : Port P4$_3$ output mode	0	×	○
4		0 : Port P4$_4$ input mode 1 : Port P4$_4$ output mode	0	×	○
5		0 : Port P4$_5$ input mode 1 : Port P4$_5$ output mode	0	×	○
6		0 : Port P4$_6$ input mode 1 : Port P4$_6$ output mode	0	×	○
7		0 : Port P4$_7$ input mode 1 : Port P4$_7$ output mode	0	×	○

Note : The value of the port P4 direction register cannot be read out.

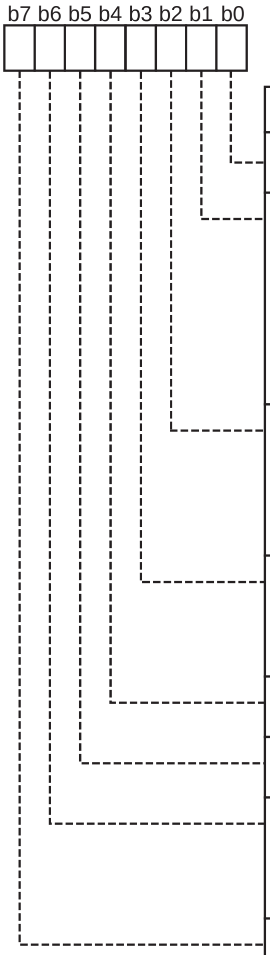
Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	P4$_7$	P4$_6$	P4$_5$	P4$_4$	P4$_3$	P4$_2$	P4$_1$	P4$_0$
	$\overline{S_{RDY1}}$	S$_{CLK1}$	TxD	RxD	INT$_2$	INT$_1$	X$_{CIN}$	X$_{COUT}$

Fig. 2.5.2 Structure of port P4 direction register

(2) Serial I/O1 control register

This register controls various functions relevant to the serial I/O1, such as transmit/receive modes, clocks, and pin functions, etc. All the bits of this register can be read and written by software. Figure 2.5.3 shows the structure of the serial I/O1 control register.

Serial I/O1 control register



Serial I/O1 control register
(SIO1CON: address $1A_{16}$)

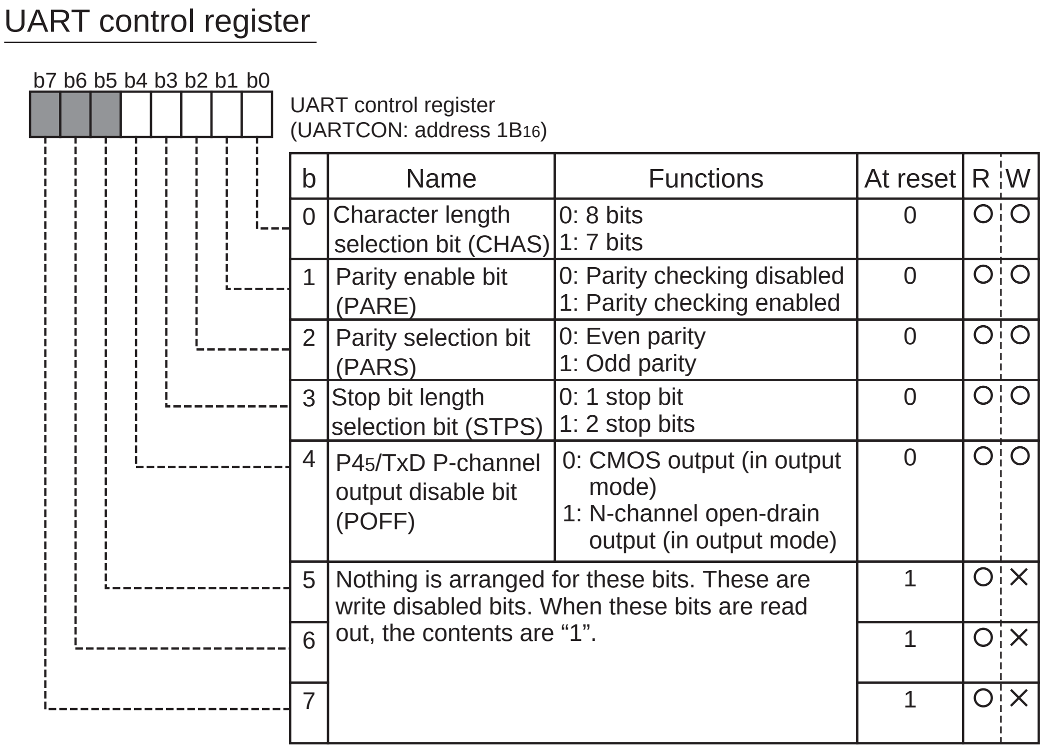
b	Name	Functions	At reset	R	W
0	BRG count source selection bit (CSS)	0: $f(X_{IN})$ 1: $f(X_{IN})/4$	0	○	○
1	Serial I/O1 synchronous clock selection bit (SCS)	•In clock synchronous mode 0: BRG output/4 1: External clock input •In UART mode 0: BRG output/16 1: External clock input/16	0	○	○
2	$\overline{\text{SRDY1}}$ output enable bit (SRDY)	0: $P4_7/\overline{\text{SRDY1}}$ pin operates as normal I/O port $P4_7$ 1: $P4_7/\overline{\text{SRDY1}}$ pin operates as signal output pin $\overline{\text{SRDY1}}$	0	○	○
3	Transmit interrupt source selection bit (TIC)	0: When transmit buffer has emptied 1: When transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0: Transmit disabled 1: Transmit enabled	0	○	○
5	Receive enable bit (RE)	0: Receive disabled 1: Receive enabled	0	○	○
6	Serial I/O1 mode selection bit (SIOM)	0: Clock asynchronous serial I/O (UART) mode 1: Clock synchronous serial I/O mode	0	○	○
7	Serial I/O1 enable bit (SIOE)	0: Serial I/O1 disabled (pins $P4_4$–$P4_7$ operate as normal I/O pins) 1: Serial I/O1 enabled (pins $P4_4$–$P4_7$ operate as serial I/O pins)	0	○	○

Fig. 2.5.3 Structure of serial I/O1 control register

(3) UART control register

This register controls the transfer data format in the UART mode and the output format of the P4$_5$/TxD pin.

Figure 2.5.4 shows the structure of the UART control register.



b	Name	Functions	At reset	R	W
0	Character length selection bit (CHAS)	0: 8 bits 1: 7 bits	0	○	○
1	Parity enable bit (PARE)	0: Parity checking disabled 1: Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0: Even parity 1: Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0: 1 stop bit 1: 2 stop bits	0	○	○
4	P4$_5$/TxD P-channel output disable bit (POFF)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “1”.		1	○	×
6			1	○	×
7			1	○	×

Fig. 2.5.4 Structure of UART control register

(4) Serial I/O1 status register

This register consists of the following:

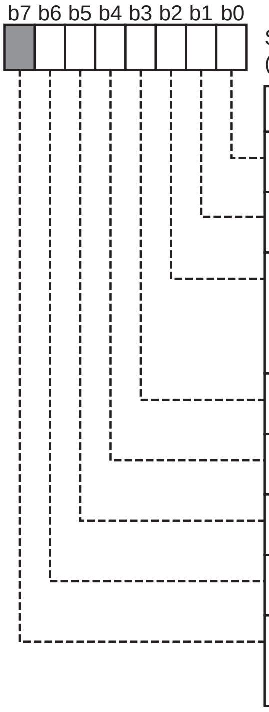
- flags indicating the states of the buffers and registers used for transmission/reception
- error flags

This is a read only register.

Bit 7 is unused and set to “1” at reading.

Figure 2.5.5 shows the structure of the serial I/O1 status register, and each bit is described below.

Serial I/O1 status register



Serial I/O1 status register (SIO1STS: address 19_{16})

b	Name	Functions	At reset	R	W
0	Transmit buffer empty flag (TBE)	0: Buffer full 1: Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0: Buffer empty 1: Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0: Transmit shift in progress 1: Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0: No error 1: Overrun error	0	○	×
4	Parity error flag (PE)	0: No error 1: Parity error	0	○	×
5	Framing error flag (FE)	0: No error 1: Framing error	0	○	×
6	Summing error flag (SE)	0: (OE) U (PE) U (FE) = 0 1: (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is arranged for this bit. This is write disabled bit. When this bit is read out, the contents is “1”.		1	○	×

Fig. 2.5.5 Structure of serial I/O1 status register

■ **Transmit buffer empty flag (bit 0)**

This bit is automatically cleared to “0” by writing transmit data into the transmit buffer register. After the transmit data is written in the transmit buffer register, it is transferred to the transmit shift register. When this transfer is completed and the transmit buffer register becomes empty, this flag is automatically is set to “1.”

It is possible to write transmit data into the transmit buffer register only while the transmit buffer empty flag is “1.”

This flag is valid in both the clock synchronous mode and the UART mode.

■ Receive buffer full flag (bit 1)

When all receive data has been input to the receive shift register and then this receive data is transferred to the receive buffer register, this flag is automatically set to “1.”
When the transferred receive data is read out from the receive buffer register, the flag is automatically cleared to “0.”
If all the next receive data is input to the receive shift register when the receive buffer full flag is “1” (the receive buffer register has not been read out), the overrun error flag* is set to “1.”
This flag is valid in both the clock synchronous mode and the UART mode.
✻: bit 3 of serial I/O1 status register

■ Transmit shift register shift completion flag (bit 2)

When a shift operation (transmission of the first data bit) starts with a synchronous clock after transmit data of the transmit shift register is transferred, this flag is cleared to “0.” When the shift operation is completed (completion of transmission of the last data bit), the flag is set to “1.”
This flag is valid in both the clock synchronous mode and the UART mode.

■ Overrun error flag (bit 3)

If all the next receive data is input to the receive shift register when the receive buffer register contains the data (not read out), this flag is set to “1” (occurrence of an overrun error). This flag is set to “0” by one of the following operations.
•Set “0” to the serial I/O1 enable bit*1
•Set “0” to the receive enable bit*2
•Write data (arbitrary) into the serial I/O1 status register
This flag is valid in both the clock synchronous mode and the UART mode.
✻1: bit 7 of serial I/O1 control register
✻2: bit 5 of serial I/O1 control register

■ Parity error flag (bit 4)

In the UART mode, this flag checks an even parity or odd parity by hardware.
When the parity of received data is different from the set parity, this flag is set to “1.”
This flag is set to “0” by one of the following operations.
•Set “0” to the serial I/O1 enable bit*1
•Set “0” to the receive enable bit*2
•Write data (arbitrary) into the serial I/O1 status register
This flag is valid only in the parity enable state in the UART mode.
✻1: bit 7 of serial I/O1 control register
✻2: bit 5 of serial I/O1 control register

■ Framing error flag (bit 5)

In the UART mode, this flag judges whether frame synchronization is abnormal.
When the stop bit of receive data cannot be received at the set timing, this flag is set to “1.”
This flag is set to “0” by one of the following operations.
•Set “0” to the serial I/O1 enable bit*1
•Set “0” to the receive enable bit*2
•Write data (arbitrary) into the serial I/O1 status register
This flag is valid only in the UART mode.
✻1: bit 7 of serial I/O1 control register
✻2: bit 5 of serial I/O1 control register

■ **Summing error flag (bit 6)**

This flag is set to “1” when any one of an overrun error, parity error, or framing error occurs. This flag is set to “0” by one of the following operations.

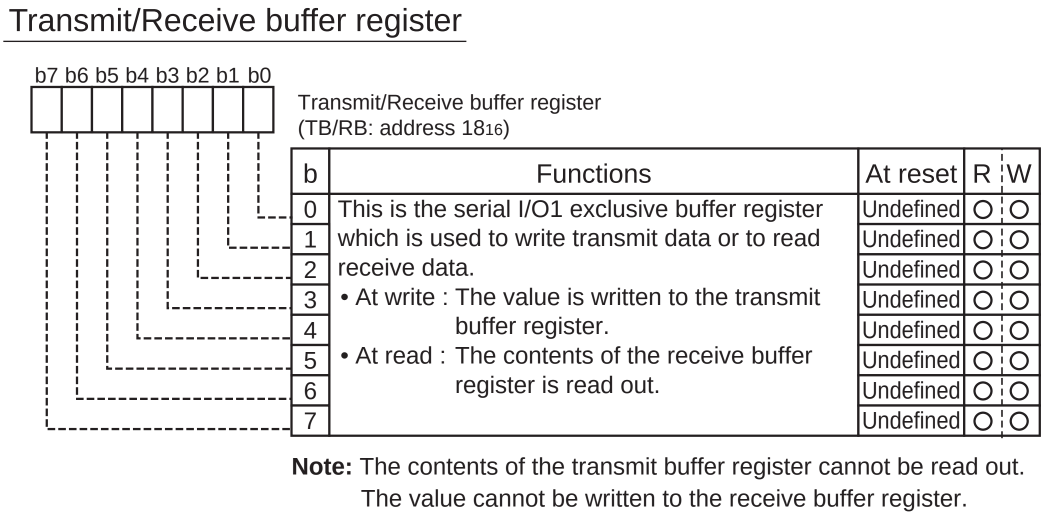
•Set “0” to the serial I/O1 enable bit*1
•Set “0” to the receive enable bit*2
•Write data (arbitrary) into the serial I/O1 status register

This flag is valid in both the clock synchronous mode and the UART mode.

∗1: bit 7 of serial I/O1 control register
∗2: bit 5 of serial I/O1 control register

(5) Transmit/Receive buffer register

This register is used to write transmit data or to read receive data when serial I/O1 is used. For data transmission, transmit data is written into this register. Received data is obtained by reading out this register. This resister is used for both the clock synchronous mode and the UART mode. Figure 2.5.6 shows the structure of the transmit/receive buffer register.



Transmit/Receive buffer register

Transmit/Receive buffer register
(TB/RB: address 18_{16})

b	Functions	At reset	R	W
0	This is the serial I/O1 exclusive buffer register which is used to write transmit data or to read receive data. • At write : The value is written to the transmit buffer register. • At read : The contents of the receive buffer register is read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

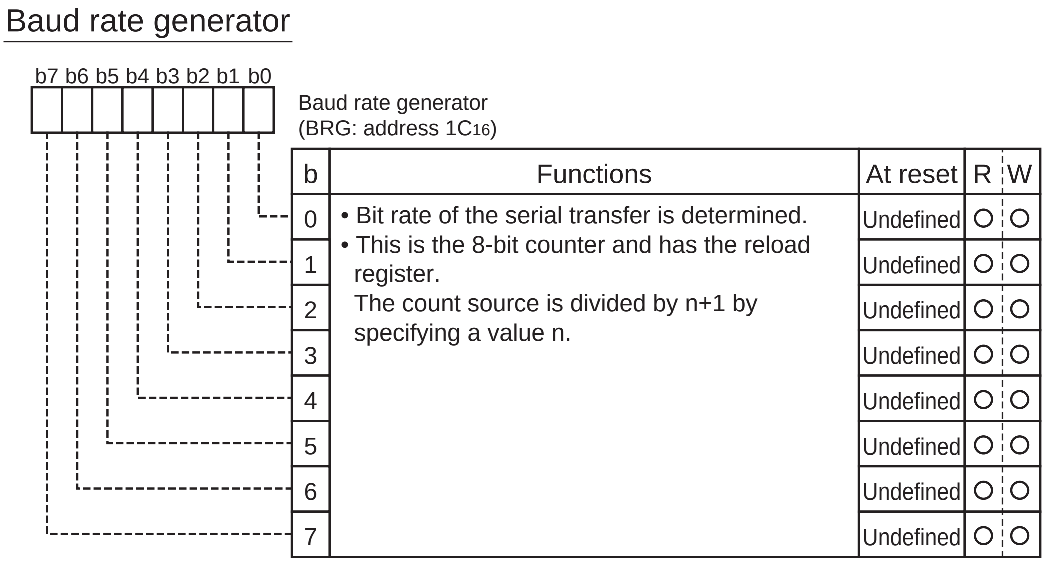
Note: The contents of the transmit buffer register cannot be read out.
The value cannot be written to the receive buffer register.

Fig. 2.5.6 Structure of transmit/receive buffer register

(6) Baud rate generator

The baud rate generator is the 8-bit counter. When an internal clock (BRG output/4 or BRG output/16) is used, this determines the data transfer rate (baud rate), which is a synchronous clock frequency. This register is used in both the clock synchronous mode and the UART mode.

Figure 2.5.7 shows the structure of the baud rate generator.



b	Functions	At reset	R	W
0	• Bit rate of the serial transfer is determined. • This is the 8-bit counter and has the reload register. The count source is divided by n+1 by specifying a value n.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 2.5.7 Structure of baud rate generator

■ **Data transfer rate (baud rate)**

The expression for baud rate, which is the frequency of a synchronous clock, is shown below.

When selecting an internal clock (using baud rate generator)

$$\underset{\text{[bps]}}{\text{Baud rate}} = \frac{f(X_{IN})}{\text{Division ratio}^{*1} \times (\text{BRG setting value}^{*2} + 1) \times X^{*3}}$$

Division ratio^{*1}: Select “1” or “4”
(Set by bit 0 of serial I/O1 control register)
BRG setting value^{*2}: Set 0 to 255 (00_{16} to FF_{16})
X^{*3}: “4” in clock synchronous mode, “16” in UART mode (set by bit 1 of serial I/O1 control register)

When selecting an external clock

Baud rate = Input clock to S_{CLK1} pin / X*
[bps]
X*: “1” in clock synchronous mode, “16” in UART mode

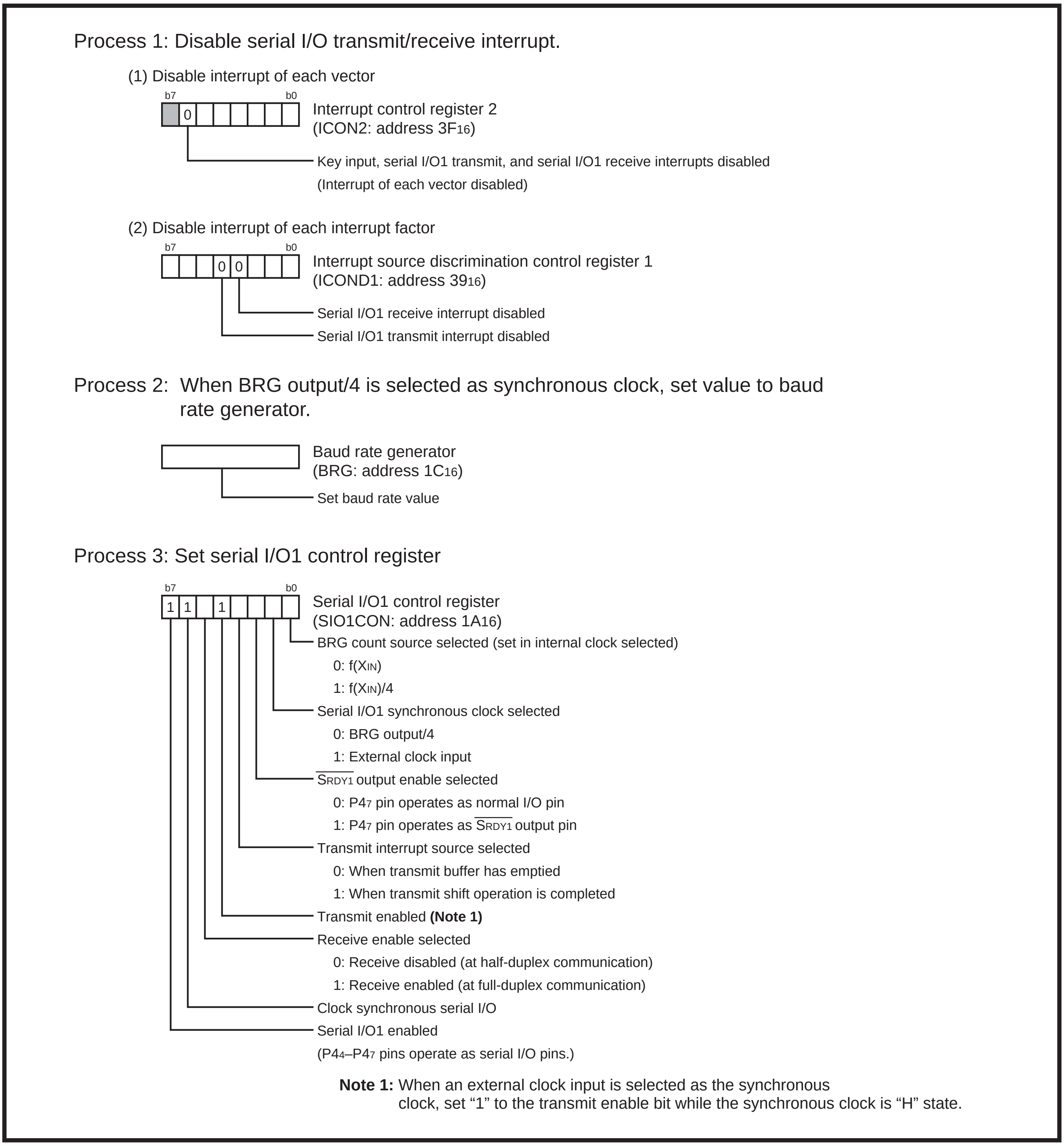
2.5.3 Clock synchronous serial I/O of serial I/O1

In the clock synchronous serial I/O mode, 8 shift clocks generated by the clock control circuit are used as a synchronous clock for transfer. Synchronizing with these synchronous clocks, the transmit operation on the transmitter and the receive operation on the receiver are simultaneously executed.

(1) Clock synchronous serial I/O of serial I/O1 setting method

■ Setting method at transmit

Figures 2.5.8 and 2.5.9 show the clock synchronous serial I/O of serial I/O1 setting method (at transmit).



Process 1: Disable serial I/O transmit/receive interrupt.

(1) Disable interrupt of each vector

Interrupt control register 2 (ICON2: address 3F$_{16}$)

Key input, serial I/O1 transmit, and serial I/O1 receive interrupts disabled (Interrupt of each vector disabled)

(2) Disable interrupt of each interrupt factor

Interrupt source discrimination control register 1 (ICOND1: address 39$_{16}$)

Serial I/O1 receive interrupt disabled

Serial I/O1 transmit interrupt disabled

Process 2: When BRG output/4 is selected as synchronous clock, set value to baud rate generator.

Baud rate generator (BRG: address 1C$_{16}$)

Set baud rate value

Process 3: Set serial I/O1 control register

Serial I/O1 control register (SIO1CON: address 1A$_{16}$)

BRG count source selected (set in internal clock selected)
0: f(X_{IN})
1: f(X_{IN})/4

Serial I/O1 synchronous clock selected
0: BRG output/4
1: External clock input

$\overline{S_{RDY1}}$ output enable selected
0: P4$_7$ pin operates as normal I/O pin
1: P4$_7$ pin operates as $\overline{S_{RDY1}}$ output pin

Transmit interrupt source selected
0: When transmit buffer has emptied
1: When transmit shift operation is completed

Transmit enabled **(Note 1)**

Receive enable selected
0: Receive disabled (at half-duplex communication)
1: Receive enabled (at full-duplex communication)

Clock synchronous serial I/O

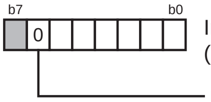
Serial I/O1 enabled (P4$_4$–P4$_7$ pins operate as serial I/O pins.)

Note 1: When an external clock input is selected as the synchronous clock, set “1” to the transmit enable bit while the synchronous clock is “H” state.

Fig. 2.5.8 Clock synchronous serial I/O of serial I/O1 setting method (at transmit) (1)

Process 4: When using the serial I/O1 transmit/receive interrupt, set “0” to the interrupt request bit.

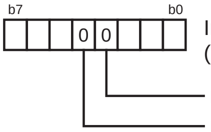
(1) Set “0” to the interrupt request bit of each vector.



Interrupt request register 2
(IREQ2: address $3D_{16}$)

No key input, serial I/O1 receive, serial I/O1 transmit interrupts request issued

(2) Set “0” to the interrupt request bit of each interrupt factor.



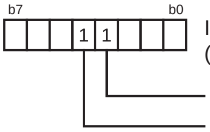
Interrupt source discrimination register 1
(IREQD1: address 38_{16})

No serial I/O1 receive interrupt request issued (in full-duplex communication)

No serial I/O1 transmit interrupt request issued

Process 5: When using the serial I/O1 transmit/receive interrupt, enable the serial I/O1 interrupt.

(1) Enable interrupt of each interrupt factor

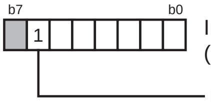


Interrupt source discrimination control register 1
(ICOND1: address 39_{16})

Serial I/O1 receive interrupt enabled (in full-duplex communication)

Serial I/O1 transmit interrupt enabled

(2) Enable interrupt of each vector **(Note 2)**

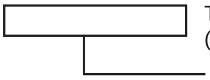


Interrupt control register 2
(ICON2: address $3F_{16}$)

Key input, serial I/O1 receive, serial I/O1 transmit interrupt enabled

Note 2: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to “1”, the interrupt may occur at this time.

Process 6: Transmit of serial data **(Note 3) (Note 4)**



Transmit/Receive buffer register
(TB/RB: address 18_{16})

Set transmit data

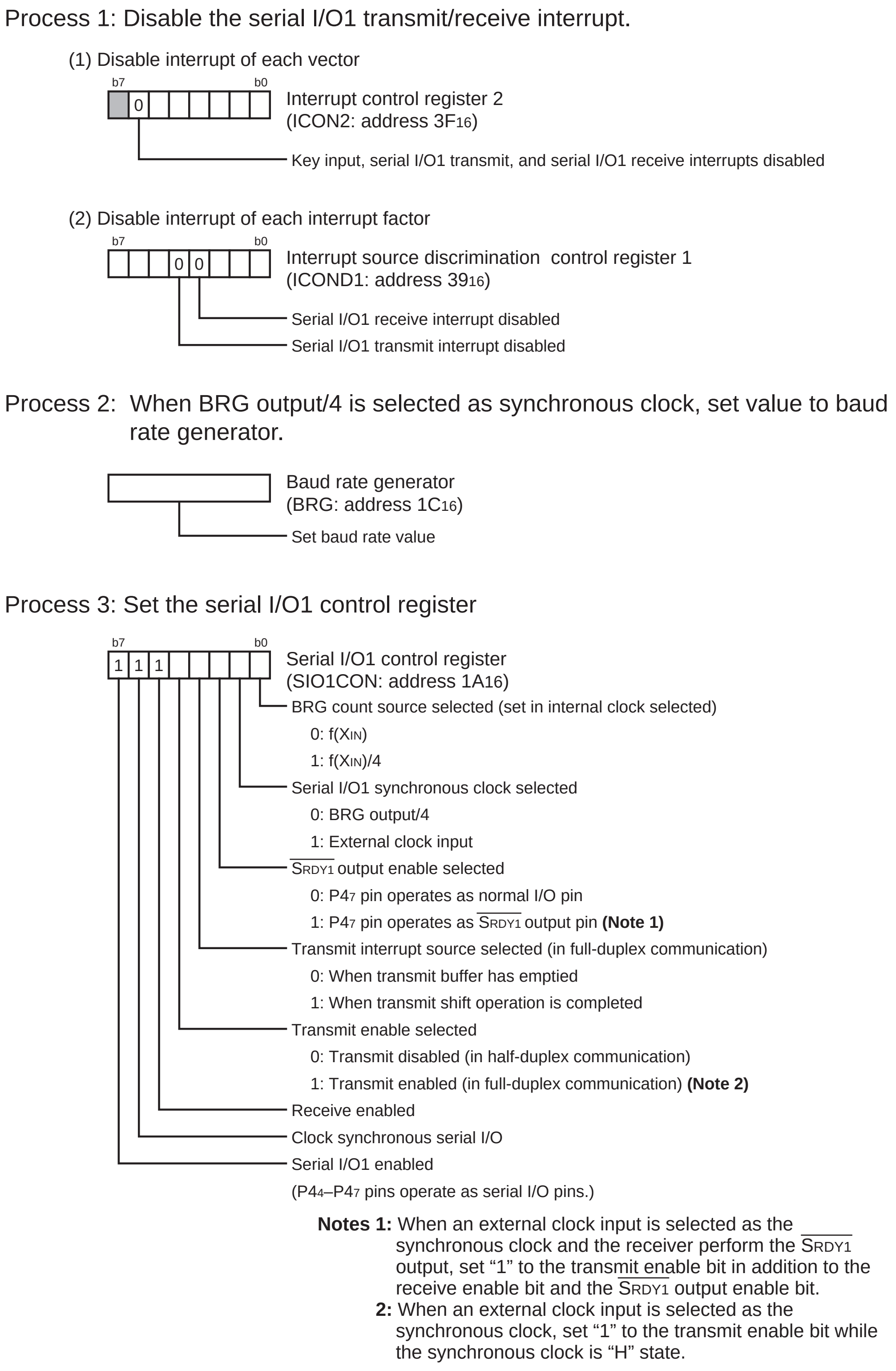
Notes 3: When an external clock input is selected as the synchronous clock, set the transmit data while the synchronous clock is “H” state.

4: When inputting the $\overline{S_{RDY1}}$ signal, set used pins to the input mode before transmitting data.

Fig. 2.5.9 Clock synchronous serial I/O of serial I/O1 setting method (at transmit) (2)

■ Setting method at receive

Figures 2.5.10 and 2.5.11 show the clock synchronous serial I/O of serial I/O1 setting method (at receive).



Process 1: Disable the serial I/O1 transmit/receive interrupt.

(1) Disable interrupt of each vector

Interrupt control register 2
(ICON2: address $3F_{16}$)

Key input, serial I/O1 transmit, and serial I/O1 receive interrupts disabled

(2) Disable interrupt of each interrupt factor

Interrupt source discrimination control register 1
(ICOND1: address 39_{16})

Serial I/O1 receive interrupt disabled

Serial I/O1 transmit interrupt disabled

Process 2: When BRG output/4 is selected as synchronous clock, set value to baud rate generator.

Baud rate generator
(BRG: address $1C_{16}$)

Set baud rate value

Process 3: Set the serial I/O1 control register

Serial I/O1 control register
(SIO1CON: address $1A_{16}$)

BRG count source selected (set in internal clock selected)
- 0: f(X_{IN})
- 1: f(X_{IN})/4

Serial I/O1 synchronous clock selected
- 0: BRG output/4
- 1: External clock input

$\overline{S_{RDY1}}$ output enable selected
- 0: $P4_7$ pin operates as normal I/O pin
- 1: $P4_7$ pin operates as $\overline{S_{RDY1}}$ output pin **(Note 1)**

Transmit interrupt source selected (in full-duplex communication)
- 0: When transmit buffer has emptied
- 1: When transmit shift operation is completed

Transmit enable selected
- 0: Transmit disabled (in half-duplex communication)
- 1: Transmit enabled (in full-duplex communication) **(Note 2)**

Receive enabled

Clock synchronous serial I/O

Serial I/O1 enabled
($P4_4$–$P4_7$ pins operate as serial I/O pins.)

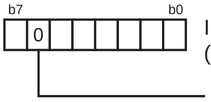
Notes 1: When an external clock input is selected as the synchronous clock and the receiver perform the $\overline{S_{RDY1}}$ output, set “1” to the transmit enable bit in addition to the receive enable bit and the $\overline{S_{RDY1}}$ output enable bit.

2: When an external clock input is selected as the synchronous clock, set “1” to the transmit enable bit while the synchronous clock is “H” state.

Fig. 2.5.10 Clock synchronous serial I/O of serial I/O1 setting method (at receive) (1)

Process 4: When using the serial I/O1 transmit/receive interrupt, set “0” to the interrupt request bit.

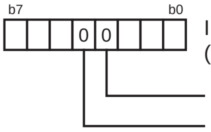
(1) Set “0” to the interrupt request bit of each vector.



Interrupt request register 2
(IREQ2: address $3D_{16}$)

No key input, serial I/O1 receive, serial I/O1 transmit interrupts request issued

(2) Set “0” to the interrupt request bit of each interrupt factor.



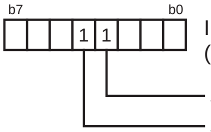
Interrupt source discrimination register 1
(IREQD1: address 38_{16})

No serial I/O1 receive interrupt request issued

No serial I/O1 transmit interrupt request issued (in full-duplex communication)

Process 5: When using the serial I/O1 transmit/receive interrupt, enable the serial I/O1 interrupt.

(1) Enable interrupt of each interrupt factor

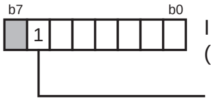


Interrupt source discrimination control register 1
(ICOND1: address 39_{16})

Serial I/O1 receive interrupt enabled

Serial I/O1 transmit interrupt enabled (in full-duplex communication)

(2) Enable interrupt of each vector **(Note 3)**

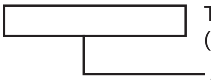


Interrupt control register 2
(ICON2: address $3F_{16}$)

Key input, serial I/O1 receive, serial I/O1 transmit interrupt enabled

Note 3: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to “1”, the interrupt may occur at this time.

Process 6: Receive of serial data **(Note 4) (Note 5)**



Transmit/Receive buffer register
(TB/RB: address 18_{16})

Set transmit data (in full-duplex communication)
Set dummy data (in half-duplex communication)

Notes 4: When an external clock input is selected as the synchronous clock, set the transmit data while the synchronous clock is “H” state.

5: When inputting the $\overline{S_{RDY1}}$ signal, set used pins to the input mode before transmitting data.

Fig. 2.5.11 Clock synchronous serial I/O of serial I/O1 setting method (at receive) (2)

(2) Clock synchronous serial I/O of serial I/O1 application example

■ **Outline**

The clock synchronous communication is performed between the 3874 groups.

■ **Specifications**

3874 group ① (transmitter in half-duplex communication)
•Synchronous clock: BRG output/4
•Port $P4_7$ is used as an $\overline{S_{RDY1}}$ signal input pin.
3874 group ② (receiver in half-duplex communication)
•Synchronous clock: External clock
•$\overline{S_{RDY1}}$ signal output

Figure 2.5.12 shows an example of connections and Figure 2.5.13 shows an example of control procedure.

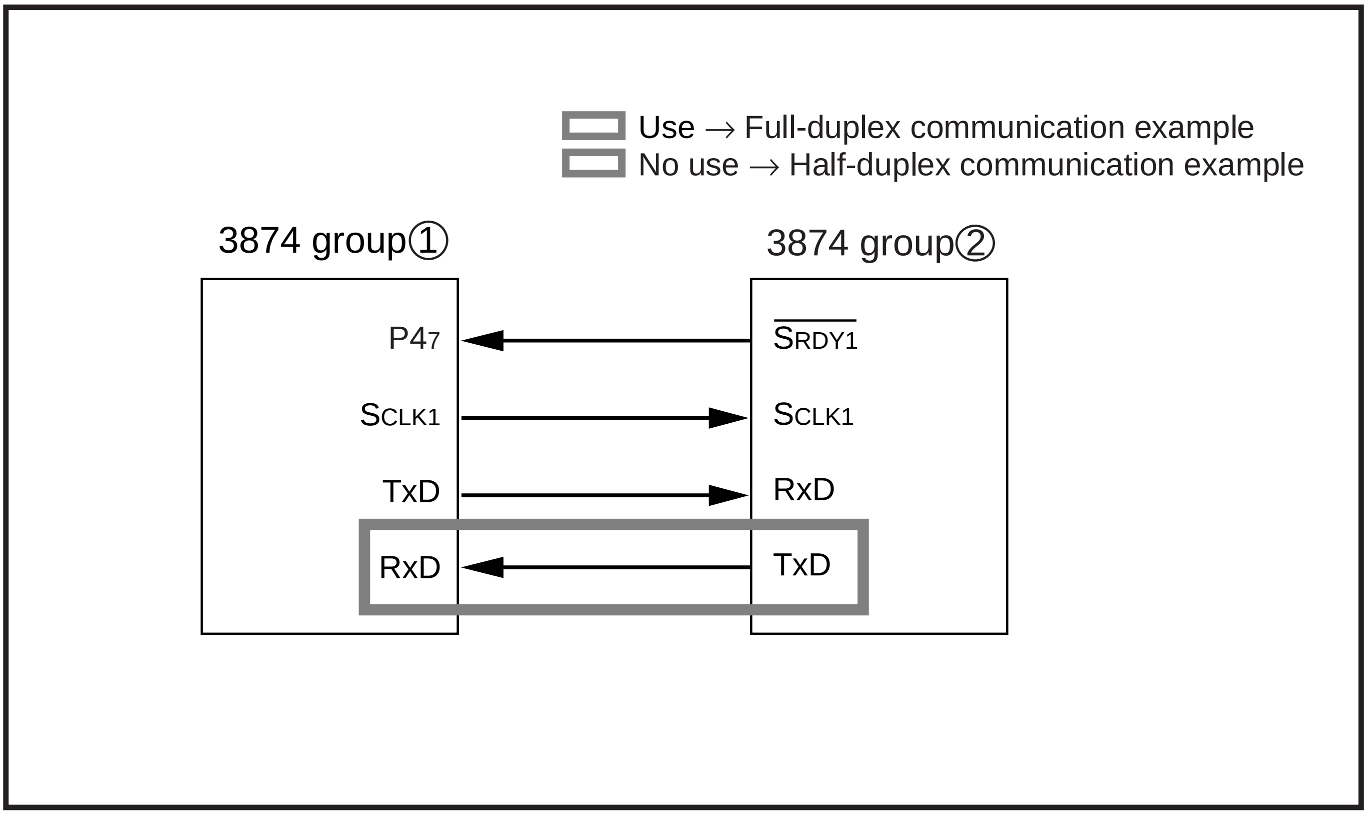
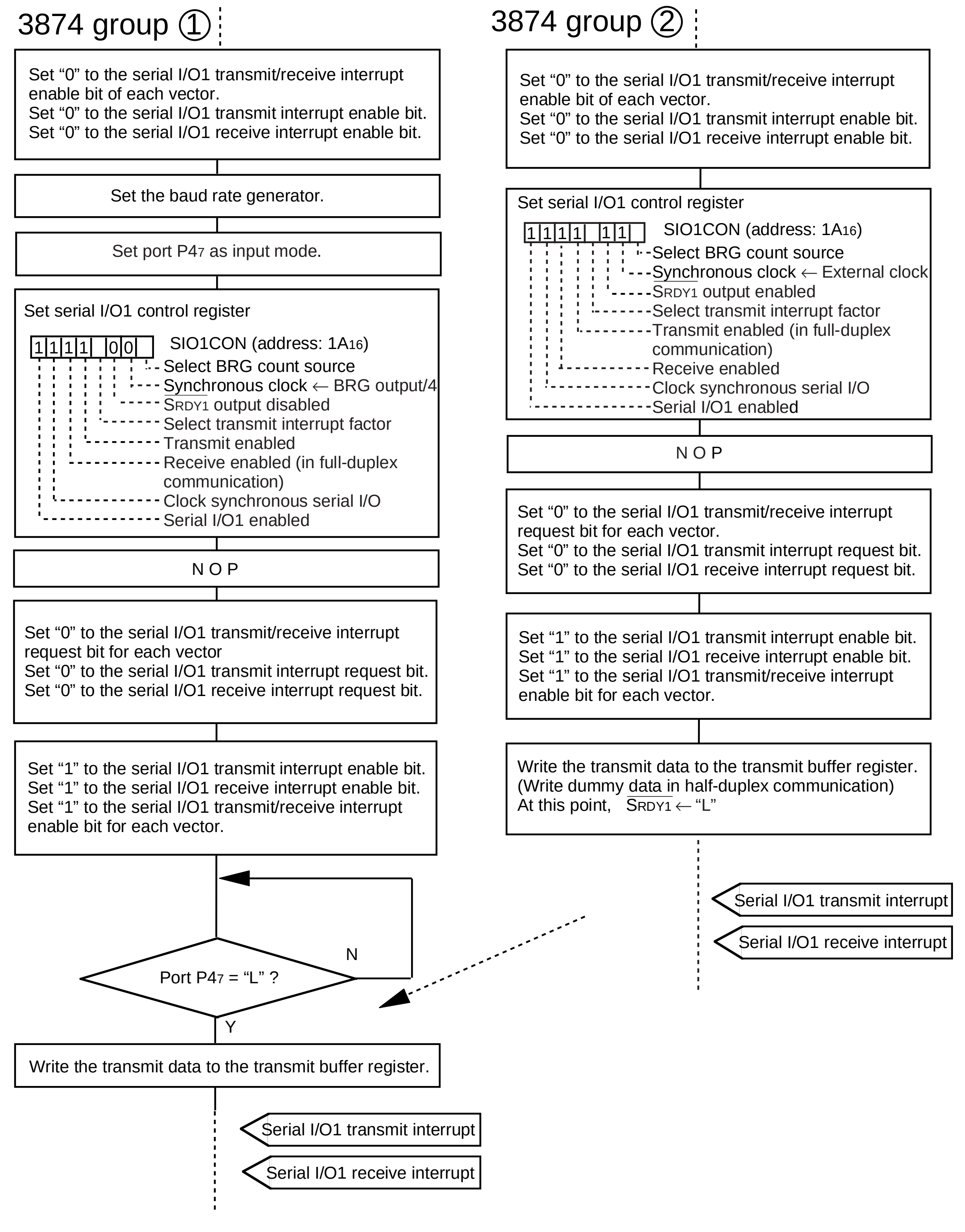

Use → Full-duplex communication example
No use → Half-duplex communication example
3874 group①
3874 group②
P47
SCLK1
TxD
RxD
SRDY1
SCLK1
RxD
TxD

Fig. 2.5.12 Example of connections (clock synchronous serial I/O mode of serial I/O1)



3874 group ①

Set “0” to the serial I/O1 transmit/receive interrupt enable bit of each vector.
Set “0” to the serial I/O1 transmit interrupt enable bit.
Set “0” to the serial I/O1 receive interrupt enable bit.

Set the baud rate generator.

Set port P4₇ as input mode.

Set serial I/O1 control register

1 1 1 1 _ 0 0 _ SIO1CON (address: 1A16)
- Select BRG count source
- Synchronous clock ← BRG output/4
- $\overline{\text{SRDY1}}$ output disabled
- Select transmit interrupt factor
- Transmit enabled
- Receive enabled (in full-duplex communication)
- Clock synchronous serial I/O
- Serial I/O1 enabled

N O P

Set “0” to the serial I/O1 transmit/receive interrupt request bit for each vector
Set “0” to the serial I/O1 transmit interrupt request bit.
Set “0” to the serial I/O1 receive interrupt request bit.

Set “1” to the serial I/O1 transmit interrupt enable bit.
Set “1” to the serial I/O1 receive interrupt enable bit.
Set “1” to the serial I/O1 transmit/receive interrupt enable bit for each vector.

Port P4₇ = “L” ?

Write the transmit data to the transmit buffer register.

Serial I/O1 transmit interrupt

Serial I/O1 receive interrupt

3874 group ②

Set “0” to the serial I/O1 transmit/receive interrupt enable bit of each vector.
Set “0” to the serial I/O1 transmit interrupt enable bit.
Set “0” to the serial I/O1 receive interrupt enable bit.

Set serial I/O1 control register

1 1 1 1 _ 1 1 _ SIO1CON (address: 1A16)
- Select BRG count source
- Synchronous clock ← External clock
- $\overline{\text{SRDY1}}$ output enabled
- Select transmit interrupt factor
- Transmit enabled (in full-duplex communication)
- Receive enabled
- Clock synchronous serial I/O
- Serial I/O1 enabled

N O P

Set “0” to the serial I/O1 transmit/receive interrupt request bit for each vector.
Set “0” to the serial I/O1 transmit interrupt request bit.
Set “0” to the serial I/O1 receive interrupt request bit.

Set “1” to the serial I/O1 transmit interrupt enable bit.
Set “1” to the serial I/O1 receive interrupt enable bit.
Set “1” to the serial I/O1 transmit/receive interrupt enable bit for each vector.

Write the transmit data to the transmit buffer register.
(Write dummy data in half-duplex communication)
At this point, $\overline{\text{SRDY1}}$ ← “L”

Serial I/O1 transmit interrupt

Serial I/O1 receive interrupt

Fig. 2.5.13 Example of control procedure

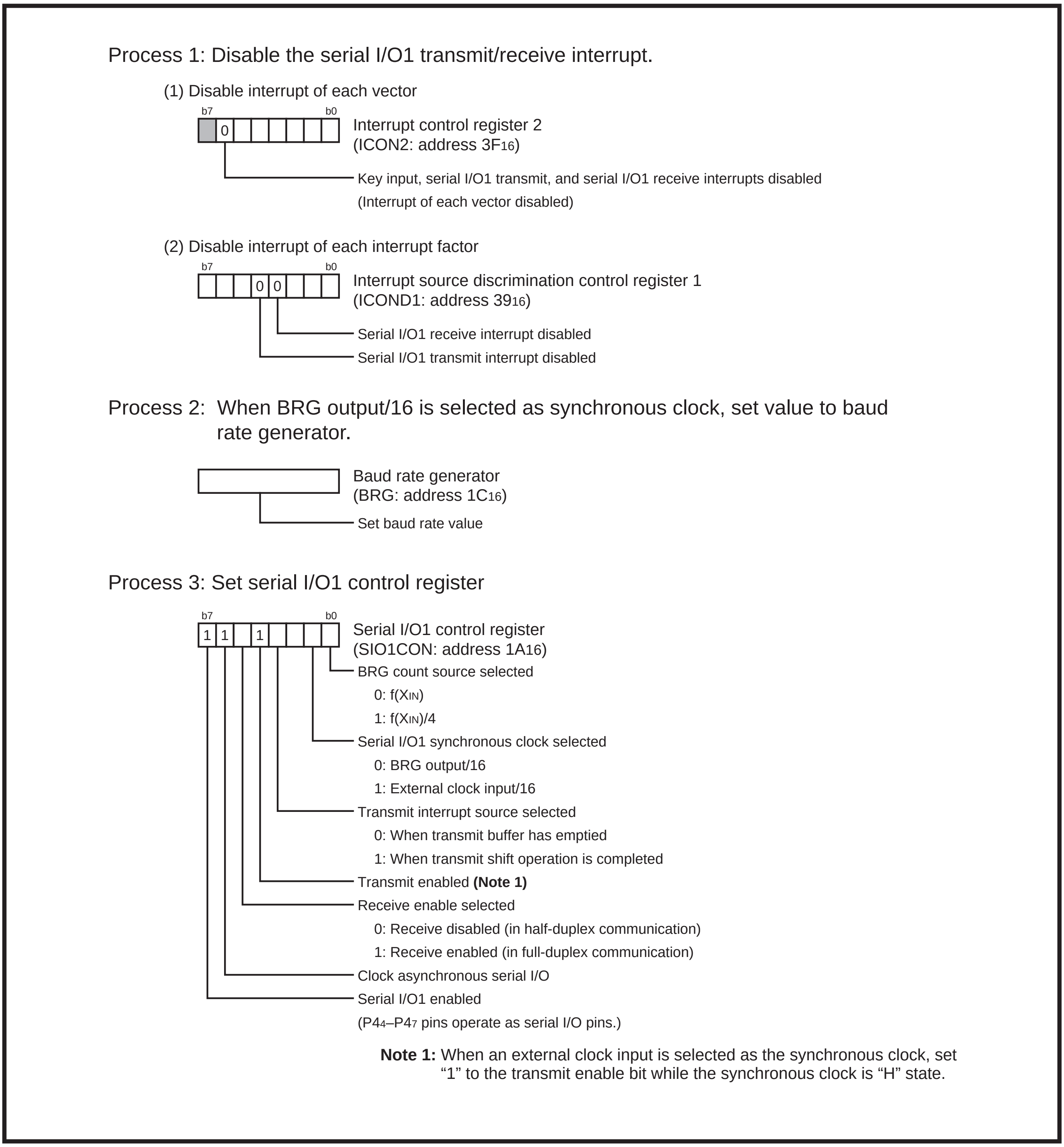
2.5.4 Clock asynchronous serial I/O (UART) of serial I/O1

In the clock asynchronous serial I/O (UART) mode, the baud rate and the transfer data formats used by a transmitter and a receiver must be identical. The data transmit/receive is performed asynchronously.

(1) UART of serial I/O1 setting method

■ Setting method at transmit

Figures 2.5.14 and 2.5.15 show the UART of serial I/O1 setting method (at transmit).



Process 1: Disable the serial I/O1 transmit/receive interrupt.

(1) Disable interrupt of each vector

Interrupt control register 2
(ICON2: address $3F_{16}$)

Key input, serial I/O1 transmit, and serial I/O1 receive interrupts disabled
(Interrupt of each vector disabled)

(2) Disable interrupt of each interrupt factor

Interrupt source discrimination control register 1
(ICOND1: address 39_{16})

Serial I/O1 receive interrupt disabled

Serial I/O1 transmit interrupt disabled

Process 2: When BRG output/16 is selected as synchronous clock, set value to baud rate generator.

Baud rate generator
(BRG: address $1C_{16}$)

Set baud rate value

Process 3: Set serial I/O1 control register

Serial I/O1 control register
(SIO1CON: address $1A_{16}$)

BRG count source selected
0: f(X_{IN})
1: f(X_{IN})/4

Serial I/O1 synchronous clock selected
0: BRG output/16
1: External clock input/16

Transmit interrupt source selected
0: When transmit buffer has emptied
1: When transmit shift operation is completed

Transmit enabled **(Note 1)**

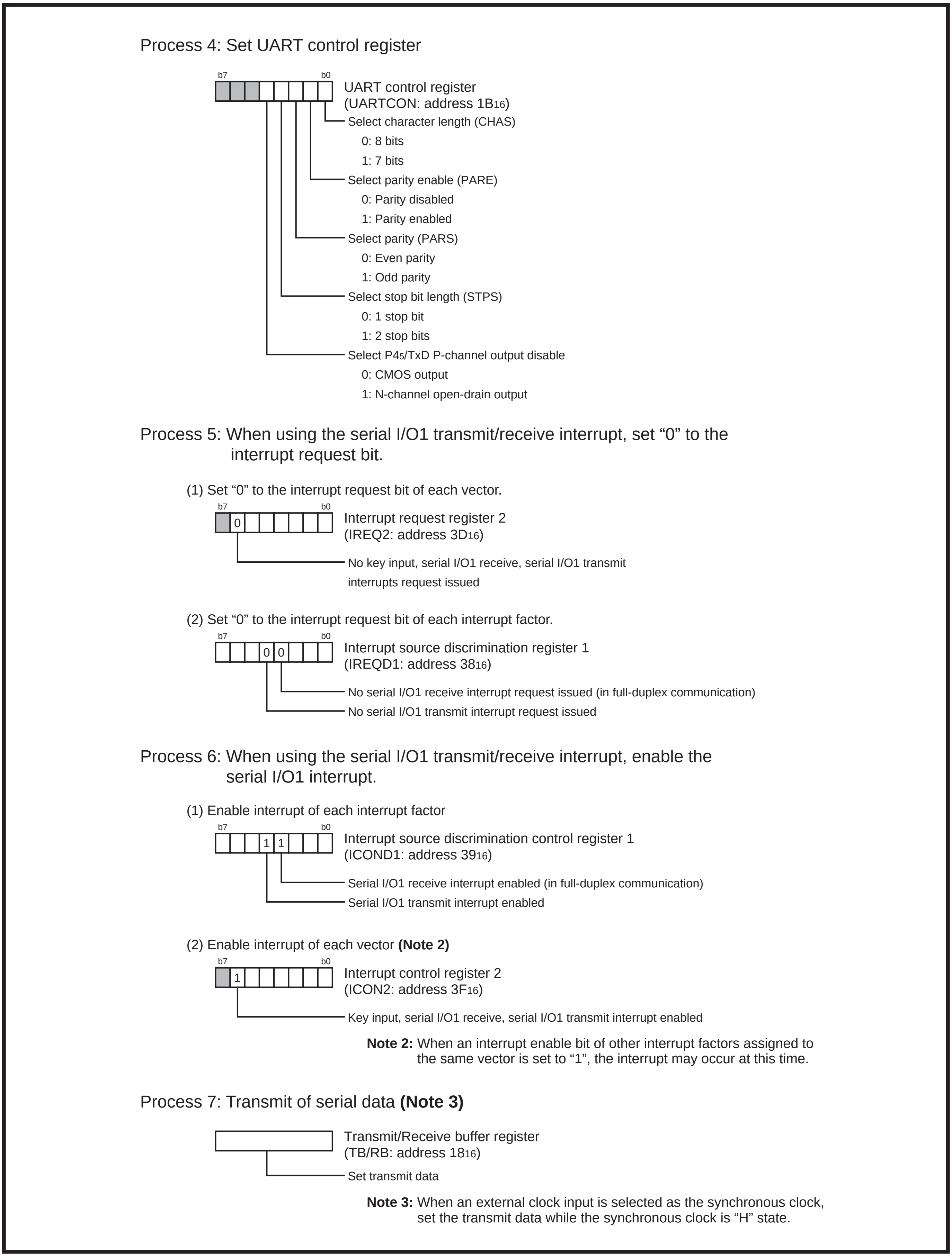
Receive enable selected
0: Receive disabled (in half-duplex communication)
1: Receive enabled (in full-duplex communication)

Clock asynchronous serial I/O

Serial I/O1 enabled
($P4_4$–$P4_7$ pins operate as serial I/O pins.)

Note 1: When an external clock input is selected as the synchronous clock, set "1" to the transmit enable bit while the synchronous clock is "H" state.

Fig. 2.5.14 UART of serial I/O1 setting method (at transmit) (1)



Process 4: Set UART control register

b7
b0
UART control register
(UARTCON: address 1B16)
Select character length (CHAS)
0: 8 bits
1: 7 bits
Select parity enable (PARE)
0: Parity disabled
1: Parity enabled
Select parity (PARS)
0: Even parity
1: Odd parity
Select stop bit length (STPS)
0: 1 stop bit
1: 2 stop bits
Select P45/TxD P-channel output disable
0: CMOS output
1: N-channel open-drain output

Process 5: When using the serial I/O1 transmit/receive interrupt, set “0” to the interrupt request bit.

(1) Set “0” to the interrupt request bit of each vector.

b7
b0
0
Interrupt request register 2
(IREQ2: address 3D16)
No key input, serial I/O1 receive, serial I/O1 transmit interrupts request issued

(2) Set “0” to the interrupt request bit of each interrupt factor.

b7
b0
0 0
Interrupt source discrimination register 1
(IREQD1: address 3816)
No serial I/O1 receive interrupt request issued (in full-duplex communication)
No serial I/O1 transmit interrupt request issued

Process 6: When using the serial I/O1 transmit/receive interrupt, enable the serial I/O1 interrupt.

(1) Enable interrupt of each interrupt factor

b7
b0
1 1
Interrupt source discrimination control register 1
(ICOND1: address 3916)
Serial I/O1 receive interrupt enabled (in full-duplex communication)
Serial I/O1 transmit interrupt enabled

(2) Enable interrupt of each vector **(Note 2)**

b7
b0
1
Interrupt control register 2
(ICON2: address 3F16)
Key input, serial I/O1 receive, serial I/O1 transmit interrupt enabled

Note 2: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to “1”, the interrupt may occur at this time.

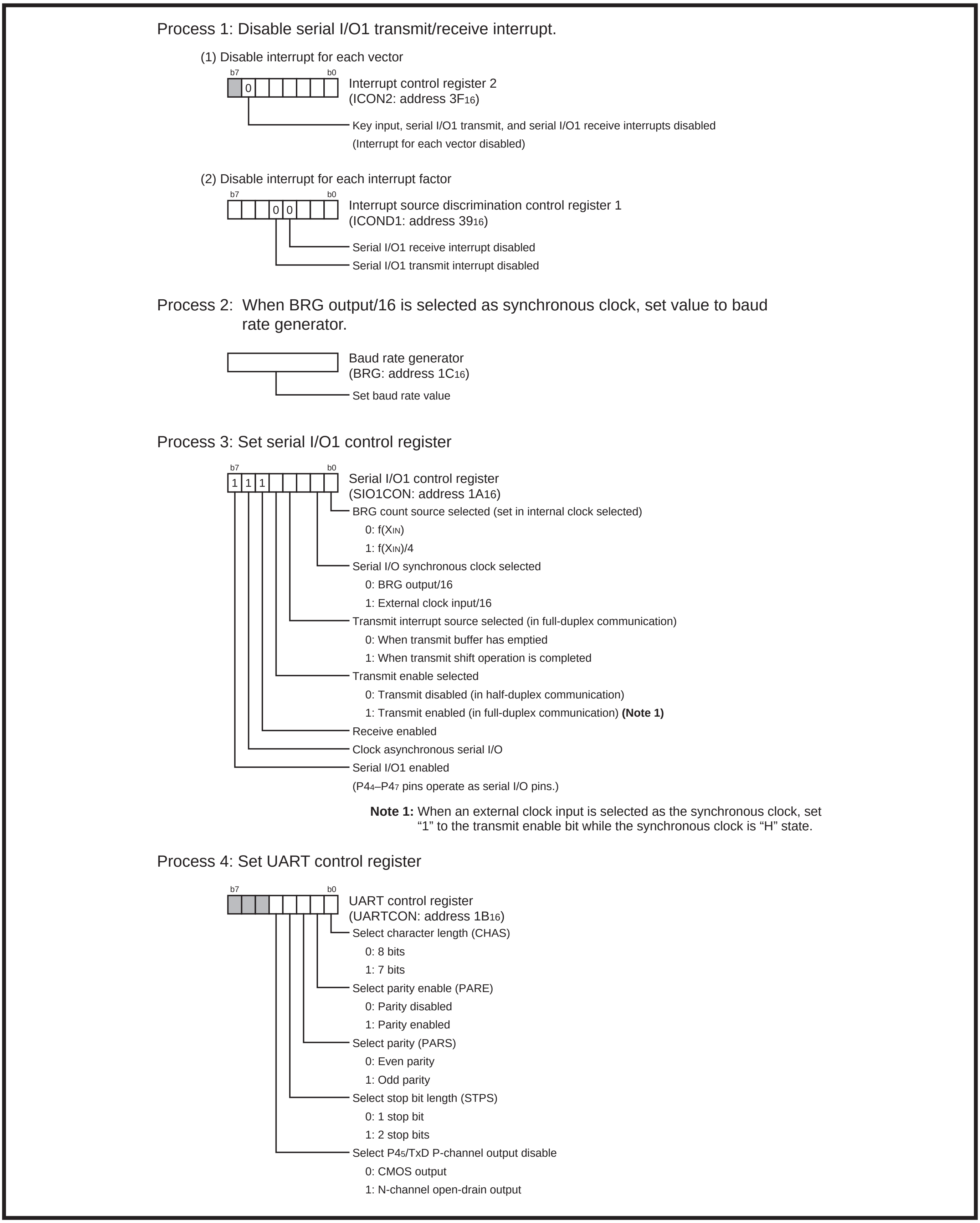
Process 7: Transmit of serial data **(Note 3)**

Note 3: When an external clock input is selected as the synchronous clock, set the transmit data while the synchronous clock is “H” state.

Fig. 2.5.15 UART of serial I/O1 setting method (at transmit) (2)

■ Setting method at receive

Figures 2.5.16 and 2.5.17 show the UART of serial I/O1 setting method (at receive).



Process 1: Disable serial I/O1 transmit/receive interrupt.

(1) Disable interrupt for each vector

Interrupt control register 2
(ICON2: address $3F_{16}$)

Key input, serial I/O1 transmit, and serial I/O1 receive interrupts disabled
(Interrupt for each vector disabled)

(2) Disable interrupt for each interrupt factor

Interrupt source discrimination control register 1
(ICOND1: address 39_{16})

Serial I/O1 receive interrupt disabled
Serial I/O1 transmit interrupt disabled

Process 2: When BRG output/16 is selected as synchronous clock, set value to baud rate generator.

Baud rate generator
(BRG: address $1C_{16}$)

Set baud rate value

Process 3: Set serial I/O1 control register

Serial I/O1 control register
(SIO1CON: address $1A_{16}$)

BRG count source selected (set in internal clock selected)
0: $f(X_{IN})$
1: $f(X_{IN})/4$

Serial I/O synchronous clock selected
0: BRG output/16
1: External clock input/16

Transmit interrupt source selected (in full-duplex communication)
0: When transmit buffer has emptied
1: When transmit shift operation is completed

Transmit enable selected
0: Transmit disabled (in half-duplex communication)
1: Transmit enabled (in full-duplex communication) **(Note 1)**

Receive enabled

Clock asynchronous serial I/O

Serial I/O1 enabled
($P4_4$–$P4_7$ pins operate as serial I/O pins.)

Note 1: When an external clock input is selected as the synchronous clock, set "1" to the transmit enable bit while the synchronous clock is "H" state.

Process 4: Set UART control register

UART control register
(UARTCON: address $1B_{16}$)

Select character length (CHAS)
0: 8 bits
1: 7 bits

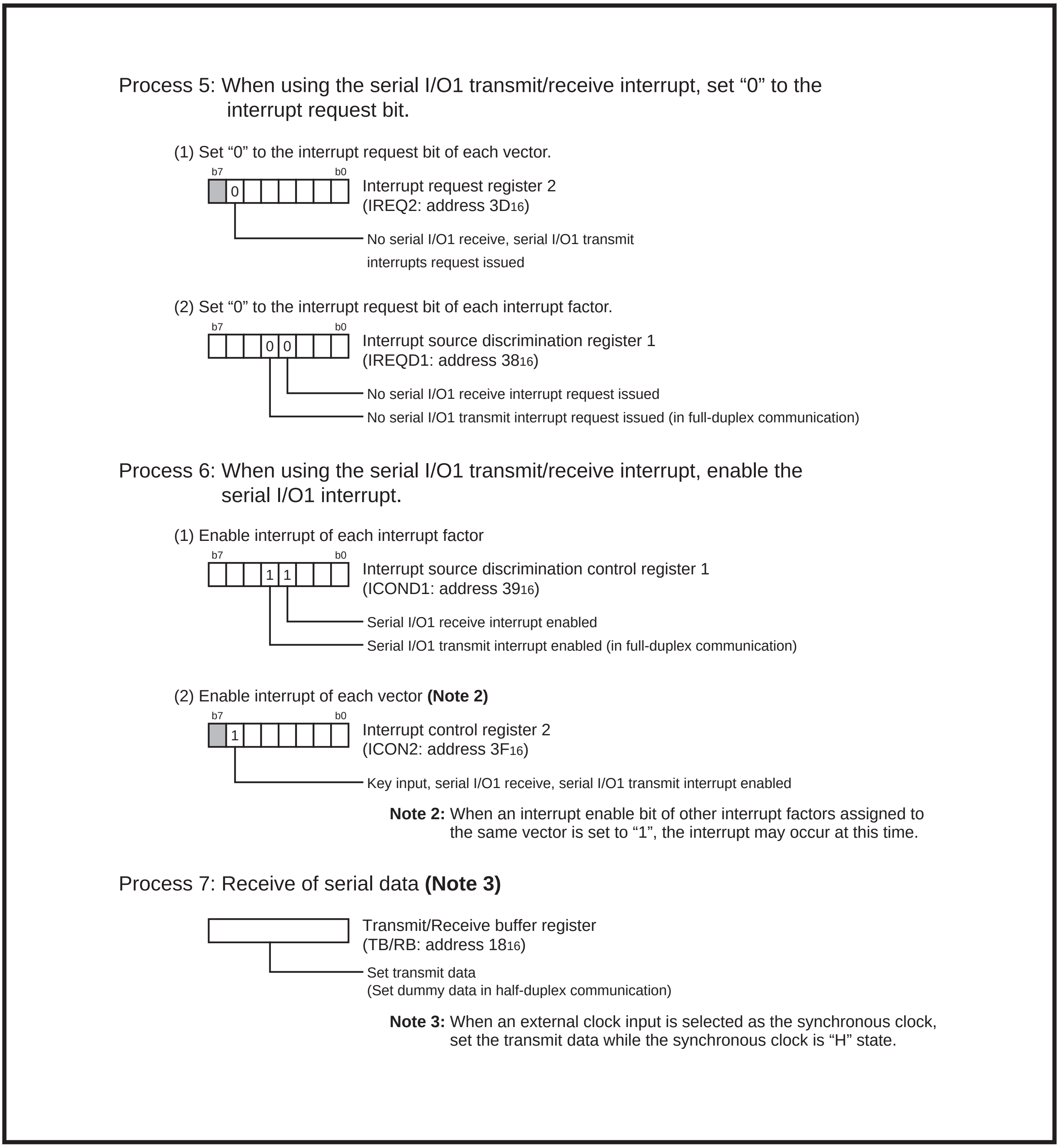
Select parity enable (PARE)
0: Parity disabled
1: Parity enabled

Select parity (PARS)
0: Even parity
1: Odd parity

Select stop bit length (STPS)
0: 1 stop bit
1: 2 stop bits

Select $P4_5$/TxD P-channel output disable
0: CMOS output
1: N-channel open-drain output

Fig. 2.5.16 UART of serial I/O1 setting method (at receive) (1)



Process 5: When using the serial I/O1 transmit/receive interrupt, set “0” to the interrupt request bit.

(1) Set “0” to the interrupt request bit of each vector.

Interrupt request register 2
(IREQ2: address $3D_{16}$)

No serial I/O1 receive, serial I/O1 transmit interrupts request issued

(2) Set “0” to the interrupt request bit of each interrupt factor.

Interrupt source discrimination register 1
(IREQD1: address 38_{16})

No serial I/O1 receive interrupt request issued

No serial I/O1 transmit interrupt request issued (in full-duplex communication)

Process 6: When using the serial I/O1 transmit/receive interrupt, enable the serial I/O1 interrupt.

(1) Enable interrupt of each interrupt factor

Interrupt source discrimination control register 1
(ICOND1: address 39_{16})

Serial I/O1 receive interrupt enabled

Serial I/O1 transmit interrupt enabled (in full-duplex communication)

(2) Enable interrupt of each vector **(Note 2)**

Interrupt control register 2
(ICON2: address $3F_{16}$)

Key input, serial I/O1 receive, serial I/O1 transmit interrupt enabled

Note 2: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to “1”, the interrupt may occur at this time.

Process 7: Receive of serial data **(Note 3)**

Transmit/Receive buffer register
(TB/RB: address 18_{16})

Set transmit data
(Set dummy data in half-duplex communication)

Note 3: When an external clock input is selected as the synchronous clock, set the transmit data while the synchronous clock is “H” state.

Fig. 2.5.17 UART of serial I/O1 setting method (at receive) (2)

(2) Clock asynchronous serial I/O of serial I/O1 application example

■ **Outline**

The clock asynchronous communication is performed between the 3874 groups.

■ **Specifications**

3874 group ① (transmitter in half-duplex communication)
•Baud rate: $f(X_{IN})/4 \times (XX_{16}* + 1) \times 16$ bps
3874 group ② (receiver in half-duplex communication)
•Baud rate: $f(X_{IN})/4 \times (XX_{16}* + 1) \times 16$ bps

$XX_{16}*$: Setting value of baud rate generator
Figure 2.5.18 shows an example of connections and Figure 2.5.19 shows an example of control procedure.

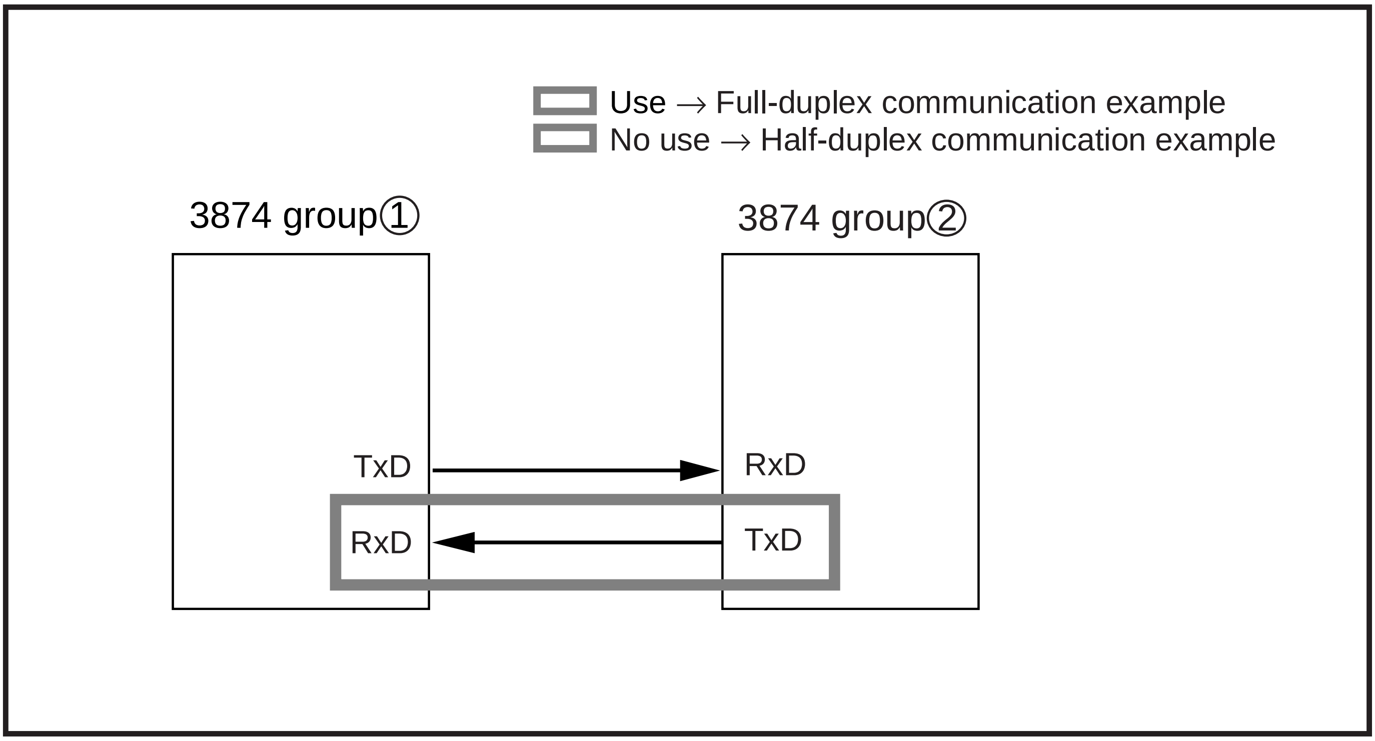
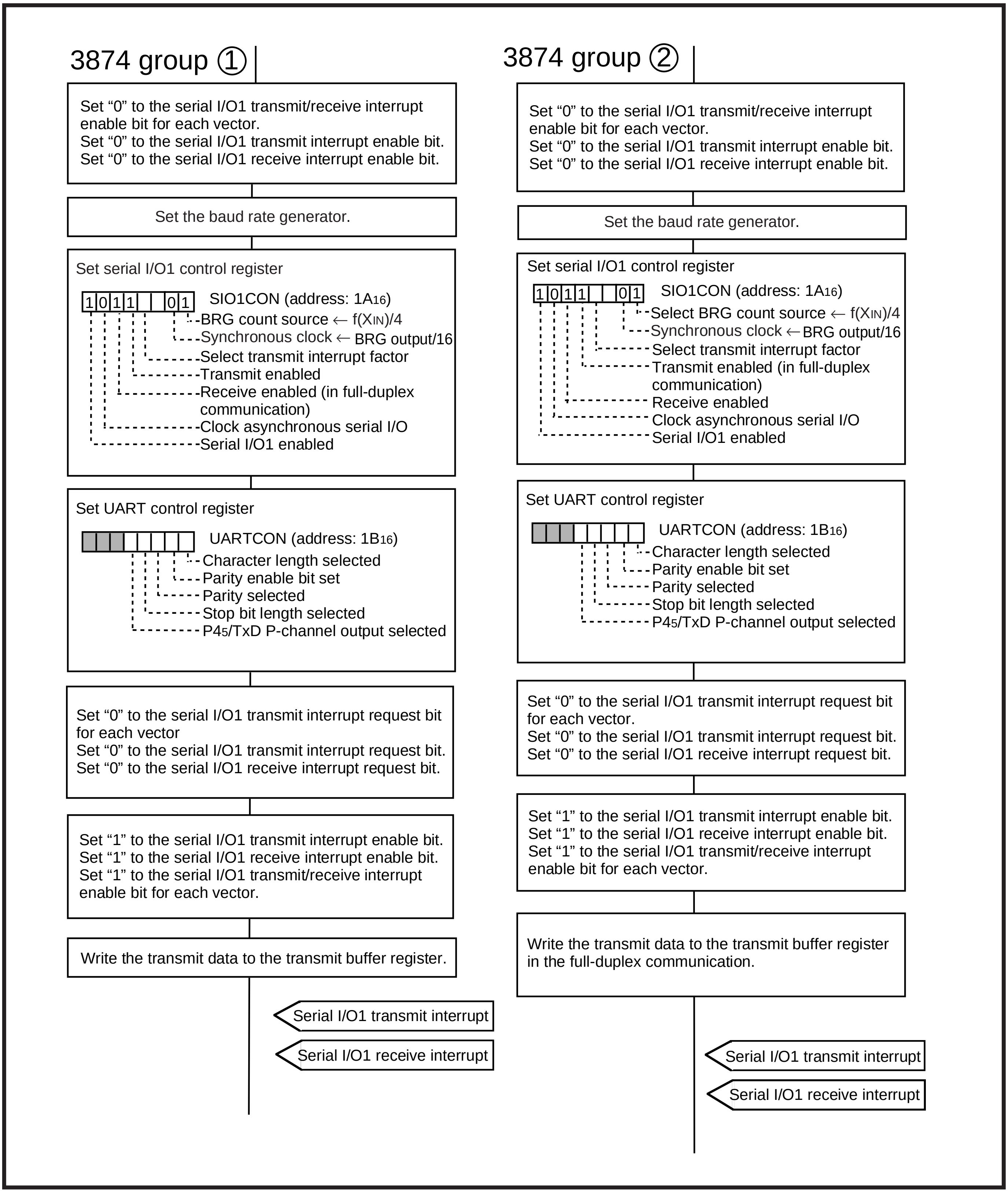

Use → Full-duplex communication example
No use → Half-duplex communication example
3874 group①
3874 group②
TxD
RxD
RxD
TxD

Fig. 2.5.18 Example of connections (clock asynchronous serial I/O mode of serial I/O1)



3874 group ①

Set “0” to the serial I/O1 transmit/receive interrupt enable bit for each vector.
Set “0” to the serial I/O1 transmit interrupt enable bit.
Set “0” to the serial I/O1 receive interrupt enable bit.

Set the baud rate generator.

Set serial I/O1 control register

1 0 1 1 0 1
SIO1CON (address: 1A16)
BRG count source ← f(XIN)/4
Synchronous clock ← BRG output/16
Select transmit interrupt factor
Transmit enabled
Receive enabled (in full-duplex communication)
Clock asynchronous serial I/O
Serial I/O1 enabled

Set UART control register

UARTCON (address: 1B16)
Character length selected
Parity enable bit set
Parity selected
Stop bit length selected
P45/TxD P-channel output selected

Set “0” to the serial I/O1 transmit interrupt request bit for each vector
Set “0” to the serial I/O1 transmit interrupt request bit.
Set “0” to the serial I/O1 receive interrupt request bit.

Set “1” to the serial I/O1 transmit interrupt enable bit.
Set “1” to the serial I/O1 receive interrupt enable bit.
Set “1” to the serial I/O1 transmit/receive interrupt enable bit for each vector.

Write the transmit data to the transmit buffer register.

3874 group ②

Set “0” to the serial I/O1 transmit/receive interrupt enable bit for each vector.
Set “0” to the serial I/O1 transmit interrupt enable bit.
Set “0” to the serial I/O1 receive interrupt enable bit.

Set the baud rate generator.

Set serial I/O1 control register

1 0 1 1 0 1
SIO1CON (address: 1A16)
Select BRG count source ← f(XIN)/4
Synchronous clock ← BRG output/16
Select transmit interrupt factor
Transmit enabled (in full-duplex communication)
Receive enabled
Clock asynchronous serial I/O
Serial I/O1 enabled

Set UART control register

Set “0” to the serial I/O1 transmit interrupt request bit for each vector.
Set “0” to the serial I/O1 transmit interrupt request bit.
Set “0” to the serial I/O1 receive interrupt request bit.

Set “1” to the serial I/O1 transmit interrupt enable bit.
Set “1” to the serial I/O1 receive interrupt enable bit.
Set “1” to the serial I/O1 transmit/receive interrupt enable bit for each vector.

Write the transmit data to the transmit buffer register in the full-duplex communication.

Fig. 2.5.19 Example of control procedure

2.5.5 Notes on serial I/O1

(1) Notes when selecting clock synchronous serial I/O

① **Stop of transmission operation**
As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the serial I/O1 enable bit and the transmit enable bit to “0” (serial I/O1 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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, S_{CLK1}, and $\overline{S_{RDY1}}$ 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 TxD pin and an operation failure occurs.

② **Stop of receive operation**
As for the serial I/O1 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/O1 enable bit to “0” (serial I/O1 disabled).

③ **Stop of transmit/receive operation**
As for the serial I/O1 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/O1 enable bit to “0” (serial I/O1 disabled) (refer to (1) ①).

(2) Notes when selecting clock asynchronous serial I/O

① **Stop of transmission operation**
As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the transmit enable bit to “0” (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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, 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 TxD pin and an operation failure occurs.

② **Stop of receive operation**
As for the serial I/O1 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).

③ **Stop of transmit/receive operation**

● **Only transmission operation is stopped**
As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the transmit enable bit to “0” (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 TxD, RxD, 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/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

● **Only receive operation is stopped**
As for the serial I/O1 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).

(3) $\overline{\text{SRDY1}}$ output of reception side
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/O control register again
Set the serial I/O 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.”

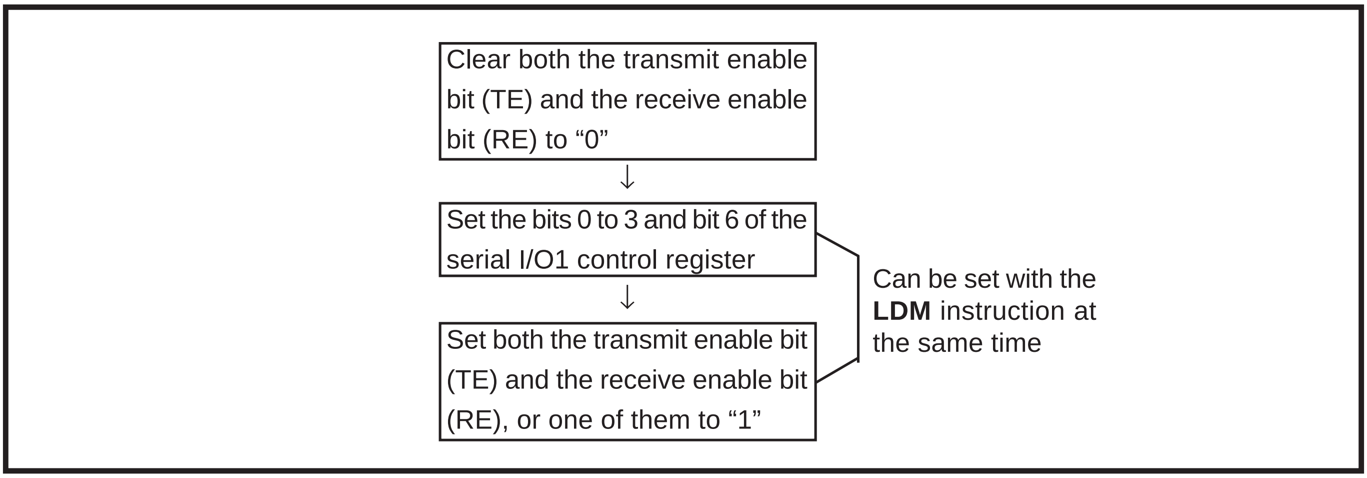

Clear both the transmit enable bit (TE) and the receive enable bit (RE) to “0”
Set the bits 0 to 3 and bit 6 of the serial I/O1 control register
Set both the transmit enable bit (TE) and the receive enable bit (RE), or one of them to “1”
Can be set with the **LDM** instruction at the same time

Fig. 2.5.20 Sequence of setting serial I/O1 control register again

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

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 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

The transmit interrupt request bit is set and the interrupt request occurs even when selecting timing that either of the following flags is set to “1” as timing where the transmit interrupt occurs.

- Transmit buffer empty flag is set to “1”
- Transmit shift register completion flag is set to “1”

Accordingly, when the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as the following sequence.

① Transmit enable bit is set to “1”
② Transmit interrupt request bit is set to “0”

● **Reason**

When the transmit enable bit is set to “1”, the transmit buffer empty flag and transmit shift register completion flag are set to “1”.

(8) Use of TxD pin

The $P4_5$/TxD P-channel output disable bit of the UART control register is valid both when using as a normal port and when using as the TxD pin. However, do not apply a voltage of Vcc + 0.3 V or more to the $P4_5$/TxD pin even when it is used as N-channel open drain output.

In the serial I/O1, after transmit is completed, the TxD pin continue latching and outputting the last bit.

2.5.6 Serial I/O2 memory assignment

Serial I/O2 can be used only as clock synchronous serial I/O.
Figure 2.5.21 shows the memory assignment of the serial I/O2 relevant registers. Each of these registers is described below.

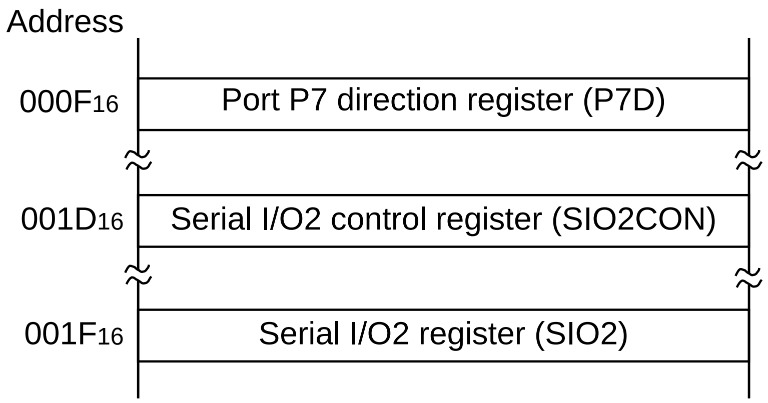

Address
000F16
Port P7 direction register (P7D)
001D16
Serial I/O2 control register (SIO2CON)
001F16
Serial I/O2 register (SIO2)

Fig. 2.5.21 Memory assignment of serial I/O2 relevant registers

2.5.7 Serial I/O2 relevant registers

(1) Port P7 direction register

■ **Selecting of port S_{IN2}/$P7_0$ direction mode**

When using serial I/O2, set “0” to bit 0 to use the S_{IN2}/$P7_0$ as the input mode.

■ **Selecting of port S_{OUT2}/$P7_1$ direction mode**

When using serial I/O2, set “1” to bit 1 to use the S_{OUT2}/$P7_1$ as the output mode.

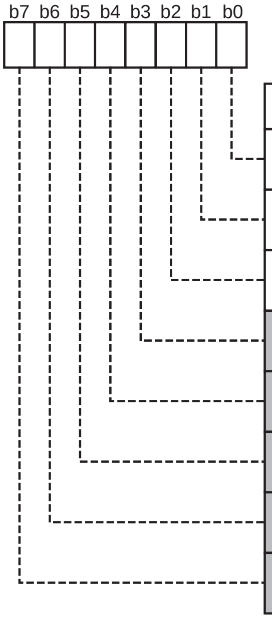
■ **Selecting of port S_{CLK2}/$P7_2$ direction mode**

When using an external clock as the synchronous clock, set “0” to bit 2 to use the S_{CLK2}/$P7_2$ as the input mode.

When providing the synchronous clock to the external, set “1” to bit 2 to use the S_{CLK2}/$P7_2$ as output mode.

Figure 2.5.22 shows the structure of the port P7 direction register.

Port P7 direction register



Port P7 direction register
($P7D$: address $0F_{16}$)

b	Name	Functions	At reset	R	W
0	Port P7 direction register	0 : Port $P7_0$ input mode 1 : Port $P7_0$ output mode	0	×	○
1		0 : Port $P7_1$ input mode 1 : Port $P7_1$ output mode	0	×	○
2		0 : Port $P7_2$ input mode 1 : Port $P7_2$ output mode	0	×	○
3		0 : Port $P7_3$ input mode 1 : Port $P7_3$ output mode	0	×	○
4		0 : Port $P7_4$ input mode 1 : Port $P7_4$ output mode	0	×	○
5		0 : Port $P7_5$ input mode 1 : Port $P7_5$ output mode	0	×	○
6		0 : Port $P7_6$ input mode 1 : Port $P7_6$ output mode	0	×	○
7		0 : Port $P7_7$ input mode 1 : Port $P7_7$ output mode	0	×	○

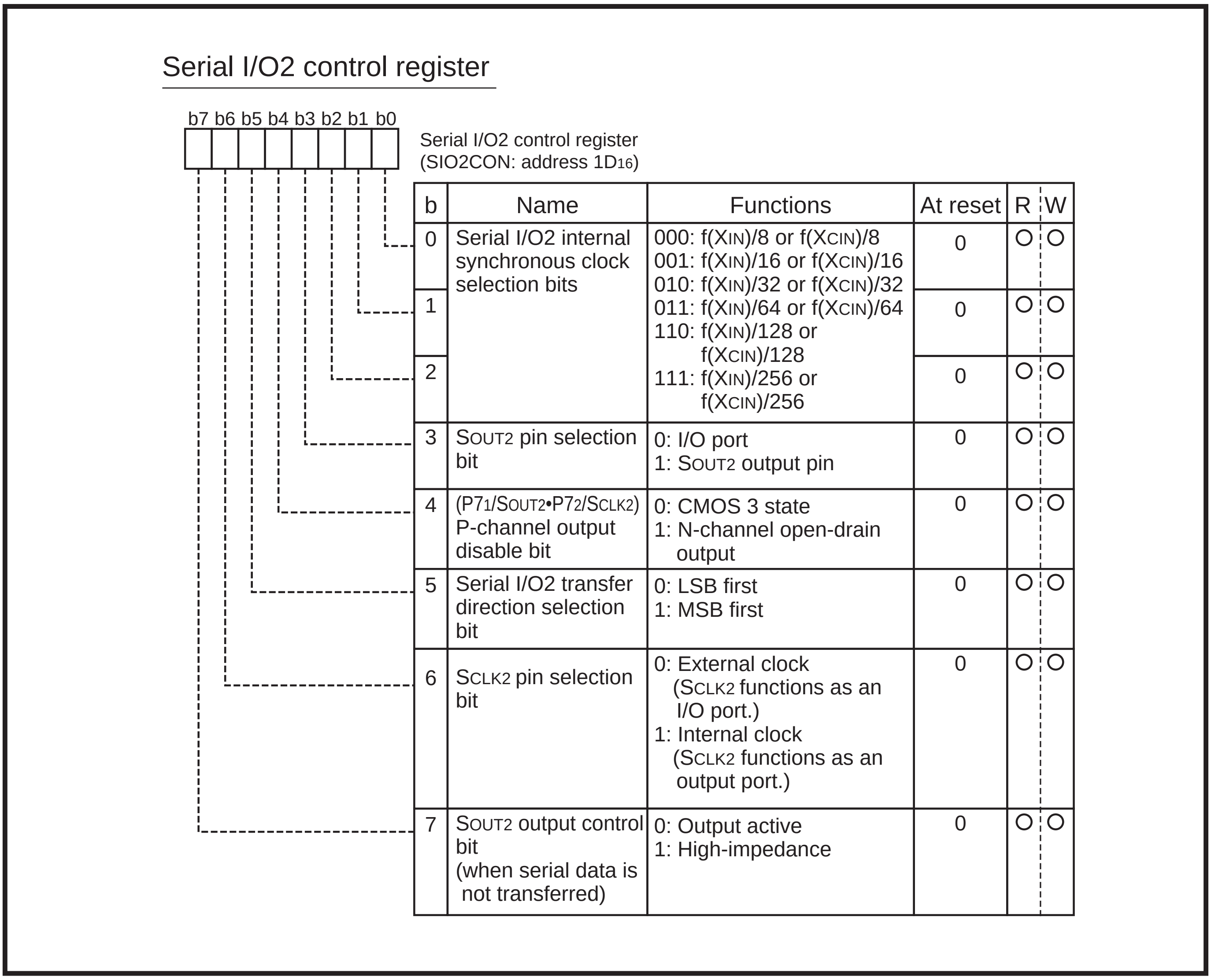
Note : The value of the port P7 direction register cannot be read out.

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
	ADT	$\overline{BUS_{IN}}$	$\overline{BUS_{OUT}}$			S_{CLK2}	S_{OUT2}	S_{IN2}

Fig. 2.5.22 Structure of port P7 direction register

(2) Serial I/O2 control register

This register controls various functions relevant to the serial I/O2. Figure 2.5.23 shows the structure of the serial I/O2 control register.



Serial I/O2 control register
b7 b6 b5 b4 b3 b2 b1 b0
Serial I/O2 control register
(SIO2CON: address 1D$_{16}$)

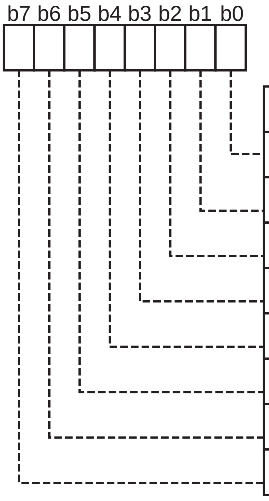
b	Name	Functions	At reset	R	W
0	Serial I/O2 internal synchronous clock selection bits	000: f(XIN)/8 or f(XCIN)/8 001: f(XIN)/16 or f(XCIN)/16 010: f(XIN)/32 or f(XCIN)/32 011: f(XIN)/64 or f(XCIN)/64 110: f(XIN)/128 or f(XCIN)/128 111: f(XIN)/256 or f(XCIN)/256	0	○	○
1			0	○	○
2			0	○	○
3	SOUT2 pin selection bit	0: I/O port 1: SOUT2 output pin	0	○	○
4	(P7$_1$/SOUT2•P7$_2$/SCLK2) P-channel output disable bit	0: CMOS 3 state 1: N-channel open-drain output	0	○	○
5	Serial I/O2 transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
6	SCLK2 pin selection bit	0: External clock (SCLK2 functions as an I/O port.) 1: Internal clock (SCLK2 functions as an output port.)	0	○	○
7	SOUT2 output control bit (when serial data is not transferred)	0: Output active 1: High-impedance	0	○	○

Fig. 2.5.23 Structure of serial I/O2 control register

(3) Serial I/O2 register

When the transfer data is written to the serial I/O2 register, transmit/receive is started. Figure 2.5.24 shows the structure of the serial I/O2 register.

Serial I/O2 register



Serial I/O2 register
(SIO2: address $1F_{16}$)

b	Functions	At reset	R	W
0	This is the serial I/O2 exclusive transmit/receive buffer register. At transmit: “00_{16}” to “FF_{16}” can be set as the transmit data. The transmit data is automatically transferred to the transmit shift register by writing the transmit data. At receive: When all receive data has been input into the receive shift register, the receive data is automatically transferred to this register.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 2.5.24 Structure of serial I/O2 register

2.5.8 Clock synchronous serial I/O of serial I/O2

(1) Serial I/O2 setting method

■ Setting method at transmit

Figures 2.5.25 and 2.5.26 show the serial I/O2 setting method (at transmit).

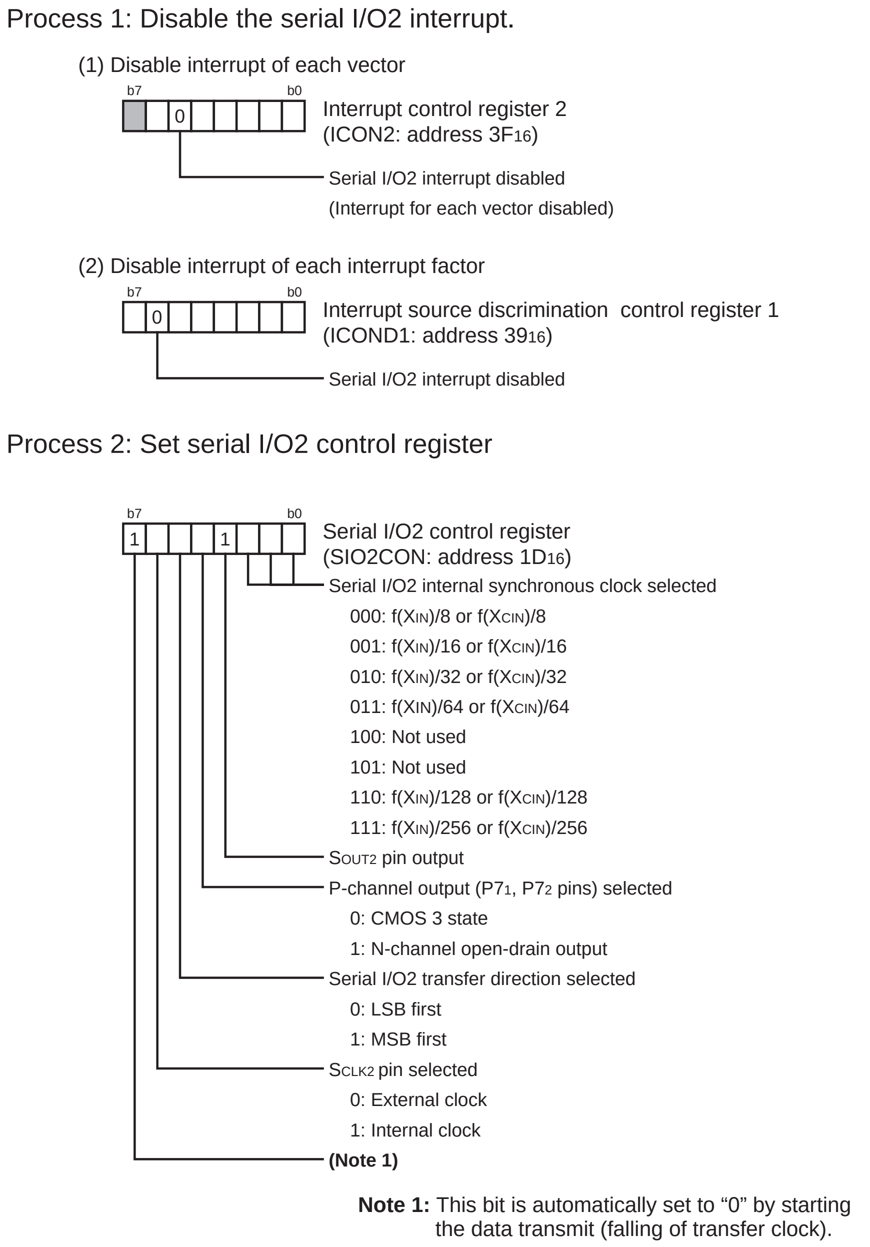
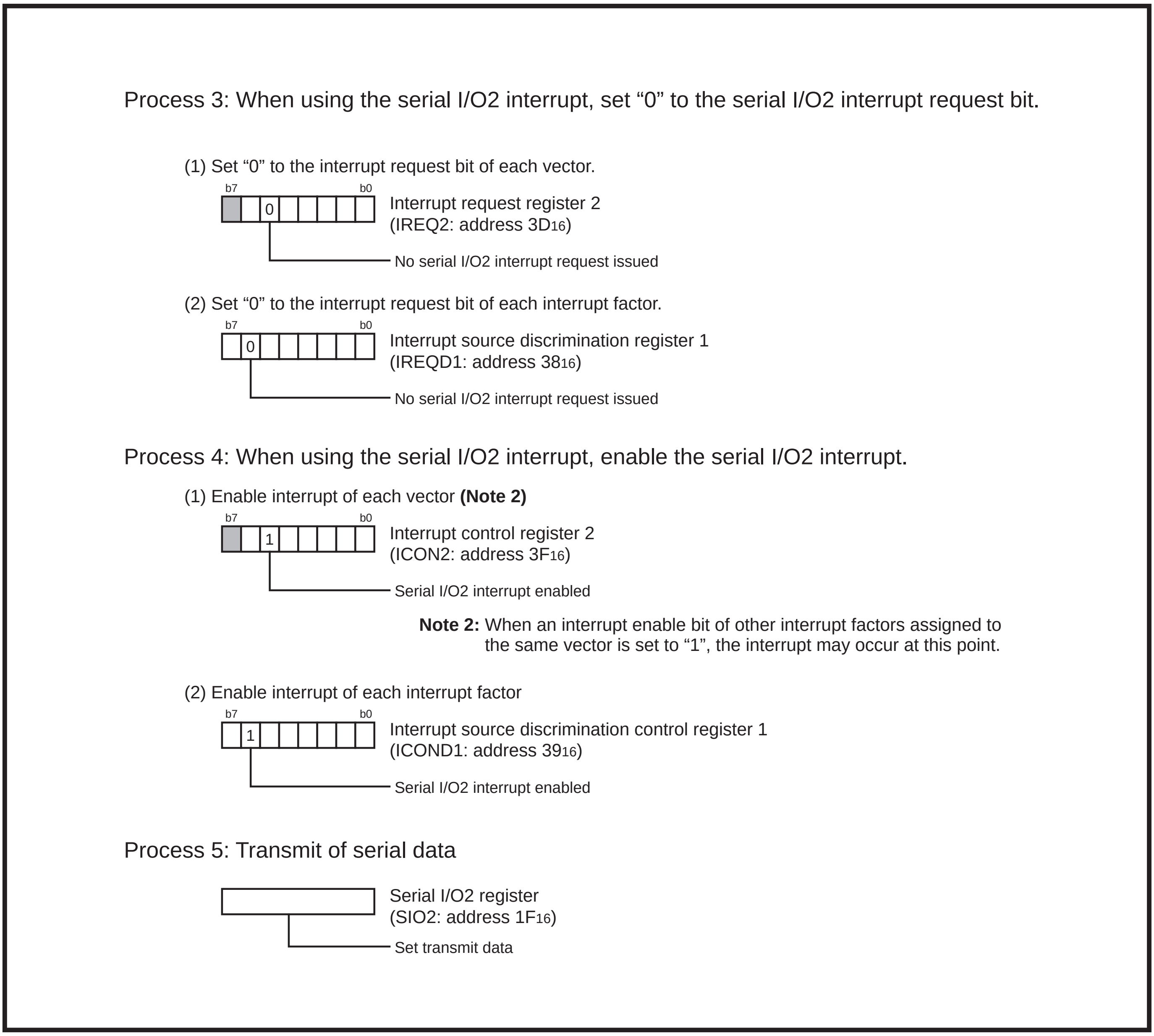

Process 1: Disable the serial I/O2 interrupt.
(1) Disable interrupt of each vector
b7
b0
0
Interrupt control register 2
(ICON2: address 3F16)
Serial I/O2 interrupt disabled
(Interrupt for each vector disabled)
(2) Disable interrupt of each interrupt factor
b7
b0
0
Interrupt source discrimination control register 1
(ICOND1: address 3916)
Serial I/O2 interrupt disabled
Process 2: Set serial I/O2 control register
b7
b0
1
1
Serial I/O2 control register
(SIO2CON: address 1D16)
Serial I/O2 internal synchronous clock selected
000: f(XIN)/8 or f(XCIN)/8
001: f(XIN)/16 or f(XCIN)/16
010: f(XIN)/32 or f(XCIN)/32
011: f(XIN)/64 or f(XCIN)/64
100: Not used
101: Not used
110: f(XIN)/128 or f(XCIN)/128
111: f(XIN)/256 or f(XCIN)/256
SOUT2 pin output
P-channel output (P71, P72 pins) selected
0: CMOS 3 state
1: N-channel open-drain output
Serial I/O2 transfer direction selected
0: LSB first
1: MSB first
SCLK2 pin selected
0: External clock
1: Internal clock
(Note 1)
Note 1: This bit is automatically set to “0” by starting the data transmit (falling of transfer clock).

Fig. 2.5.25 Serial I/O2 setting method (at transmit) (1)



Process 3: When using the serial I/O2 interrupt, set “0” to the serial I/O2 interrupt request bit.

(1) Set “0” to the interrupt request bit of each vector.

b7
b0
0
Interrupt request register 2
(IREQ2: address 3D16)
No serial I/O2 interrupt request issued

(2) Set “0” to the interrupt request bit of each interrupt factor.

b7
b0
0
Interrupt source discrimination register 1
(IREQD1: address 3816)
No serial I/O2 interrupt request issued

Process 4: When using the serial I/O2 interrupt, enable the serial I/O2 interrupt.

(1) Enable interrupt of each vector **(Note 2)**

b7
b0
1
Interrupt control register 2
(ICON2: address 3F16)
Serial I/O2 interrupt enabled

Note 2: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to “1”, the interrupt may occur at this point.

(2) Enable interrupt of each interrupt factor

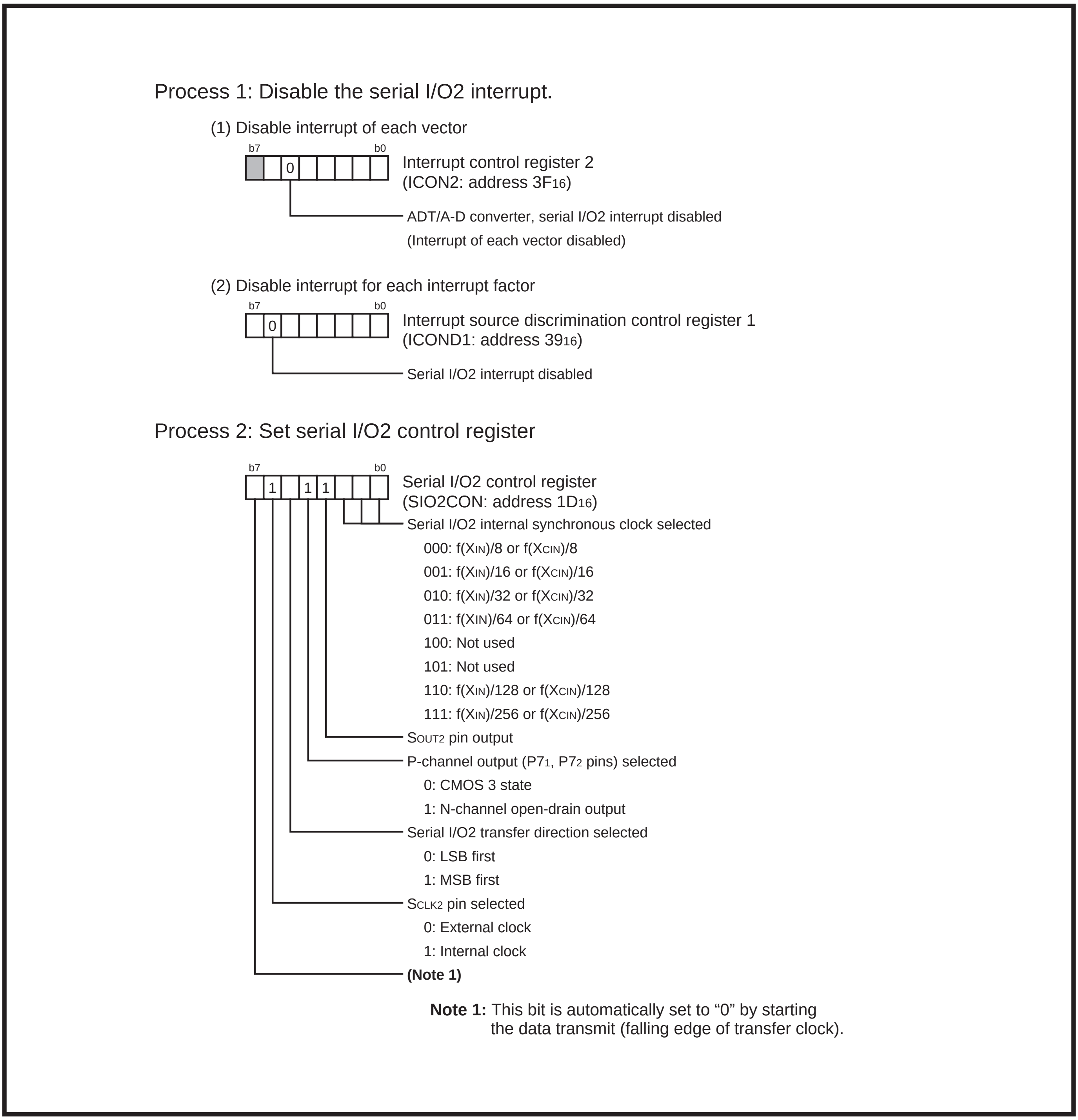
b7
b0
1
Interrupt source discrimination control register 1
(ICOND1: address 3916)
Serial I/O2 interrupt enabled

Process 5: Transmit of serial data

Fig. 2.5.26 Serial I/O2 setting method (at transmit) (2)

■ Setting method at receive

Figures 2.5.27 and 2.5.28 show the serial I/O2 setting method (at receive).



Process 1: Disable the serial I/O2 interrupt.

(1) Disable interrupt of each vector

Interrupt control register 2
(ICON2: address $3F_{16}$)

ADT/A-D converter, serial I/O2 interrupt disabled
(Interrupt of each vector disabled)

(2) Disable interrupt for each interrupt factor

Interrupt source discrimination control register 1
(ICOND1: address 39_{16})

Serial I/O2 interrupt disabled

Process 2: Set serial I/O2 control register

Serial I/O2 control register
(SIO2CON: address $1D_{16}$)

Serial I/O2 internal synchronous clock selected
000: f(X_{IN})/8 or f(X_{CIN})/8
001: f(X_{IN})/16 or f(X_{CIN})/16
010: f(X_{IN})/32 or f(X_{CIN})/32
011: f(X_{IN})/64 or f(X_{CIN})/64
100: Not used
101: Not used
110: f(X_{IN})/128 or f(X_{CIN})/128
111: f(X_{IN})/256 or f(X_{CIN})/256

S_{OUT2} pin output

P-channel output (P7_1, P7_2 pins) selected
0: CMOS 3 state
1: N-channel open-drain output

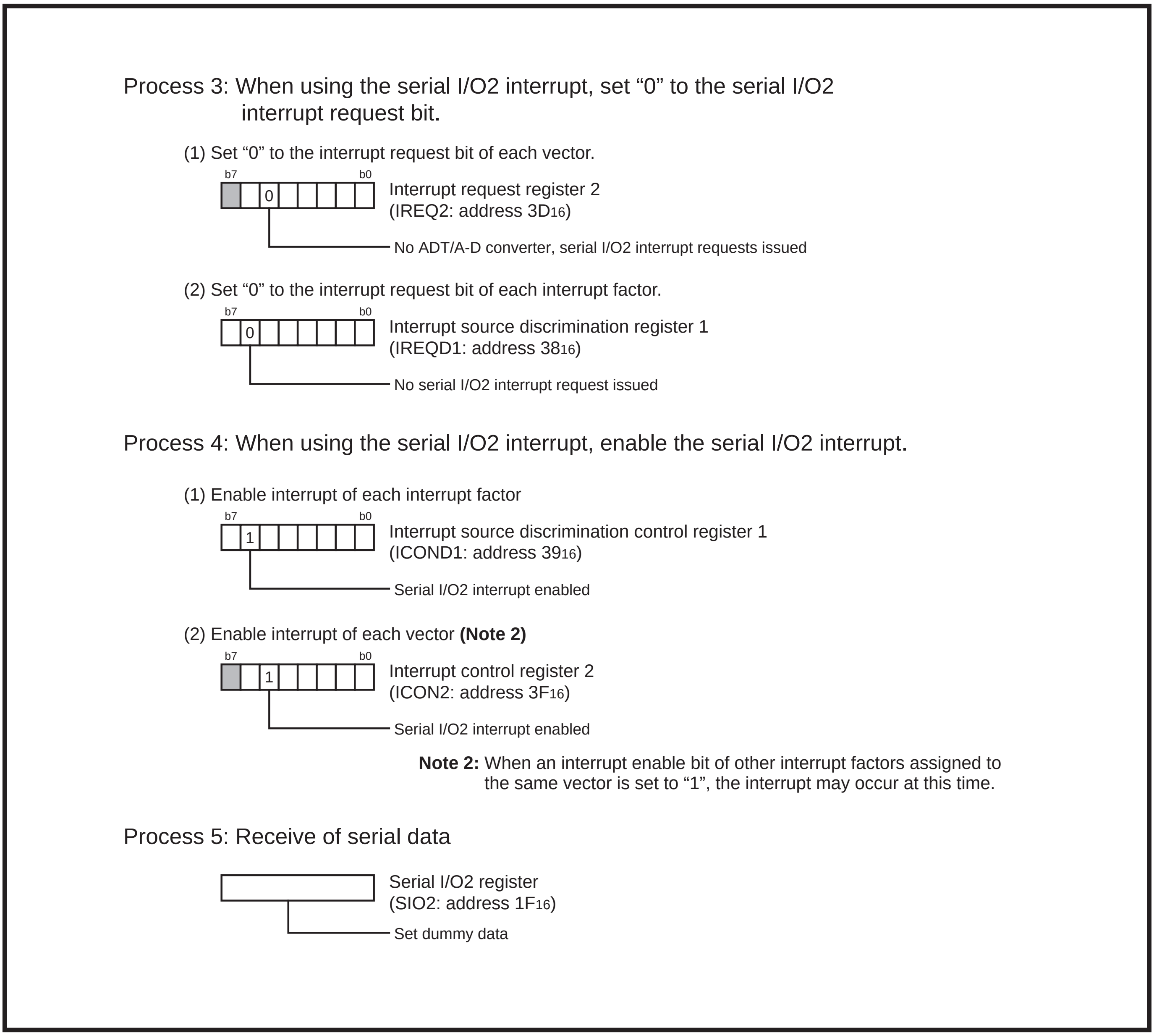
Serial I/O2 transfer direction selected
0: LSB first
1: MSB first

S_{CLK2} pin selected
0: External clock
1: Internal clock

(Note 1)

Note 1: This bit is automatically set to “0” by starting the data transmit (falling edge of transfer clock).

Fig. 2.5.27 Serial I/O2 setting method (at receive) (1)



Process 3: When using the serial I/O2 interrupt, set “0” to the serial I/O2 interrupt request bit.

(1) Set “0” to the interrupt request bit of each vector.

Interrupt request register 2
(IREQ2: address $3D_{16}$)

No ADT/A-D converter, serial I/O2 interrupt requests issued

(2) Set “0” to the interrupt request bit of each interrupt factor.

Interrupt source discrimination register 1
(IREQD1: address 38_{16})

No serial I/O2 interrupt request issued

Process 4: When using the serial I/O2 interrupt, enable the serial I/O2 interrupt.

(1) Enable interrupt of each interrupt factor

Interrupt source discrimination control register 1
(ICOND1: address 39_{16})

Serial I/O2 interrupt enabled

(2) Enable interrupt of each vector **(Note 2)**

Interrupt control register 2
(ICON2: address $3F_{16}$)

Serial I/O2 interrupt enabled

Note 2: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to “1”, the interrupt may occur at this time.

Process 5: Receive of serial data

Serial I/O2 register
(SIO2: address $1F_{16}$)

Set dummy data

Fig. 2.5.28 Serial I/O2 setting method (at receive) (2)

(2) Clock synchronous serial I/O of serial I/O2 application examples

■ Application example 1 (transmit data to E^{2}PROM: write control)

● **Outline**

Writing data is performed to E^{2}PROM by using serial I/O2.

● **Specifications**

•Synchronous clock: 25 kHz (dividing f(X_{IN}) = 6.4 MHz by 256)
•Transfer direction: LSB first
•Serial I/O2 interrupt is not used.
•Transmit control by connecting port $P5_5$ and $\overline{CS}$ pin (“L” level active) of peripheral IC
•M6M80011AP (Mitsubishi) is used as E^{2}PROM

Figure 2.5.29 shows the peripheral circuit example and Figure 2.5.30 shows the timing in write mode.

Figures 2.5.31 and 2.5.32 show the control procedure example.

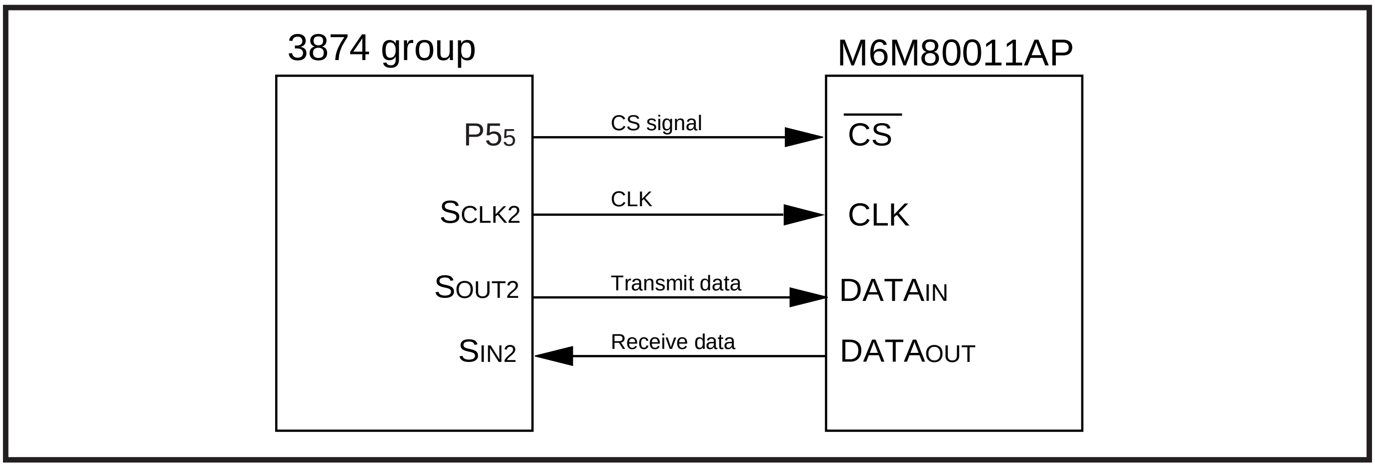
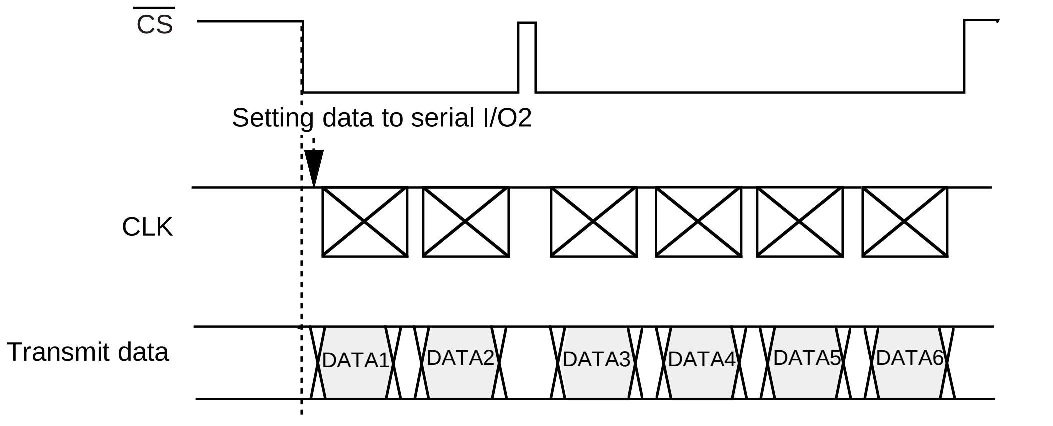
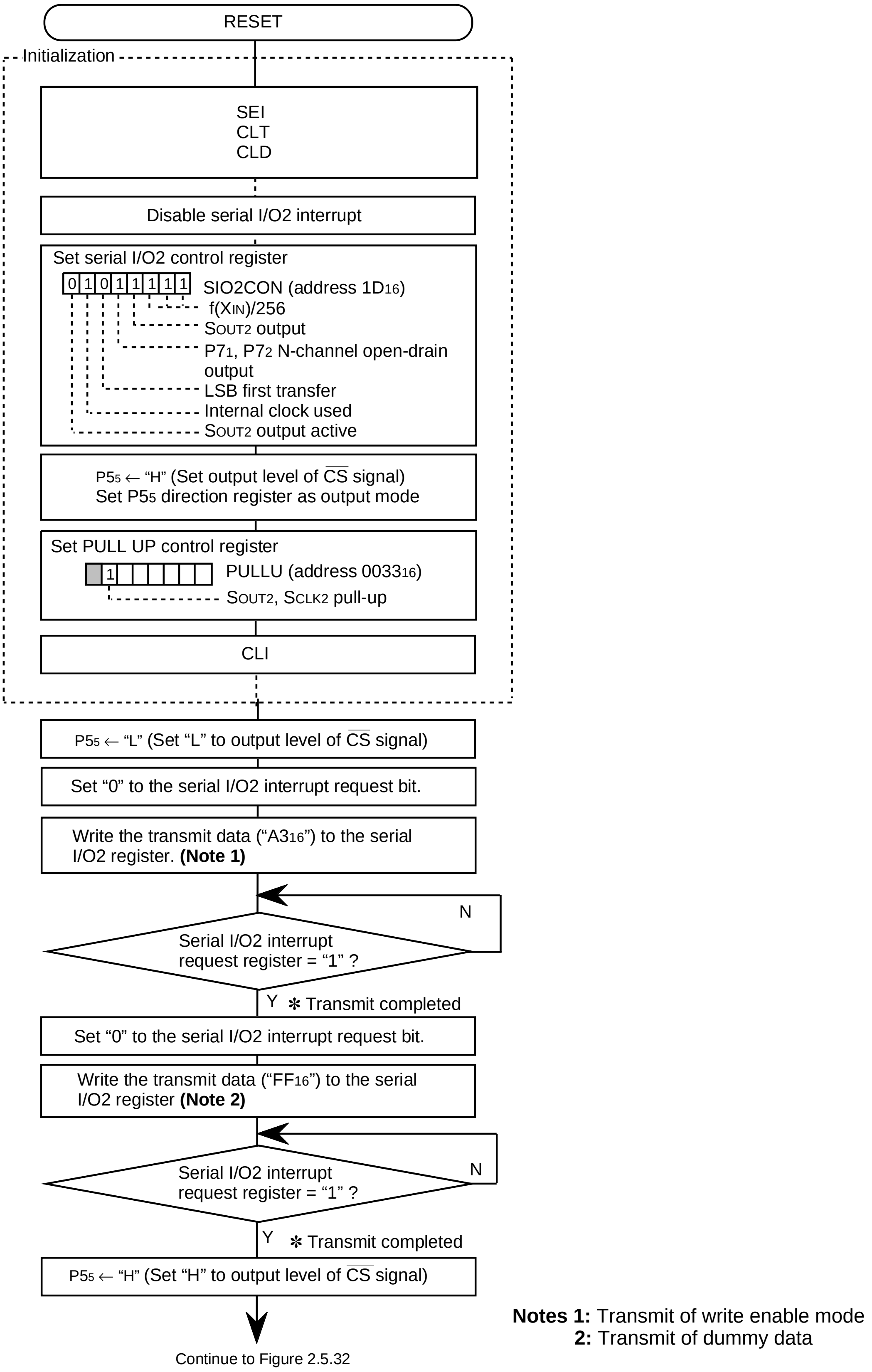


Fig. 2.5.29 Peripheral circuit example



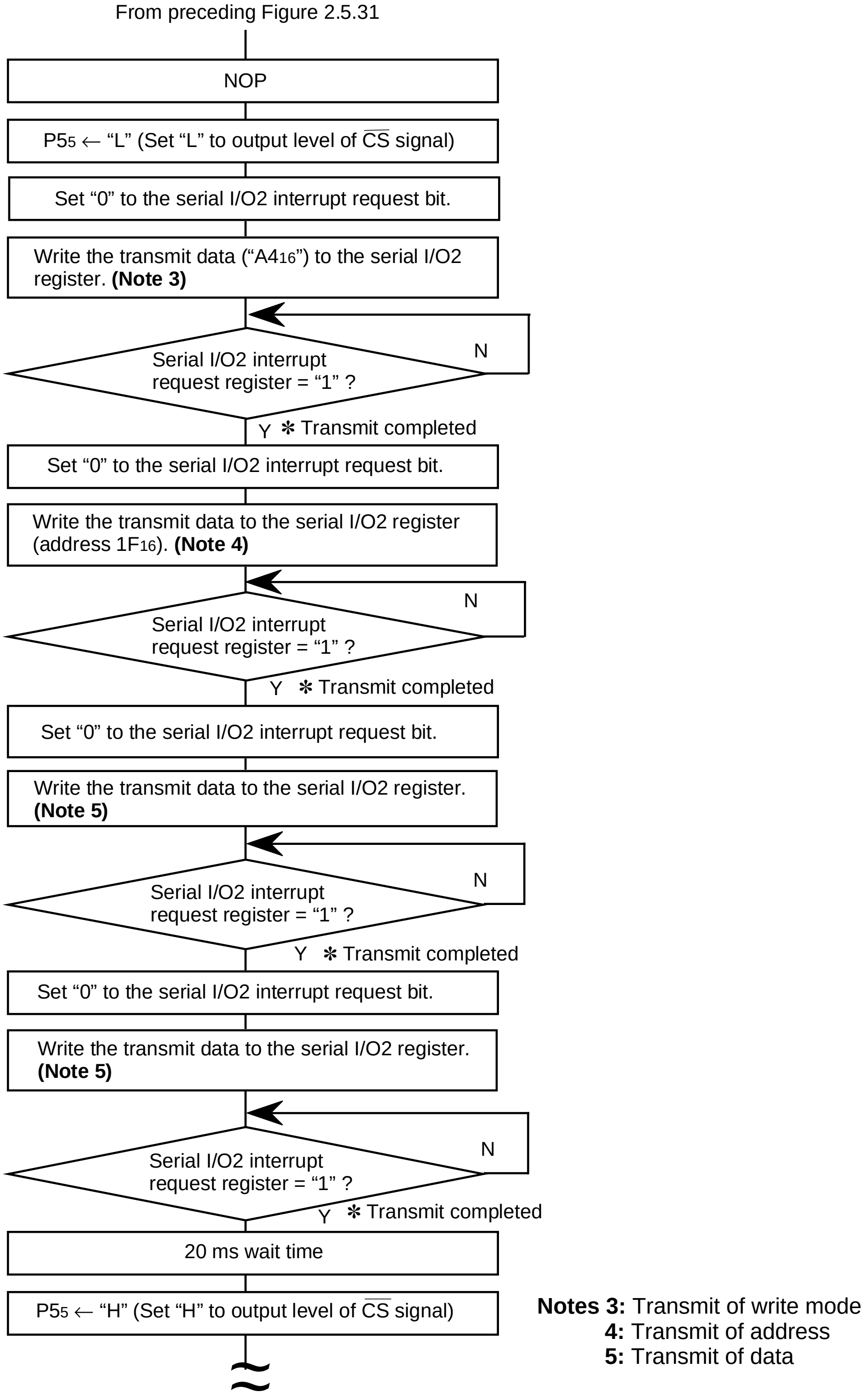
DATA1 : Write enable mode ($A3_{16}$)
DATA2 : Dummy data
DATA3 : Write mode ($A4_{16}$)
DATA4 : Address
DATA5 : Data
DATA6 : Data

Fig. 2.5.30 Timing in write mode


RESET
Initialization
SEI
CLT
CLD
Disable serial I/O2 interrupt
Set serial I/O2 control register
0 1 0 1 1 1 1 1
SIO2CON (address 1D16)
f(XIN)/256
SOUT2 output
P71, P72 N-channel open-drain output
LSB first transfer
Internal clock used
SOUT2 output active
P55 ← "H" (Set output level of CS signal)
Set P55 direction register as output mode
Set PULL UP control register
1
PULLU (address 003316)
SOUT2, SCLK2 pull-up
CLI
P55 ← "L" (Set "L" to output level of CS signal)
Set "0" to the serial I/O2 interrupt request bit.
Write the transmit data ("A316") to the serial I/O2 register. (Note 1)
N
Serial I/O2 interrupt request register = "1" ?
Y
✻ Transmit completed
Set "0" to the serial I/O2 interrupt request bit.
Write the transmit data ("FF16") to the serial I/O2 register (Note 2)
N
Serial I/O2 interrupt request register = "1" ?
Y
✻ Transmit completed
P55 ← "H" (Set "H" to output level of CS signal)
Continue to Figure 2.5.32

Notes 1: Transmit of write enable mode
2: Transmit of dummy data

Fig. 2.5.31 Control procedure example (1)


From preceding Figure 2.5.31
NOP
P5$_5$ ← "L" (Set "L" to output level of $\overline{\text{CS}}$ signal)
Set "0" to the serial I/O2 interrupt request bit.
Write the transmit data ("A4$_{16}$") to the serial I/O2 register. (Note 3)
Serial I/O2 interrupt request register = "1" ?
N
Y
✻ Transmit completed
Set "0" to the serial I/O2 interrupt request bit.
Write the transmit data to the serial I/O2 register (address 1F$_{16}$). (Note 4)
Serial I/O2 interrupt request register = "1" ?
N
Y
✻ Transmit completed
Set "0" to the serial I/O2 interrupt request bit.
Write the transmit data to the serial I/O2 register. (Note 5)
Serial I/O2 interrupt request register = "1" ?
N
Y
✻ Transmit completed
Set "0" to the serial I/O2 interrupt request bit.
Write the transmit data to the serial I/O2 register. (Note 5)
Serial I/O2 interrupt request register = "1" ?
N
Y
✻ Transmit completed
20 ms wait time
P5$_5$ ← "H" (Set "H" to output level of $\overline{\text{CS}}$ signal)
≈

Notes 3: Transmit of write mode
4: Transmit of address
5: Transmit of data

Fig. 2.5.32 Control procedure example (2)

■ Application example 2 (receive data from E²PROM: read control)

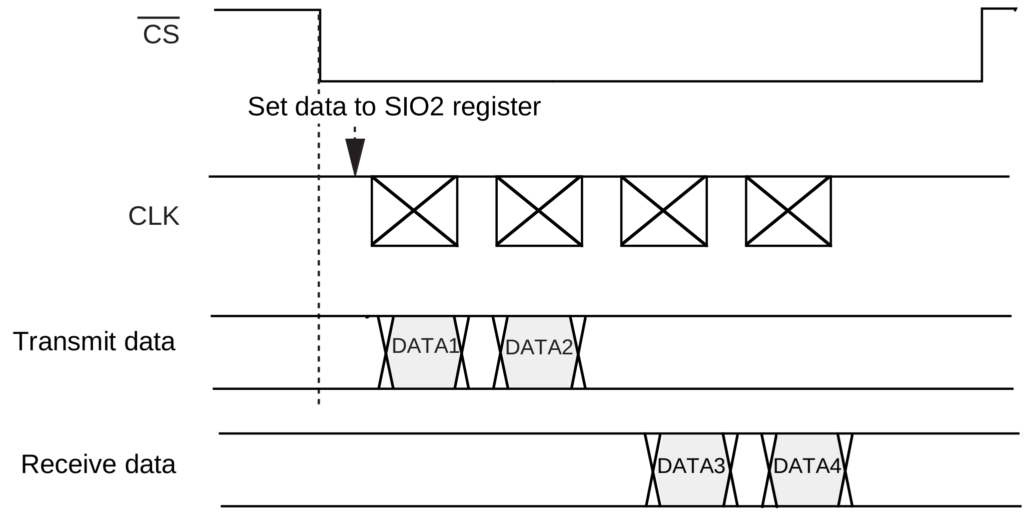
● Outline

Data is read from E²PROM by using serial I/O2.

● Specifications

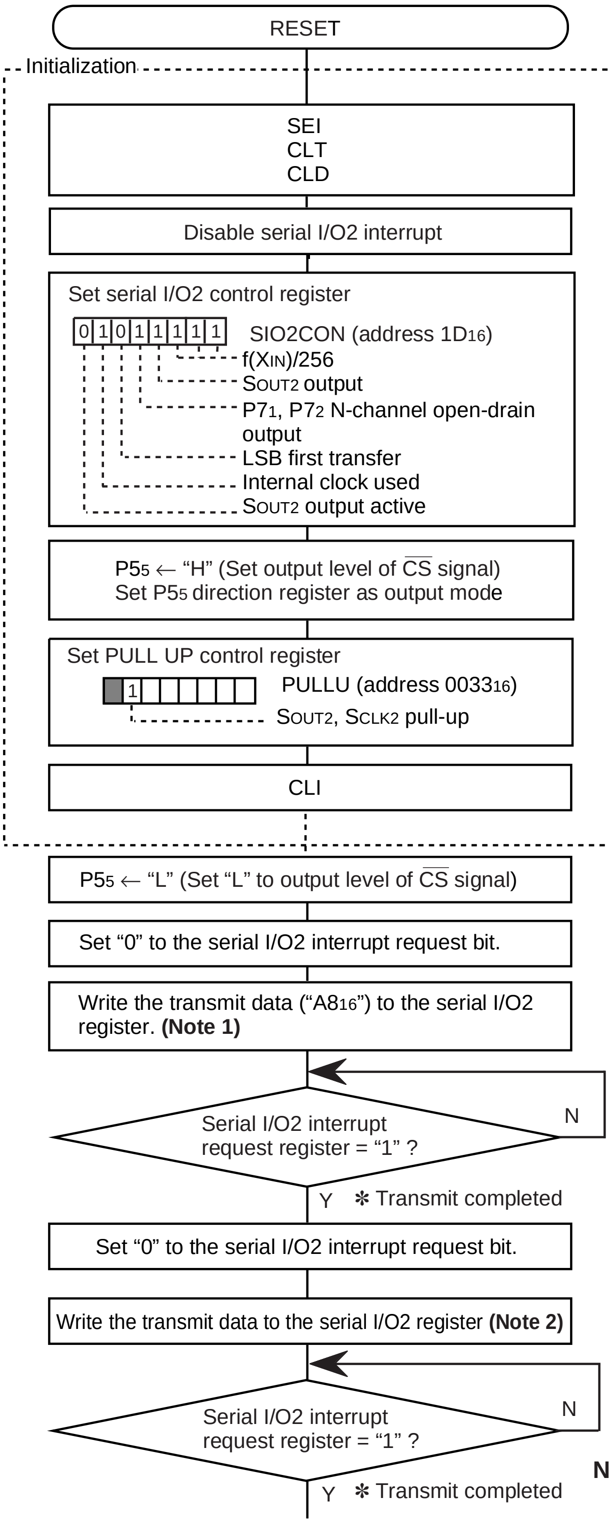
- •Synchronous clock: 25 kHz (dividing f(X_{IN}) = 6.4 MHz by 256)
- •Transfer direction: LSB first
- •Serial I/O2 interrupt is not used.
- •Transmit control by connecting port $P5_5$ and $\overline{CS}$ pin (“L” level active) of peripheral IC
- •M6M80011AP (Mitsubishi) is used as E²PROM

Figure 2.5.33 shows the timing in read mode.
Figures 2.5.34 and 2.5.35 show the control procedure example.
For the peripheral circuit example, refer to **“Figure 2.5.29”**.



DATA1 : Read mode ($A8_{16}$)
DATA2 : Address of reading data
DATA3 : Receive data
DATA4 : Receive data

Fig. 2.5.33 Timing in read mode


RESET
Initialization
SEI
CLT
CLD
Disable serial I/O2 interrupt
Set serial I/O2 control register
0 1 0 1 1 1 1 1
SIO2CON (address 1D16)
f(XIN)/256
SOUT2 output
P71, P72 N-channel open-drain output
LSB first transfer
Internal clock used
SOUT2 output active
P55 ← "H" (Set output level of CS signal)
Set P55 direction register as output mode
Set PULL UP control register
1
PULLU (address 003316)
SOUT2, SCLK2 pull-up
CLI
P55 ← "L" (Set "L" to output level of CS signal)
Set "0" to the serial I/O2 interrupt request bit.
Write the transmit data ("A816") to the serial I/O2 register. (Note 1)
Serial I/O2 interrupt request register = "1" ?
N
Y
✻ Transmit completed
Set "0" to the serial I/O2 interrupt request bit.
Write the transmit data to the serial I/O2 register (Note 2)
Serial I/O2 interrupt request register = "1" ?
N
Y
✻ Transmit completed

Notes 1: Transmit of read mode
2: Transmit of transmit data's address

Continue to Figure 2.5.35

Fig. 2.5.34 Control procedure example (1)

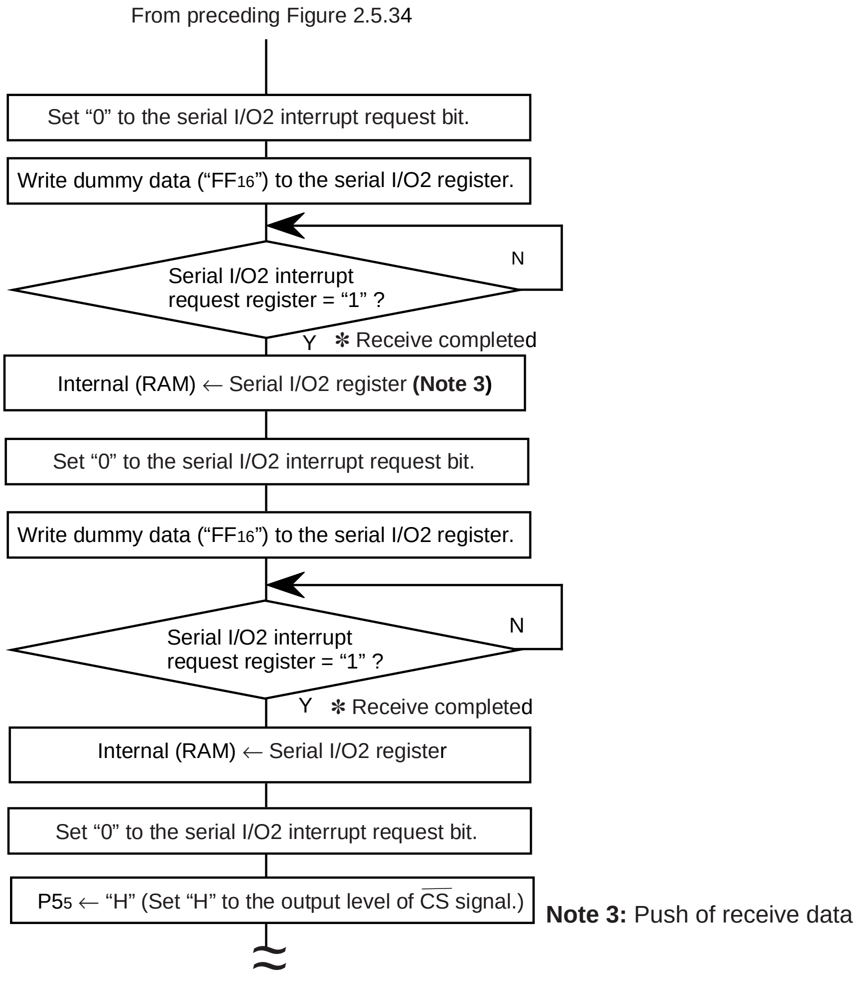

From preceding Figure 2.5.34
Set “0” to the serial I/O2 interrupt request bit.
Write dummy data (“FF16”) to the serial I/O2 register.
Serial I/O2 interrupt request register = “1” ?
N
Y
✻ Receive completed
Internal (RAM) ← Serial I/O2 register (Note 3)
Set “0” to the serial I/O2 interrupt request bit.
Write dummy data (“FF16”) to the serial I/O2 register.
Serial I/O2 interrupt request register = “1” ?
N
Y
✻ Receive completed
Internal (RAM) ← Serial I/O2 register
Set “0” to the serial I/O2 interrupt request bit.
P55 ← “H” (Set “H” to the output level of CS signal.)
Note 3: Push of receive data

Fig. 2.5.35 Control procedure example (2)

2.5.9 Notes on serial I/O2

(1) When using external clock

- The serial I/O2 interrupt request bit is set to “1” by counting eight times of the transfer clock when the synchronous clock is the internal clock or the external clock. However, when using the external clock, the contents of the serial I/O2 register is being shifted continuously while the transfer clock is input to the serial I/O2’s circuit. Stop the transfer clock at 8 times. (When using the internal clock, the transfer clock automatically stops.)

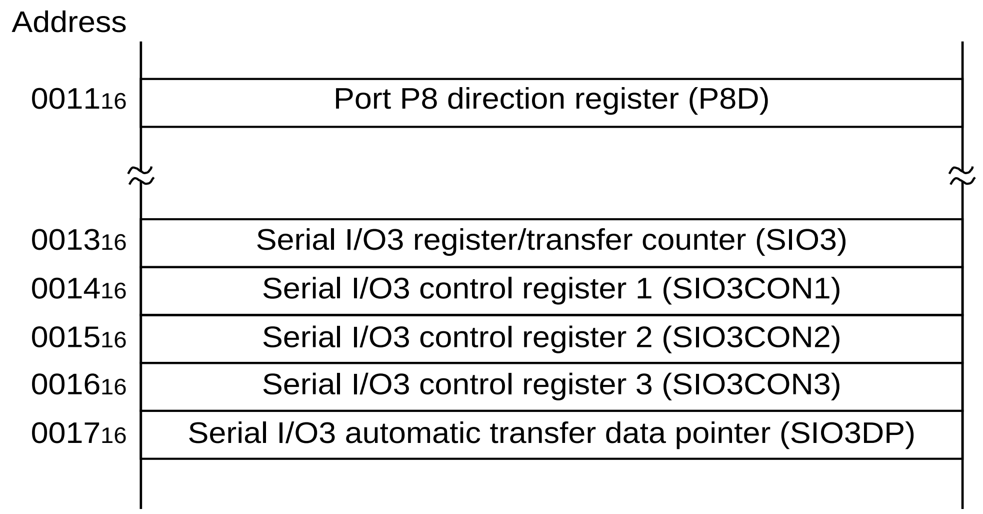
- When using the external clock, the S_{OUT2} pin does not become the high-impedance state after the data transfer is completed. Set “1” to the S_{OUT2} output control bit of the serial I/O2 control register after the data transfer is completed.
  When using the internal clock, the S_{OUT2} pin automatically becomes the high-impedance state after the data transfer is completed.

- When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLK2 input level. Also, write data to the serial I/O2 register at “H” of the SCLK2 input level.

2.5.10 Serial I/O3 memory assignment

Serial I/O3 has the following modes: 8-bit serial I/O, arbitrary bit from 1 to 256 bytes serial I/O, automatic transfer up to 256 bytes serial I/O.

Figure 2.5.36 shows the memory assignment of the serial I/O3 relevant registers.



Address	
0011_{16}	Port P8 direction register (P8D)
0013_{16}	Serial I/O3 register/transfer counter (SIO3)
0014_{16}	Serial I/O3 control register 1 (SIO3CON1)
0015_{16}	Serial I/O3 control register 2 (SIO3CON2)
0016_{16}	Serial I/O3 control register 3 (SIO3CON3)
0017_{16}	Serial I/O3 automatic transfer data pointer (SIO3DP)

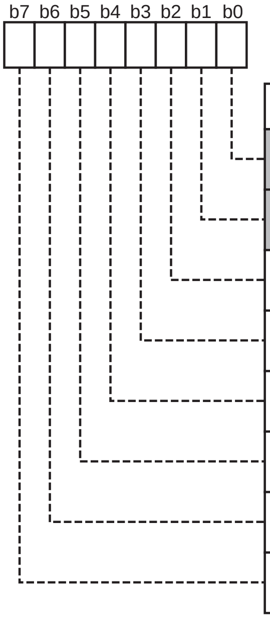
Fig. 2.5.36 Memory assignment of serial I/O3 relevant registers

2.5.11 Serial I/O3 relevant registers

(1) Port P8 direction register

Figure 2.5.37 shows the structure of the port P8 direction register.

Port P8 direction register



Port P8 direction register
(P8D: address 11_{16})

b	Name	Functions	At reset	R	W
0	Port P8 direction register	0 : Port $P8_0$ input mode 1 : Port $P8_0$ output mode	0	×	○
1		0 : Port $P8_1$ input mode 1 : Port $P8_1$ output mode	0	×	○
2		0 : Port $P8_2$ input mode 1 : Port $P8_2$ output mode	0	×	○
3		0 : Port $P8_3$ input mode 1 : Port $P8_3$ output mode	0	×	○
4		0 : Port $P8_4$ input mode 1 : Port $P8_4$ output mode	0	×	○
5		0 : Port $P8_5$ input mode 1 : Port $P8_5$ output mode	0	×	○
6		0 : Port $P8_6$ input mode 1 : Port $P8_6$ output mode	0	×	○
7		0 : Port $P8_7$ input mode 1 : Port $P8_7$ output mode	0	×	○

Note : The value of the port P8 direction register can be read out.

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	$P8_7$	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
	SSTB3	SBUSY3	SRDY3	SCLK3	SIN3	SOUT3		DA

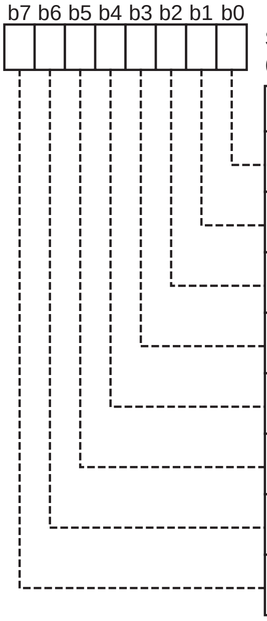
Fig. 2.5.37 Structure of port P8 direction register

(2) Serial I/O3 register/transfer counter

When the 8-bit serial I/O mode is selected by the serial I/O3 control register 1, this register becomes a buffer register for the data transmit/receive. When selecting the automatic transfer serial I/O mode or the arbitrary bits serial I/O mode, this register becomes the transfer counter.

Figure 2.5.38 shows the structure of the serial I/O3 register/transfer counter.

Serial I/O3 register/transfer counter



Serial I/O3 register/transfer counter
(SIO3: address 13_{16})

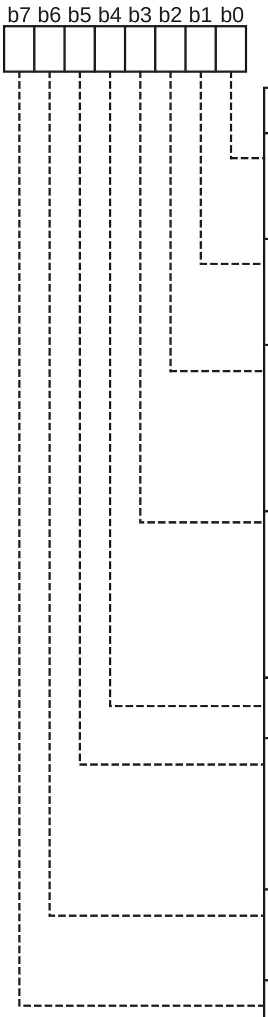
b	Functions	At reset	R	W
0	● When selecting the 8-bit serial I/O mode, this register becomes the transmit/receive buffer register. When selecting the internal synchronous clock, the transfer is started by writing the transmit data. ● When selecting the automatic transfer serial I/O mode, this register becomes the data transfer counter. Set (byte number – 1) to this register. ● When selecting the arbitrary bit serial I/O mode, this register becomes the data transfer counter. Set (bit number – 1) to this register. When the value is set to this register, the serial I/O communication is started at the same time.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 2.5.38 Structure of serial I/O3 register/transfer counter

(3) **Serial I/O3 control register 1**

This register controls various functions relevant to the serial I/O3. Figure 2.5.39 shows the structure of the serial I/O3 control register 1.

Serial I/O3 control register 1



Serial I/O3 control register 1
(SIO3CON1•SC31: address 14_{16})

b	Name	Functions	At reset	R	W
0	Serial transfer selection bits	00: Serial I/O disabled (P8$_2$ to P8$_7$ pins are I/O ports.) 01: 8-bit serial I/O 10: Arbitrary bit serial I/O 11: Automatic transfer serial I/O (8 bits)	0	○	○
1			0	○	○
2	Serial I/O3 synchronous clock selection bits (P8$_7$/S$_{STB3}$ pin control bits)	00: Internal synchronous clock (P8$_7$ pin is I/O port.) 01: External synchronous clock (P8$_7$ pin is I/O port.) 10: Internal synchronous clock (P8$_7$ pin is $\overline{\mathrm{S_{STB3}}}$ output.) 11: Internal synchronous clock (P8$_7$ pin is S$_{STB3}$ output.)	0	○	○
3			0	○	○
4	Serial I/O initialization bit	0: Serial I/O initialization 1: Serial I/O enabled	0	○	○
5	Transfer mode selection bit	0: Full-duplex (transmit/receive) mode (P8$_3$ pin is S$_{IN3}$ I/O.) 1: Transmit-only mode (P8$_3$ pin is I/O port.)	0	○	○
6	Serial I/O3 transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
7	Automatic transfer RAM transmit/receive address selection bit	0: Transmit/Receive address match 200_{16} to $2FF_{16}$ **(Note 1)** 1: Transmit address 200_{16} to $27F_{16}$ Receive address 280_{16} to $2FF_{16}$ **(Note 2)**	0	○	○

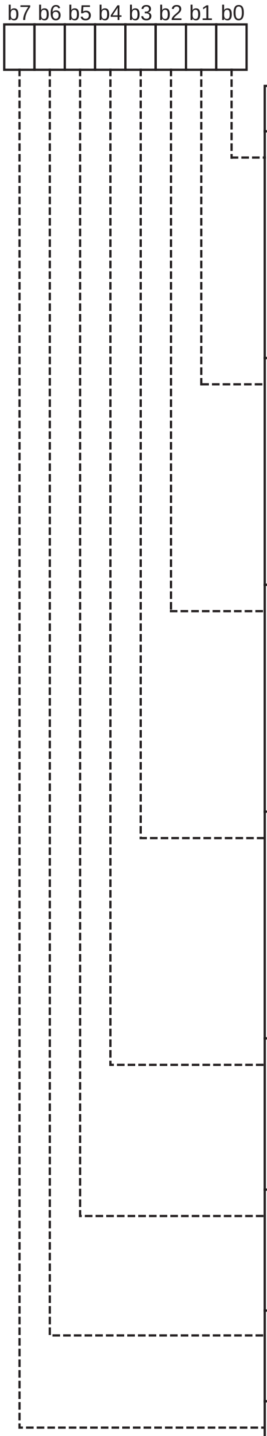
Notes 1: Set "00_{16}" to "FF_{16}" to the automatic transfer data pointer.
2: Set "00_{16}" to "$7F_{16}$" to the automatic transfer data pointer.

Fig. 2.5.39 Structure of serial I/O3 control register 1

(4) Serial I/O3 control register 2

This register controls various functions relevant to the serial I/O3. Figure 2.5.40 shows the structure of the serial I/O3 control register 2.

Serial I/O3 control register 2



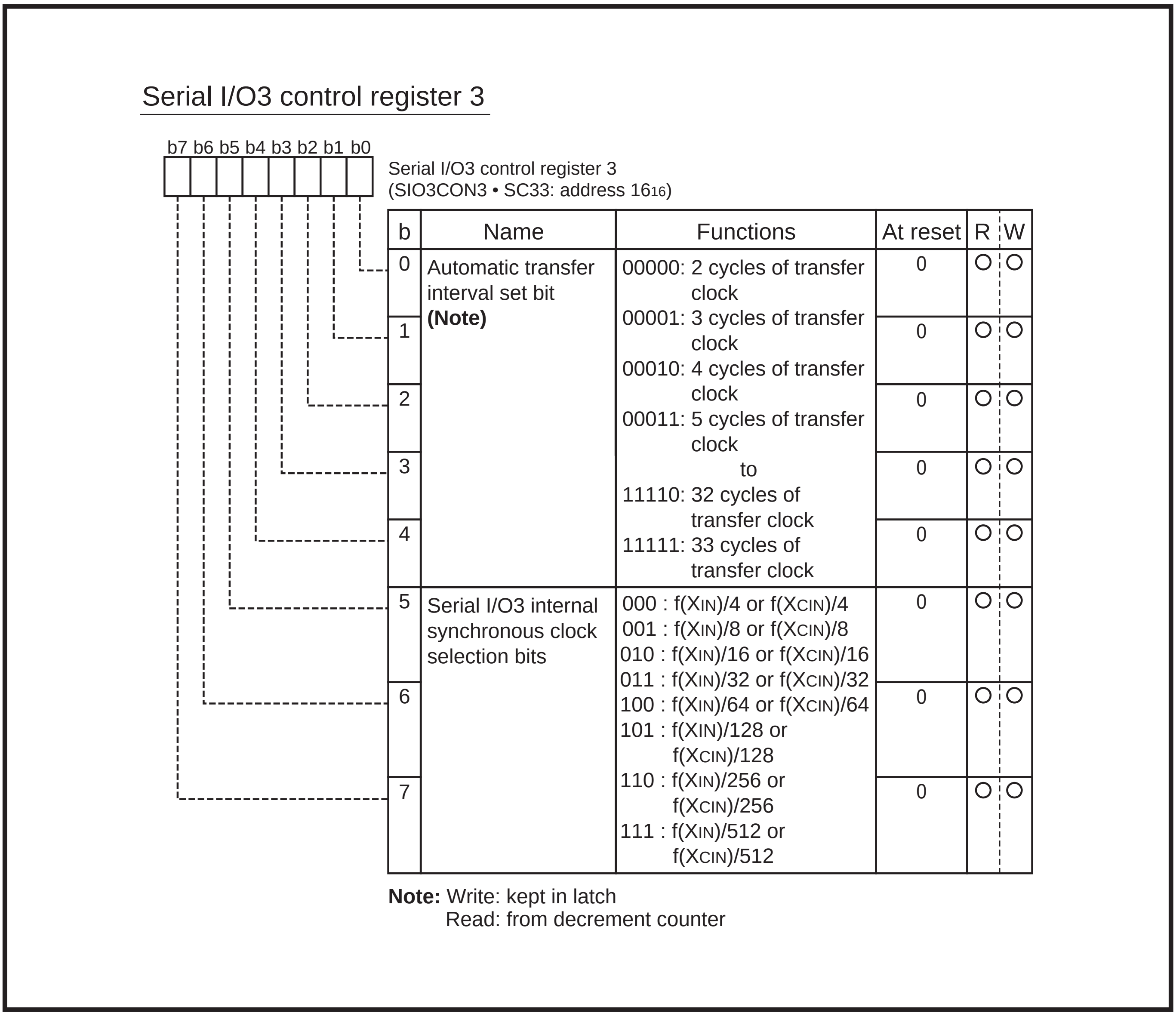
Serial I/O3 control register 2
(SIO3CON2 • SC32: address 15_{16})

b	Name	Functions	At reset	R	W
0	$P8_5$/SRDY3 • $P8_6$/SBUSY3 pin control bits	0000: $P8_5$, $P8_6$ pins are I/O ports. 0001: Not used 0010: $P8_5$ pin is $\overline{\text{SRDY3}}$ output; $P8_6$ pin is I/O port. 0011: $P8_5$ pin is SRDY3 output; $P8_6$ pin is I/O port. 0100: $P8_5$ pin is I/O port; $P8_6$ pin is $\overline{\text{SBUSY3}}$ input. 0101: $P8_5$ pin is I/O port; $P8_6$ pin is SBUSY3 input. 0110: $P8_5$ pin is I/O port; $P8_6$ pin is $\overline{\text{SBUSY3}}$ output. 0111: $P8_5$ pin is I/O port; $P8_6$ pin is SBUSY3 output. 1000: $P8_5$ pin is $\overline{\text{SRDY3}}$ input; $P8_6$ pin is $\overline{\text{SBUSY3}}$ output. 1001: $P8_5$ pin is $\overline{\text{SRDY3}}$ input; $P8_6$ pin is SBUSY3 output. 1010: $P8_5$ pin is SRDY3 input; $P8_6$ pin is $\overline{\text{SBUSY3}}$ output. 1011: $P8_5$ pin is SRDY3 input; $P8_6$ pin is SBUSY3 output. 1100: $P8_5$ pin is $\overline{\text{SRDY3}}$ output; $P8_6$ pin is $\overline{\text{SBUSY3}}$ input. 1101: $P8_5$ pin is $\overline{\text{SRDY3}}$ output; $P8_6$ pin is SBUSY3 input. 1110: $P8_5$ pin is SRDY3 output; $P8_6$ pin is $\overline{\text{SBUSY3}}$ input. 1111: $P8_5$ pin is SRDY3 output; $P8_6$ pin is SBUSY3 input.	0	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4	SBUSY3 output • SSTB3 output function selection bit (Valid in automatic transfer mode)	0: Functions as signal for each 1-byte 1: Functions as signal for each transfer data set	0	○	○
5	Serial transfer status flag	0: Serial transfer completed 1: Serial transfer in-progress	0	○	×
6	SOUT3 output control bit (when serial data is not transferred)	0: Output active 1: Output high-impedance	0	○	○
7	$P8_2$/SOUT3 • $P8_4$/SCLK3 P-channel output disable bit	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○

Fig. 2.5.40 Structure of serial I/O3 control register 2

(5) Serial I/O3 control register 3

This register controls various functions relevent to the serial I/O3. Figure 2.5.41 shows the structure of the serial I/O3 control register 3.



Serial I/O3 control register 3

Serial I/O3 control register 3
(SIO3CON3 • SC33: address 16$_{16}$)

b	Name	Functions	At reset	R	W
0	Automatic transfer interval set bit **(Note)**	00000: 2 cycles of transfer clock 00001: 3 cycles of transfer clock 00010: 4 cycles of transfer clock 00011: 5 cycles of transfer clock to 11110: 32 cycles of transfer clock 11111: 33 cycles of transfer clock	0	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5	Serial I/O3 internal synchronous clock selection bits	000 : f(XIN)/4 or f(XCIN)/4 001 : f(XIN)/8 or f(XCIN)/8 010 : f(XIN)/16 or f(XCIN)/16 011 : f(XIN)/32 or f(XCIN)/32 100 : f(XIN)/64 or f(XCIN)/64 101 : f(XIN)/128 or f(XCIN)/128 110 : f(XIN)/256 or f(XCIN)/256 111 : f(XIN)/512 or f(XCIN)/512	0	○	○
6			0	○	○
7			0	○	○

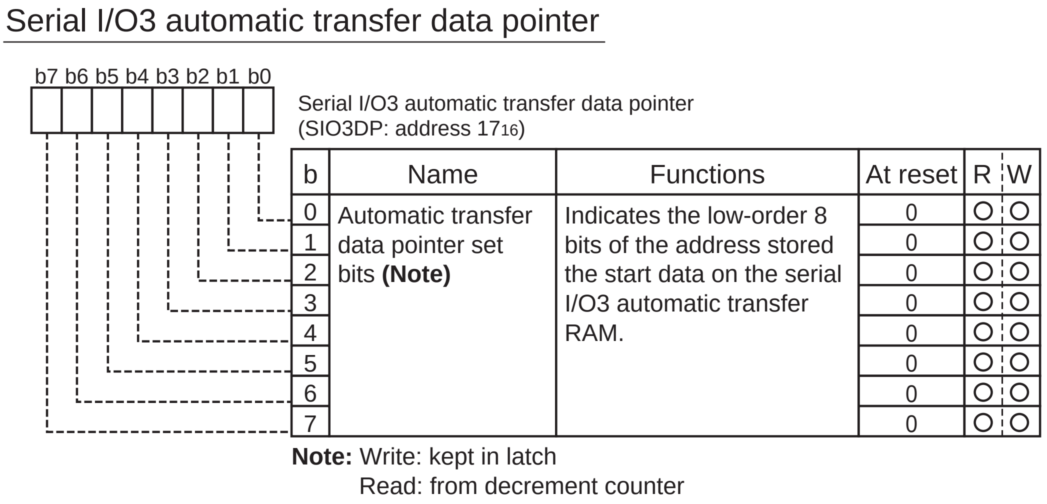
Note: Write: kept in latch
Read: from decrement counter

Fig. 2.5.41 Structure of serial I/O3 control register 3

(6) Serial I/O3 automatic transfer data pointer

When the automatic transfer serial I/O mode or the arbitrary bit serial I/O mode is selected, set the low-order 8 bits of the start address on the automatic transfer RAM, in which data is stored, to this register. When the automatic transfer RAM transmit/receive address selection bit of the serial I/O3 control register 1 is “0” (transmit and receive addresses match: 200_{16} to $2FF_{16}$), set “00_{16}” to “FF_{16}” to this register. When that bit is “1” (transmit addresses: 200_{16} to $27F_{16}$, receive addresses: 280_{16} to $2FF_{16}$), set “00_{16}” to “$7F_{16}$” to this register.

Figure 2.5.42 shows the structure of the serial I/O3 automatic transfer data pointer.



Serial I/O3 automatic transfer data pointer

Serial I/O3 automatic transfer data pointer
(SIO3DP: address 17_{16})

b	Name	Functions	At reset	R	W
0	Automatic transfer data pointer set bits **(Note)**	Indicates the low-order 8 bits of the address stored the start data on the serial I/O3 automatic transfer RAM.	0	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Note: Write: kept in latch
Read: from decrement counter

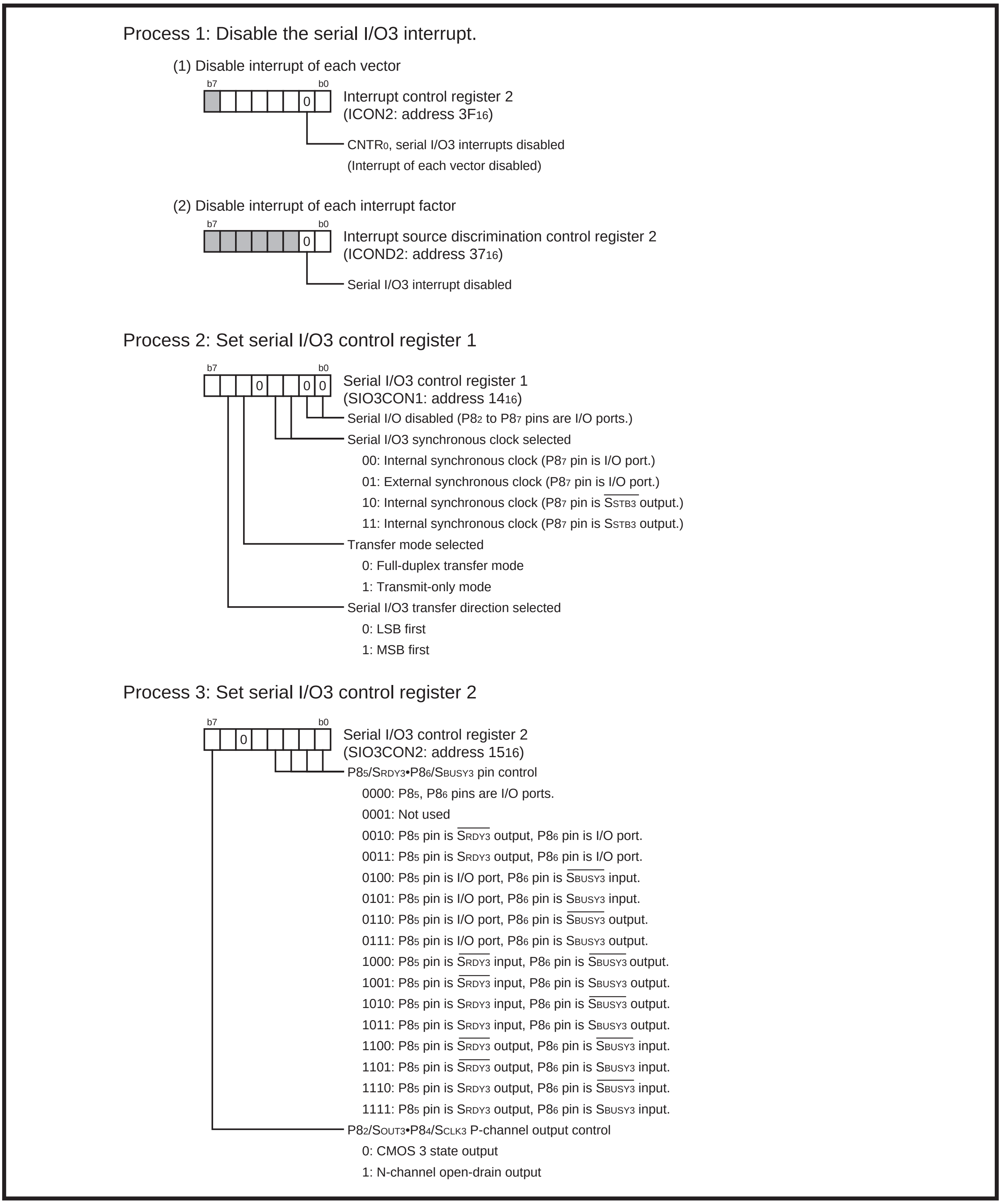
Fig. 2.5.42 Structure of serial I/O3 automatic transfer data pointer

2.5.12 8-bit serial I/O mode

In the 8-bit serial I/O mode, the serial I/O synchronous transmit/receive of 8-bit data is performed.

(1) 8-bit serial I/O mode setting method

Figures 2.5.43 and 2.5.44 show the 8-bit serial I/O mode setting method.



Process 1: Disable the serial I/O3 interrupt.

(1) Disable interrupt of each vector

Interrupt control register 2 (ICON2: address 3F$_{16}$)

CNTR$_0$, serial I/O3 interrupts disabled (Interrupt of each vector disabled)

(2) Disable interrupt of each interrupt factor

Interrupt source discrimination control register 2 (ICOND2: address 37$_{16}$)

Serial I/O3 interrupt disabled

Process 2: Set serial I/O3 control register 1

Serial I/O3 control register 1 (SIO3CON1: address 14$_{16}$)

Serial I/O disabled (P8$_2$ to P8$_7$ pins are I/O ports.)

Serial I/O3 synchronous clock selected
00: Internal synchronous clock (P8$_7$ pin is I/O port.)
01: External synchronous clock (P8$_7$ pin is I/O port.)
10: Internal synchronous clock (P8$_7$ pin is $\overline{\text{S}_{\text{STB3}}}$ output.)
11: Internal synchronous clock (P8$_7$ pin is S$_{\text{STB3}}$ output.)

Transfer mode selected
0: Full-duplex transfer mode
1: Transmit-only mode

Serial I/O3 transfer direction selected
0: LSB first
1: MSB first

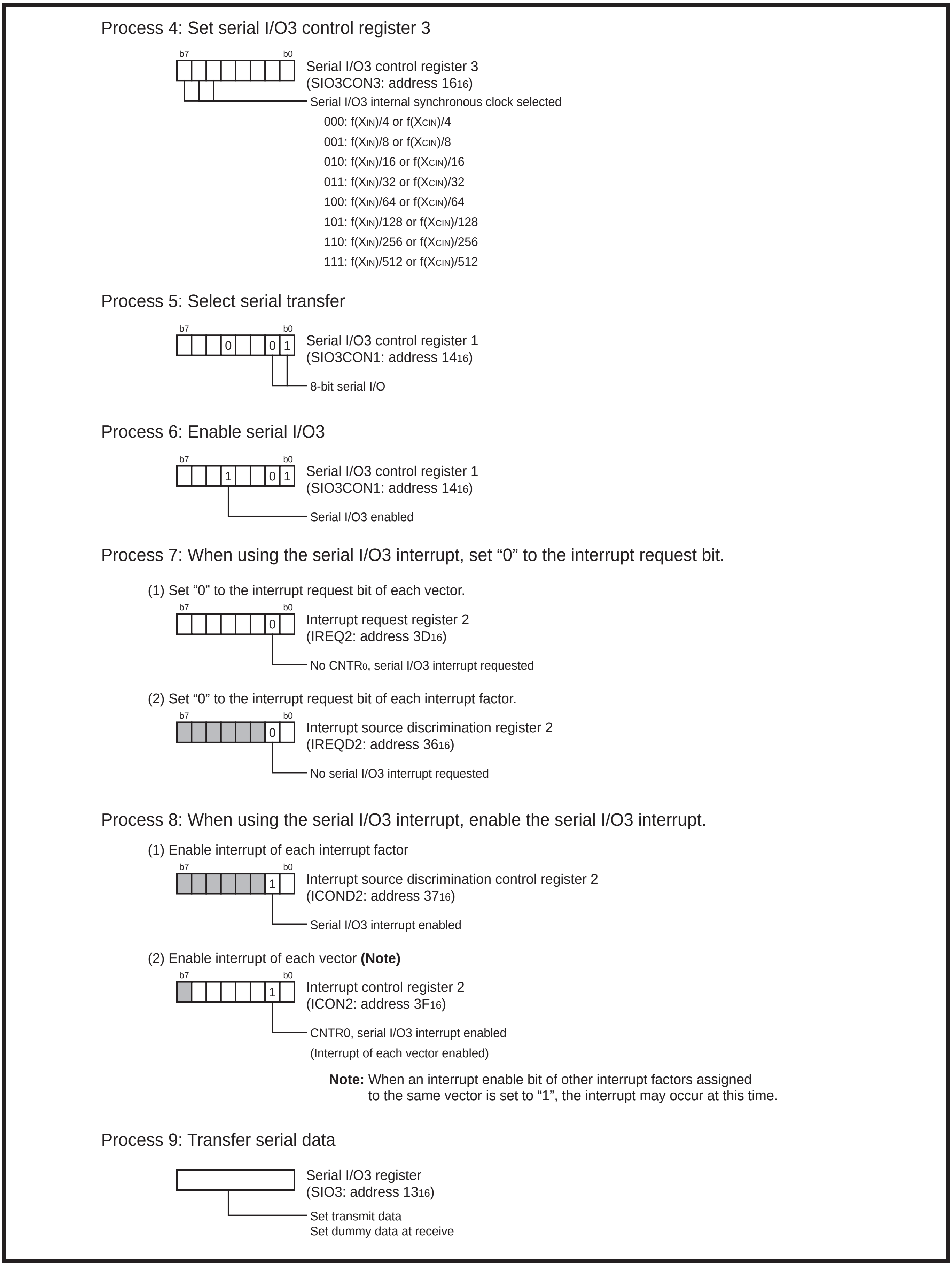
Process 3: Set serial I/O3 control register 2

Serial I/O3 control register 2 (SIO3CON2: address 15$_{16}$)

P8$_5$/S$_{\text{RDY3}}$•P8$_6$/S$_{\text{BUSY3}}$ pin control
0000: P8$_5$, P8$_6$ pins are I/O ports.
0001: Not used
0010: P8$_5$ pin is $\overline{\text{S}_{\text{RDY3}}}$ output, P8$_6$ pin is I/O port.
0011: P8$_5$ pin is S$_{\text{RDY3}}$ output, P8$_6$ pin is I/O port.
0100: P8$_5$ pin is I/O port, P8$_6$ pin is $\overline{\text{S}_{\text{BUSY3}}}$ input.
0101: P8$_5$ pin is I/O port, P8$_6$ pin is S$_{\text{BUSY3}}$ input.
0110: P8$_5$ pin is I/O port, P8$_6$ pin is $\overline{\text{S}_{\text{BUSY3}}}$ output.
0111: P8$_5$ pin is I/O port, P8$_6$ pin is S$_{\text{BUSY3}}$ output.
1000: P8$_5$ pin is $\overline{\text{S}_{\text{RDY3}}}$ input, P8$_6$ pin is $\overline{\text{S}_{\text{BUSY3}}}$ output.
1001: P8$_5$ pin is $\overline{\text{S}_{\text{RDY3}}}$ input, P8$_6$ pin is S$_{\text{BUSY3}}$ output.
1010: P8$_5$ pin is S$_{\text{RDY3}}$ input, P8$_6$ pin is $\overline{\text{S}_{\text{BUSY3}}}$ output.
1011: P8$_5$ pin is S$_{\text{RDY3}}$ input, P8$_6$ pin is S$_{\text{BUSY3}}$ output.
1100: P8$_5$ pin is $\overline{\text{S}_{\text{RDY3}}}$ output, P8$_6$ pin is $\overline{\text{S}_{\text{BUSY3}}}$ input.
1101: P8$_5$ pin is $\overline{\text{S}_{\text{RDY3}}}$ output, P8$_6$ pin is S$_{\text{BUSY3}}$ input.
1110: P8$_5$ pin is S$_{\text{RDY3}}$ output, P8$_6$ pin is $\overline{\text{S}_{\text{BUSY3}}}$ input.
1111: P8$_5$ pin is S$_{\text{RDY3}}$ output, P8$_6$ pin is S$_{\text{BUSY3}}$ input.

P8$_2$/S$_{\text{OUT3}}$•P8$_4$/S$_{\text{CLK3}}$ P-channel output control
0: CMOS 3 state output
1: N-channel open-drain output

Fig. 2.5.43 8-bit serial I/O mode setting method (1)



Process 4: Set serial I/O3 control register 3

b7 b0

Serial I/O3 control register 3
(SIO3CON3: address 16_{16})

Serial I/O3 internal synchronous clock selected

000: f(X_{IN})/4 or f(X_{CIN})/4
001: f(X_{IN})/8 or f(X_{CIN})/8
010: f(X_{IN})/16 or f(X_{CIN})/16
011: f(X_{IN})/32 or f(X_{CIN})/32
100: f(X_{IN})/64 or f(X_{CIN})/64
101: f(X_{IN})/128 or f(X_{CIN})/128
110: f(X_{IN})/256 or f(X_{CIN})/256
111: f(X_{IN})/512 or f(X_{CIN})/512

Process 5: Select serial transfer

b7 b0

0 0 1

Serial I/O3 control register 1
(SIO3CON1: address 14_{16})

8-bit serial I/O

Process 6: Enable serial I/O3

b7 b0

1 0 1

Serial I/O3 control register 1
(SIO3CON1: address 14_{16})

Serial I/O3 enabled

Process 7: When using the serial I/O3 interrupt, set “0” to the interrupt request bit.

(1) Set “0” to the interrupt request bit of each vector.

b7 b0

0

Interrupt request register 2
(IREQ2: address $3D_{16}$)

No CNTR$_0$, serial I/O3 interrupt requested

(2) Set “0” to the interrupt request bit of each interrupt factor.

b7 b0

0

Interrupt source discrimination register 2
(IREQD2: address 36_{16})

No serial I/O3 interrupt requested

Process 8: When using the serial I/O3 interrupt, enable the serial I/O3 interrupt.

(1) Enable interrupt of each interrupt factor

b7 b0

1

Interrupt source discrimination control register 2
(ICOND2: address 37_{16})

Serial I/O3 interrupt enabled

(2) Enable interrupt of each vector **(Note)**

b7 b0

1

Interrupt control register 2
(ICON2: address $3F_{16}$)

CNTR0, serial I/O3 interrupt enabled
(Interrupt of each vector enabled)

Note: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to “1”, the interrupt may occur at this time.

Process 9: Transfer serial data

Serial I/O3 register
(SIO3: address 13_{16})

Set transmit data
Set dummy data at receive

Fig. 2.5.44 8-bit serial I/O mode setting method (2)

(2) 8-bit serial I/O mode application examples

8-bit serial I/O mode application examples are the same as the clock synchronous serial I/O application examples. Refer to section “**2.5.8 (2) Clock synchronous serial I/O of serial I/O2 application examples**”.

2.5.13 Automatic transfer serial I/O mode

In the automatic transfer serial I/O mode, the serial transfer is performed through the serial I/O automatic transfer RAM (addresses 0200_{16} to $02FF_{16}$).

(1) Automatic transfer serial I/O mode setting method

Figures 2.5.45 to 2.5.47 show the automatic transfer serial I/O mode setting method.

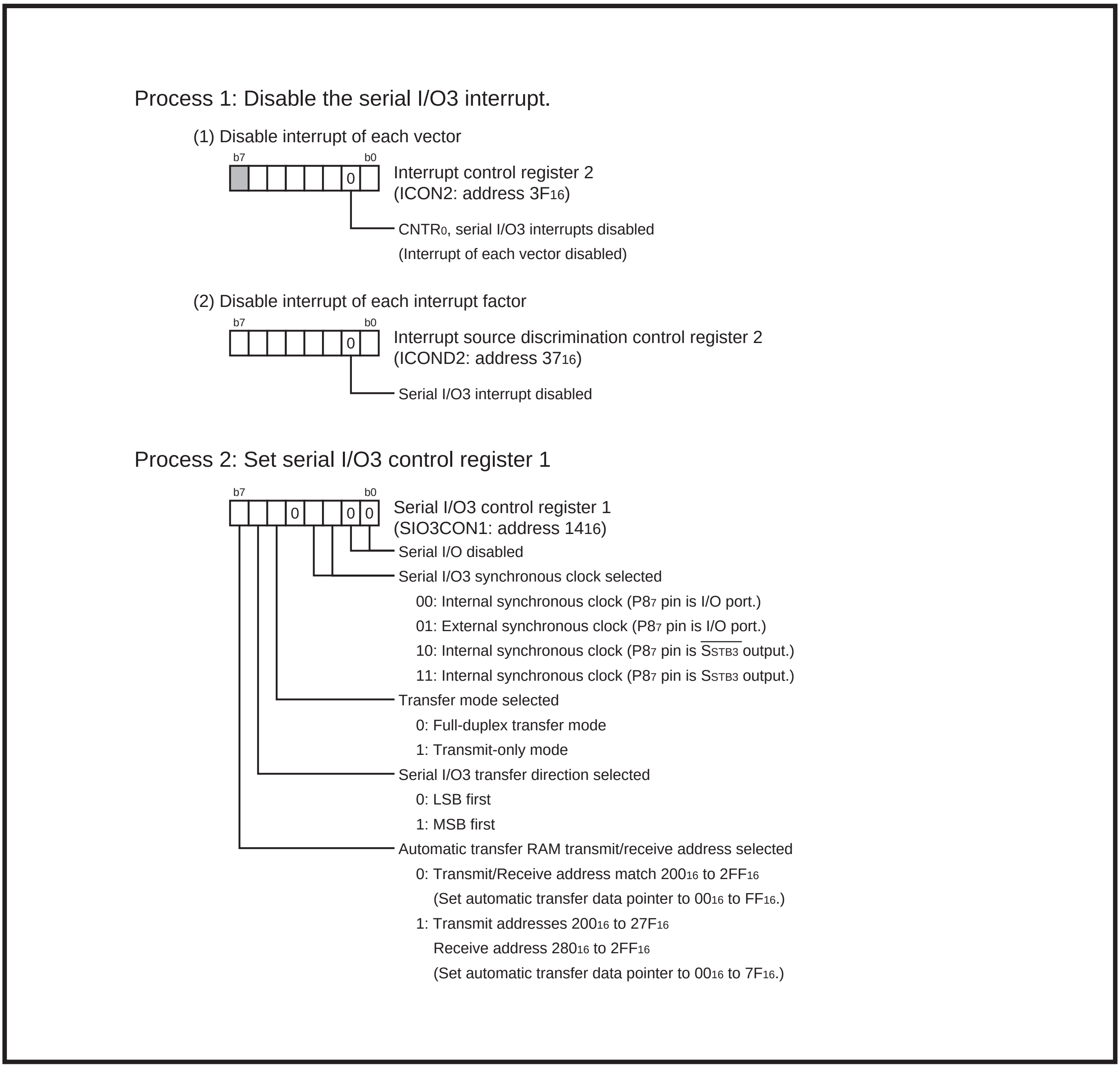
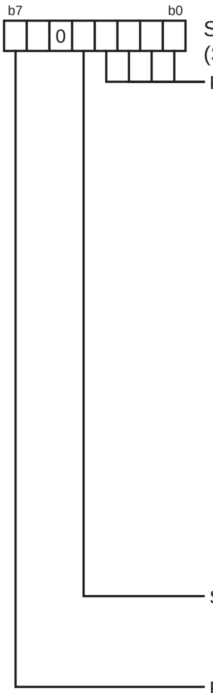

Process 1: Disable the serial I/O3 interrupt.
(1) Disable interrupt of each vector
b7
b0
0
Interrupt control register 2
(ICON2: address $3F_{16}$)
$CNTR_0$, serial I/O3 interrupts disabled
(Interrupt of each vector disabled)
(2) Disable interrupt of each interrupt factor
b7
b0
0
Interrupt source discrimination control register 2
(ICOND2: address 37_{16})
Serial I/O3 interrupt disabled
Process 2: Set serial I/O3 control register 1
b7
b0
0
0
0
Serial I/O3 control register 1
(SIO3CON1: address 14_{16})
Serial I/O disabled
Serial I/O3 synchronous clock selected
00: Internal synchronous clock ($P8_7$ pin is I/O port.)
01: External synchronous clock ($P8_7$ pin is I/O port.)
10: Internal synchronous clock ($P8_7$ pin is $\overline{S_{STB3}}$ output.)
11: Internal synchronous clock ($P8_7$ pin is S_{STB3} output.)
Transfer mode selected
0: Full-duplex transfer mode
1: Transmit-only mode
Serial I/O3 transfer direction selected
0: LSB first
1: MSB first
Automatic transfer RAM transmit/receive address selected
0: Transmit/Receive address match 200_{16} to $2FF_{16}$
(Set automatic transfer data pointer to 00_{16} to FF_{16}.)
1: Transmit addresses 200_{16} to $27F_{16}$
Receive address 280_{16} to $2FF_{16}$
(Set automatic transfer data pointer to 00_{16} to $7F_{16}$.)

Fig. 2.5.45 Automatic transfer serial I/O mode setting method (1)

Process 3: Set serial I/O3 control register 2



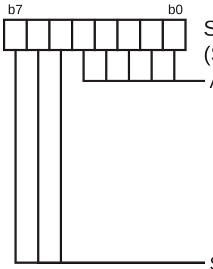
Serial I/O3 control register 2
(SIO3CON2: address 15_{16})

$P8_5/S_{RDY3}$•$P8_6/S_{BUSY3}$ pin control
0000: $P8_5$, $P8_6$ pins are I/O ports.
0001: Not used
0010: $P8_5$ pin is $\overline{S_{RDY3}}$ output, $P8_6$ pin is I/O port.
0011: $P8_5$ pin is S_{RDY3} output, $P8_6$ pin is I/O port.
0100: $P8_5$ pin is I/O port, $P8_6$ pin is $\overline{S_{BUSY3}}$ input.
0101: $P8_5$ pin is I/O port, $P8_6$ pin is S_{BUSY3} input.
0110: $P8_5$ pin is I/O port, $P8_6$ pin is $\overline{S_{BUSY3}}$ output.
0111: $P8_5$ pin is I/O port, $P8_6$ pin is S_{BUSY3} output.
1000: $P8_5$ pin is $\overline{S_{RDY3}}$ input, $P8_6$ pin is $\overline{S_{BUSY3}}$ output.
1001: $P8_5$ pin is $\overline{S_{RDY3}}$ input, $P8_6$ pin is S_{BUSY3} output.
1010: $P8_5$ pin is S_{RDY3} input, $P8_6$ pin is $\overline{S_{BUSY3}}$ output.
1011: $P8_5$ pin is S_{RDY3} input, $P8_6$ pin is S_{BUSY3} output.
1100: $P8_5$ pin is $\overline{S_{RDY3}}$ output, $P8_6$ pin is $\overline{S_{BUSY3}}$ input.
1101: $P8_5$ pin is $\overline{S_{RDY3}}$ output, $P8_6$ pin is S_{BUSY3} input.
1110: $P8_5$ pin is S_{RDY3} output, $P8_6$ pin is $\overline{S_{BUSY3}}$ input.
1111: $P8_5$ pin is S_{RDY3} output, $P8_6$ pin is S_{BUSY3} input.

S_{BUSY3} output•S_{STB3} output function selected
0: Functions as signal for each 1-byte
1: Functions as signal for each transfer data set

$P8_2/S_{OUT3}$•$P8_4/S_{CLK3}$ P-channel output control
0: CMOS 3 state output
1: N-channel open-drain output

Process 4: Set serial I/O3 control register 3

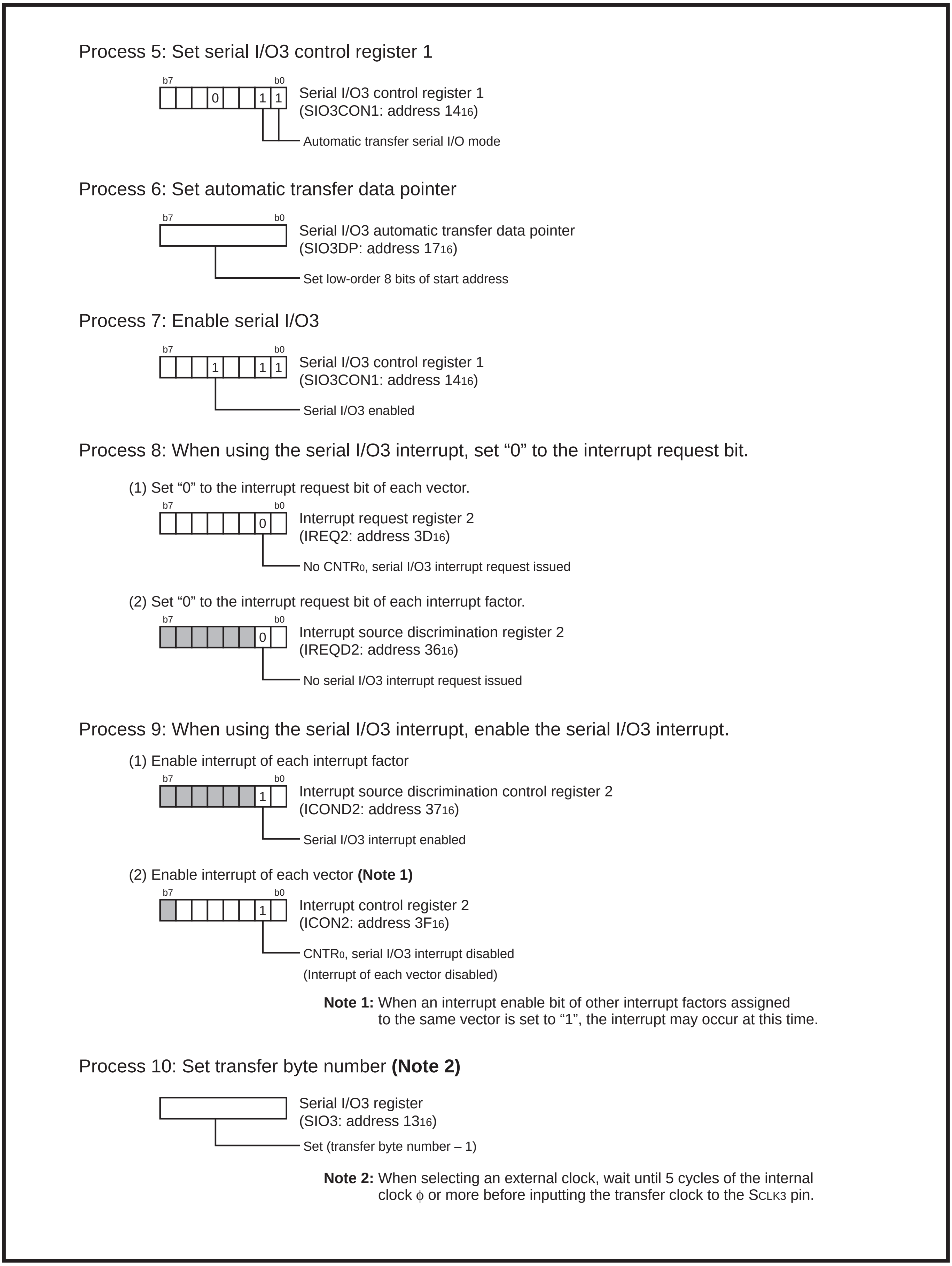


Serial I/O3 control register 3
(SIO3CON3: address 16_{16})

Automatic transfer interval set (when selecting internal clock)
00000: 2 cycles of transfer clock
00001: 3 cycles of transfer clock
to
11110: 32 cycles of transfer clock
11111: 33 cycles of transfer clock

Serial I/O3 internal synchronous clock selected
000: $f(X_{IN})/4$ or $f(X_{CIN})/4$
001: $f(X_{IN})/8$ or $f(X_{CIN})/8$
010: $f(X_{IN})/16$ or $f(X_{CIN})/16$
011: $f(X_{IN})/32$ or $f(X_{CIN})/32$
100: $f(X_{IN})/64$ or $f(X_{CIN})/64$
101: $f(X_{IN})/128$ or $f(X_{CIN})/128$
110: $f(X_{IN})/256$ or $f(X_{CIN})/256$
111: $f(X_{IN})/512$ or $f(X_{CIN})/512$

Fig. 2.5.46 Automatic transfer serial I/O mode setting method (2)



Process 5: Set serial I/O3 control register 1

b7
b0
0
1
1
Serial I/O3 control register 1
(SIO3CON1: address 14_{16})
Automatic transfer serial I/O mode

Process 6: Set automatic transfer data pointer

b7
b0
Serial I/O3 automatic transfer data pointer
(SIO3DP: address 17_{16})
Set low-order 8 bits of start address

Process 7: Enable serial I/O3

b7
b0
1
1
1
Serial I/O3 control register 1
(SIO3CON1: address 14_{16})
Serial I/O3 enabled

Process 8: When using the serial I/O3 interrupt, set “0” to the interrupt request bit.

(1) Set “0” to the interrupt request bit of each vector.

b7
b0
0
Interrupt request register 2
(IREQ2: address $3D_{16}$)
No CNTR$_0$, serial I/O3 interrupt request issued

(2) Set “0” to the interrupt request bit of each interrupt factor.

b7
b0
0
Interrupt source discrimination register 2
(IREQD2: address 36_{16})
No serial I/O3 interrupt request issued

Process 9: When using the serial I/O3 interrupt, enable the serial I/O3 interrupt.

(1) Enable interrupt of each interrupt factor

b7
b0
1
Interrupt source discrimination control register 2
(ICOND2: address 37_{16})
Serial I/O3 interrupt enabled

(2) Enable interrupt of each vector **(Note 1)**

b7
b0
1
Interrupt control register 2
(ICON2: address $3F_{16}$)
CNTR$_0$, serial I/O3 interrupt disabled
(Interrupt of each vector disabled)

Note 1: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to “1”, the interrupt may occur at this time.

Process 10: Set transfer byte number **(Note 2)**

Note 2: When selecting an external clock, wait until 5 cycles of the internal clock ϕ or more before inputting the transfer clock to the S_{CLK3} pin.

Fig. 2.5.47 Automatic transfer serial I/O mode setting method (3)

(2) Automatic transfer serial I/O mode application examples

■ **Outline**

The serial transmit/receive control is performed by using the automatic transfer serial I/O.

■ **Specifications**

- Transfer clock frequency: 131 kHz (f(X_{IN}) = 4.19 MHz/32)
- Transfer direction: LSB first
- Transmit/receive byte number: 8 bytes are transmitted as 1 block
- Automatic transfer RAM: addresses 200_{16} to 207_{16} used
- Transfer interval: 244 μs (32 bits of transfer clock)
- Serial I/O3 interrupt is not used.

Figure 2.5.48 shows the peripheral circuit example and Figure 2.5.49 shows the timing diagram. Figures 2.5.50 and 2.5.51 show the control procedure example.

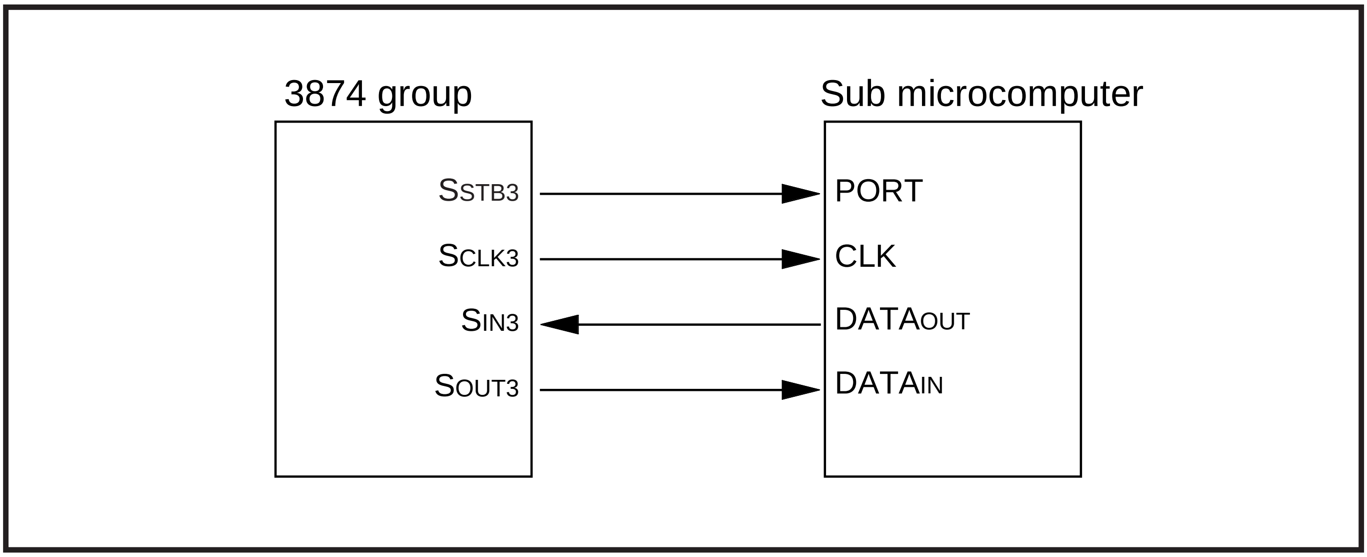


Fig. 2.5.48 Peripheral circuit example

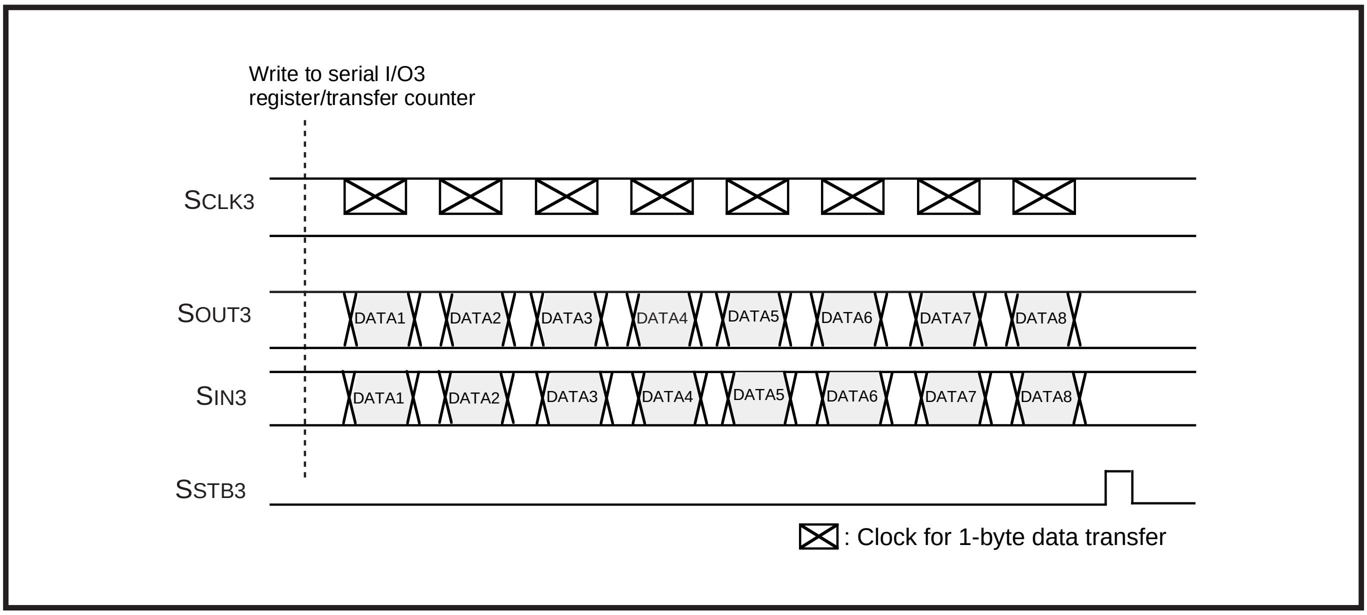

Write to serial I/O3 register/transfer counter
SCLK3
SOUT3
SIN3
SSTB3
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
DATA8
: Clock for 1-byte data transfer

Fig. 2.5.49 Timing diagram

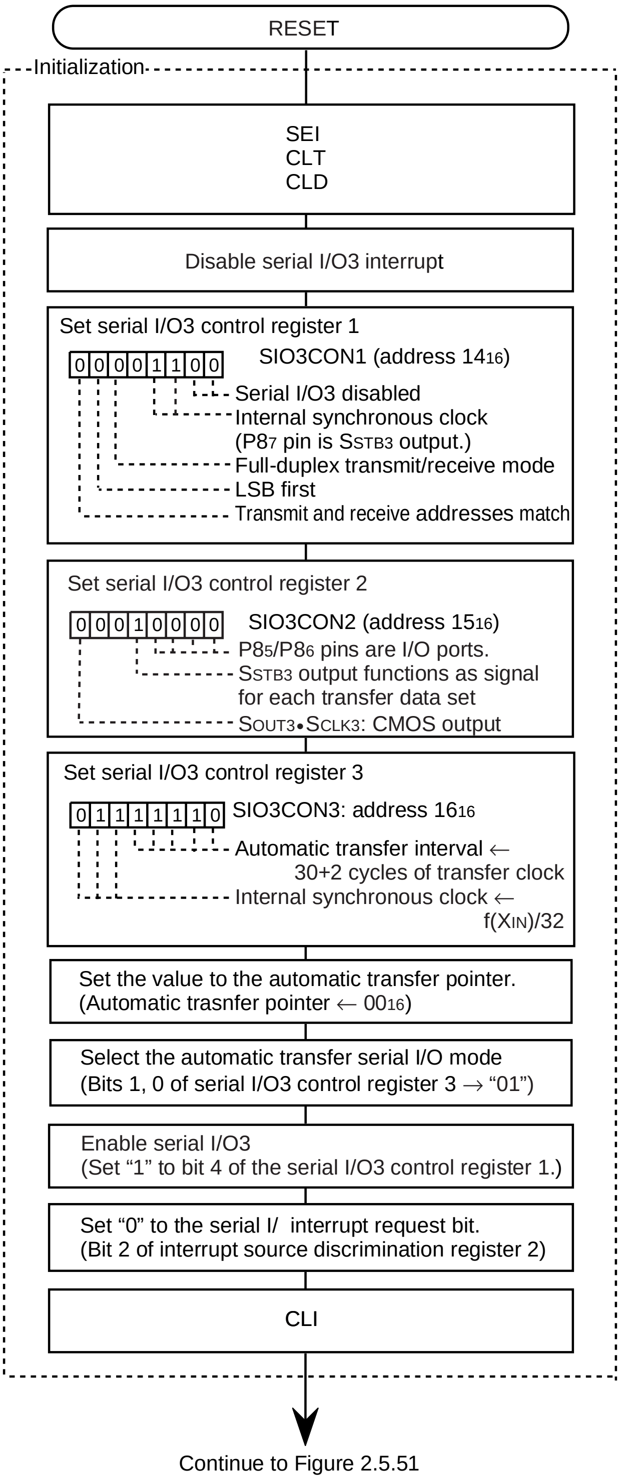

RESET
Initialization
SEI
CLT
CLD
Disable serial I/O3 interrupt
Set serial I/O3 control register 1
0 0 0 0 1 1 0 0
SIO3CON1 (address 14_{16})
Serial I/O3 disabled
Internal synchronous clock
(P8$_7$ pin is SSTB3 output.)
Full-duplex transmit/receive mode
LSB first
Transmit and receive addresses match
Set serial I/O3 control register 2
0 0 0 1 0 0 0 0
SIO3CON2 (address 15_{16})
P8$_5$/P8$_6$ pins are I/O ports.
SSTB3 output functions as signal
for each transfer data set
SOUT3•SCLK3: CMOS output
Set serial I/O3 control register 3
0 1 1 1 1 1 1 0
SIO3CON3: address 16_{16}
Automatic transfer interval ←
30+2 cycles of transfer clock
Internal synchronous clock ←
f(XIN)/32
Set the value to the automatic transfer pointer.
(Automatic trasnfer pointer ← 00_{16})
Select the automatic transfer serial I/O mode
(Bits 1, 0 of serial I/O3 control register 3 → “01”)
Enable serial I/O3
(Set “1” to bit 4 of the serial I/O3 control register 1.)
Set “0” to the serial I/ interrupt request bit.
(Bit 2 of interrupt source discrimination register 2)
CLI
Continue to Figure 2.5.51

Fig. 2.5.50 Control procedure example (1)

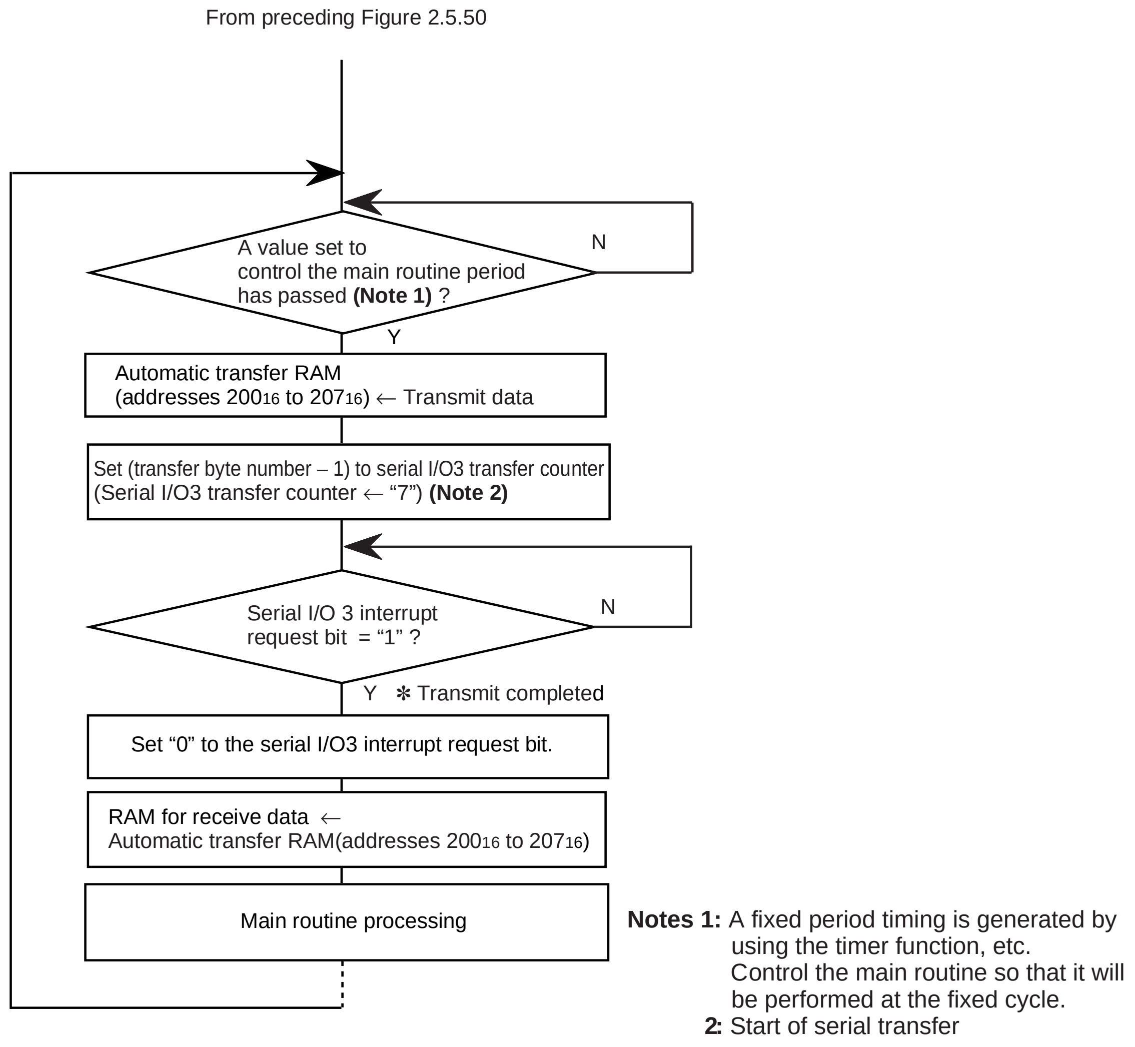

From preceding Figure 2.5.50
A value set to control the main routine period has passed (Note 1) ?
N
Y
Automatic transfer RAM (addresses 200₁₆ to 207₁₆) ← Transmit data
Set (transfer byte number – 1) to serial I/O3 transfer counter (Serial I/O3 transfer counter ← "7") (Note 2)
Serial I/O 3 interrupt request bit = "1" ?
N
Y ✻ Transmit completed
Set "0" to the serial I/O3 interrupt request bit.
RAM for receive data ← Automatic transfer RAM(addresses 200₁₆ to 207₁₆)
Main routine processing
Notes 1: A fixed period timing is generated by using the timer function, etc. Control the main routine so that it will be performed at the fixed cycle.
2: Start of serial transfer

Fig. 2.5.51 Control procedure example (2)

2.5.14 Arbitrary bit serial I/O mode

In the arbitrary bit serial I/O mode, the serial transfer of arbitrary bit number is performed through the serial I/O automatic transfer RAM (addresses 0200_{16} to $02FF_{16}$).

(1) Arbitrary bit serial I/O mode setting method

Figures 2.5.52 to 2.5.54 show the arbitrary bit serial I/O mode setting method.

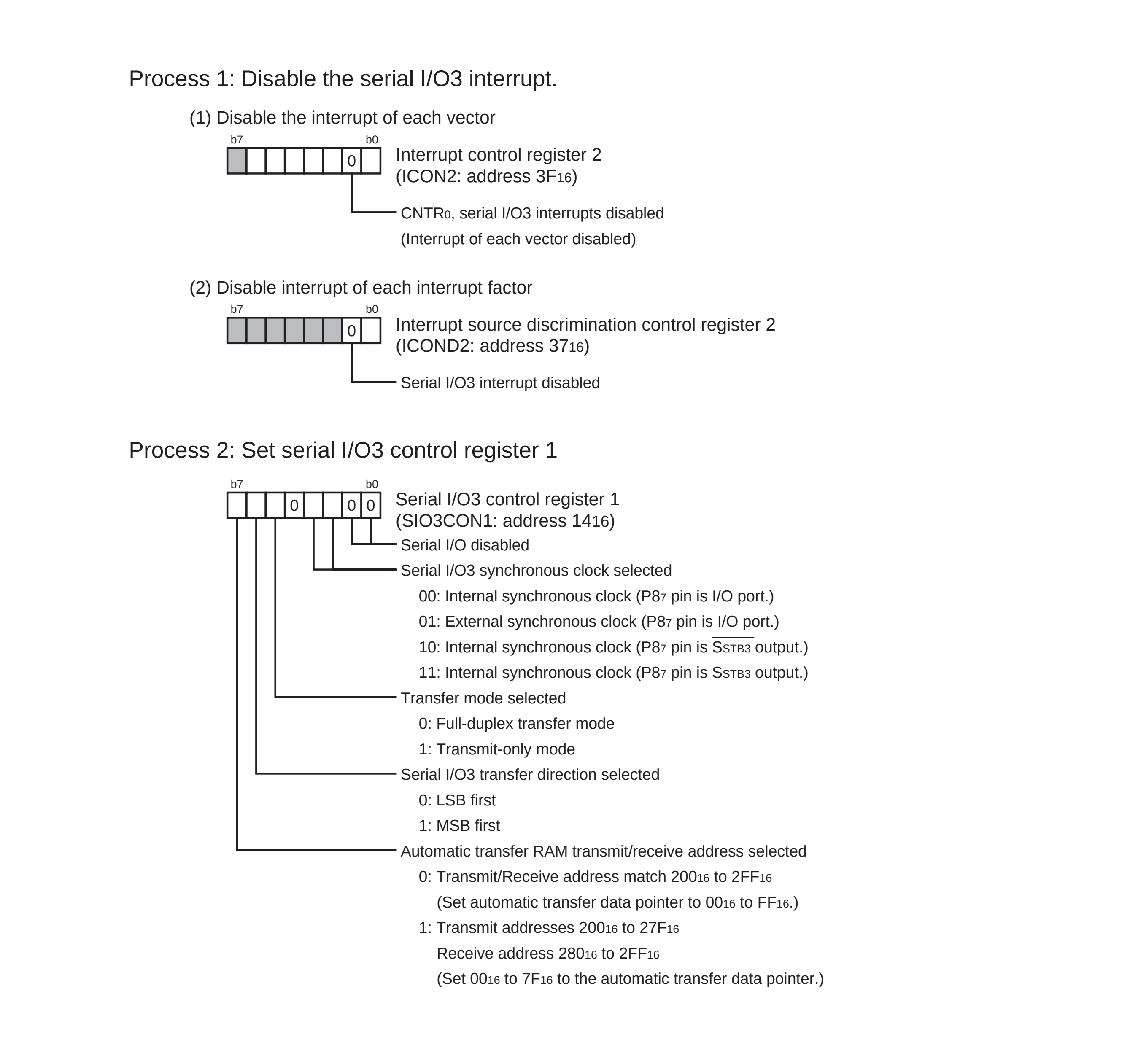
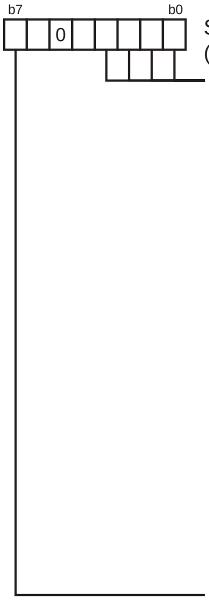

Process 1: Disable the serial I/O3 interrupt.
(1) Disable the interrupt of each vector
b7
b0
0
Interrupt control register 2
(ICON2: address $3F_{16}$)
CNTR$_0$, serial I/O3 interrupts disabled
(Interrupt of each vector disabled)
(2) Disable interrupt of each interrupt factor
b7
b0
0
Interrupt source discrimination control register 2
(ICOND2: address 37_{16})
Serial I/O3 interrupt disabled
Process 2: Set serial I/O3 control register 1
b7
b0
0
0
0
Serial I/O3 control register 1
(SIO3CON1: address 14_{16})
Serial I/O disabled
Serial I/O3 synchronous clock selected
00: Internal synchronous clock (P8$_7$ pin is I/O port.)
01: External synchronous clock (P8$_7$ pin is I/O port.)
10: Internal synchronous clock (P8$_7$ pin is $\overline{S_{STB3}}$ output.)
11: Internal synchronous clock (P8$_7$ pin is S_{STB3} output.)
Transfer mode selected
0: Full-duplex transfer mode
1: Transmit-only mode
Serial I/O3 transfer direction selected
0: LSB first
1: MSB first
Automatic transfer RAM transmit/receive address selected
0: Transmit/Receive address match 200_{16} to $2FF_{16}$
(Set automatic transfer data pointer to 00_{16} to FF_{16}.)
1: Transmit addresses 200_{16} to $27F_{16}$
Receive address 280_{16} to $2FF_{16}$
(Set 00_{16} to $7F_{16}$ to the automatic transfer data pointer.)

Fig. 2.5.52 Arbitrary bit serial I/O mode setting method (1)

Process 3: Set serial I/O3 control register 2



Serial I/O3 control register 2
(SIO3CON2: address 15_{16})

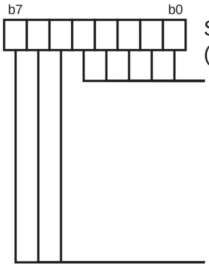
$P8_5/S_{RDY3}$•$P8_6/S_{BUSY3}$ pin control

0000: $P8_5$, $P8_6$ pins are I/O ports.
0001: Not used
0010: $P8_5$ pin is $\overline{S_{RDY3}}$ output, $P8_6$ pin is I/O port.
0011: $P8_5$ pin is S_{RDY3} output, $P8_6$ pin is I/O port.
0100: $P8_5$ pin is I/O port, $P8_6$ pin is $\overline{S_{BUSY3}}$ input.
0101: $P8_5$ pin is I/O port, $P8_6$ pin is S_{BUSY3} input.
0110: $P8_5$ pin is I/O port, $P8_6$ pin is $\overline{S_{BUSY3}}$ output.
0111: $P8_5$ pin is I/O port, $P8_6$ pin is S_{BUSY3} output.
1000: $P8_5$ pin is $\overline{S_{RDY3}}$ input, $P8_6$ pin is $\overline{S_{BUSY3}}$ output.
1001: $P8_5$ pin is $\overline{S_{RDY3}}$ input, $P8_6$ pin is S_{BUSY3} output.
1010: $P8_5$ pin is S_{RDY3} input, $P8_6$ pin is $\overline{S_{BUSY3}}$ output.
1011: $P8_5$ pin is S_{RDY3} input, $P8_6$ pin is S_{BUSY3} output.
1100: $P8_5$ pin is $\overline{S_{RDY3}}$ output, $P8_6$ pin is $\overline{S_{BUSY3}}$ input.
1101: $P8_5$ pin is $\overline{S_{RDY3}}$ output, $P8_6$ pin is S_{BUSY3} input.
1110: $P8_5$ pin is S_{RDY3} output, $P8_6$ pin is $\overline{S_{BUSY3}}$ input.
1111: $P8_5$ pin is S_{RDY3} output, $P8_6$ pin is S_{BUSY3} input.

$P8_2/S_{OUT3}$•$P8_4/S_{CLK3}$ P-channel output control

0: CMOS 3 state output
1: N-channel open-drain output

Process 4: Set serial I/O3 control register 3



Serial I/O3 control register 3
(SIO3CON3: address 16_{16})

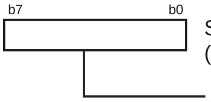
Automatic transfer interval set (when selecting internal clock)

00000: 2 cycles of transfer clock
00001: 3 cycles of transfer clock
to
11110: 32 cycles of transfer clock
11111: 33 cycles of transfer clock

Serial I/O3 internal synchronous clock selected

000: $f(X_{IN})/4$ or $f(X_{CIN})/4$
001: $f(X_{IN})/8$ or $f(X_{CIN})/8$
010: $f(X_{IN})/16$ or $f(X_{CIN})/16$
011: $f(X_{IN})/32$ or $f(X_{CIN})/32$
100: $f(X_{IN})/64$ or $f(X_{CIN})/64$
101: $f(X_{IN})/128$ or $f(X_{CIN})/128$
110: $f(X_{IN})/256$ or $f(X_{CIN})/256$
111: $f(X_{IN})/512$ or $f(X_{CIN})/512$

Process 5: Set serial I/O3 automatic transfer data pointer

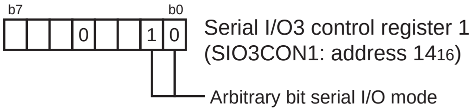


Serial I/O3 automatic transfer data pointer
(SIO3DP: address 17_{16})

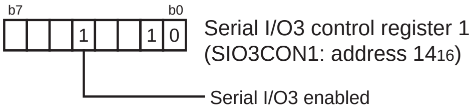
Set the low-order 8 bits of start address

Fig. 2.5.53 Arbitrary bit serial I/O mode setting method (2)

Process 6: Set serial I/O3 control register 1

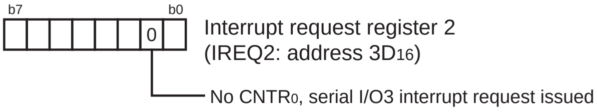

b7
b0
0
1
0
Serial I/O3 control register 1
(SIO3CON1: address 14_{16})
Arbitrary bit serial I/O mode

Process 7: Enable serial I/O3

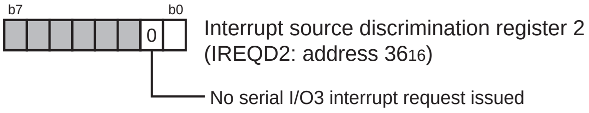


Process 8: When using the serial I/O3 interrupt, set "0" to the interrupt request bit.

(1) Set "0" to the interrupt request bit of each vector.

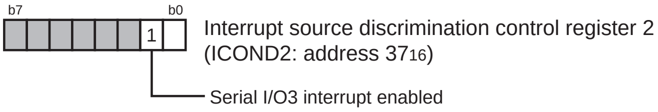

b7
b0
0
Interrupt request register 2
(IREQ2: address $3D_{16}$)
No CNTR$_0$, serial I/O3 interrupt request issued

(2) Set "0" to the interrupt request bit of each interrupt factor.

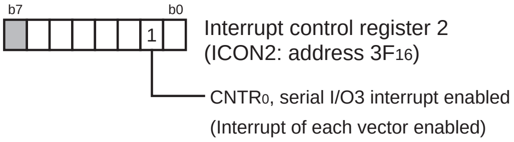

b7
b0
0
Interrupt source discrimination register 2
(IREQD2: address 36_{16})
No serial I/O3 interrupt request issued

Process 9: When using the serial I/O3 interrupt, enable the serial I/O3 interrupt.

(1) Enable interrupt ofeach interrupt factor

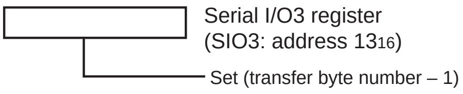

b7
b0
1
Interrupt source discrimination control register 2
(ICOND2: address 37_{16})
Serial I/O3 interrupt enabled

(2) Enable interrupt of each vector **(Note 1)**


b7
b0
1
Interrupt control register 2
(ICON2: address $3F_{16}$)
CNTR$_0$, serial I/O3 interrupt enabled
(Interrupt of each vector enabled)

Note 1: When an interrupt enable bit of other interrupt factors assigned to the same vector is set to "1", the interrupt may occur at this time.

Process 10: Set transfer bit number **(Note 2)**



Note 2: When selecting an external clock, wait until 5 cycles of the internal clock φ or more before, and then inputing the transfer clock to the SCLK3 pin.

Fig. 2.5.54 Arbitrary bit serial I/O mode setting method (3)

(2) Arbitrary bit serial I/O mode application examples

● **Outline**

The communication between the 3874 group and the external is performed by using a 24-bit I/O expander.

● **Specifications**

•Transfer clock frequency: 131 kHz (dividing f(X_{IN}) = 4.19 MHz by 32)
•Transfer direction: LSB first
•Transmit/receive byte number: 24 bits/block
•Automatic transfer RAM: addresses 0200_{16} to 0217_{16} used
•Transfer interval: 244 µs (32 bits of transfer clock)
•Serial I/O3 interrupt is not used.
•I/O expander: M66010 (Mitsubishi) used

Figure 2.5.55 shows the peripheral circuit example and Figure 2.5.56 shows the timing diagram. Figures 2.5.57 and 2.5.58 show the control procedure example.

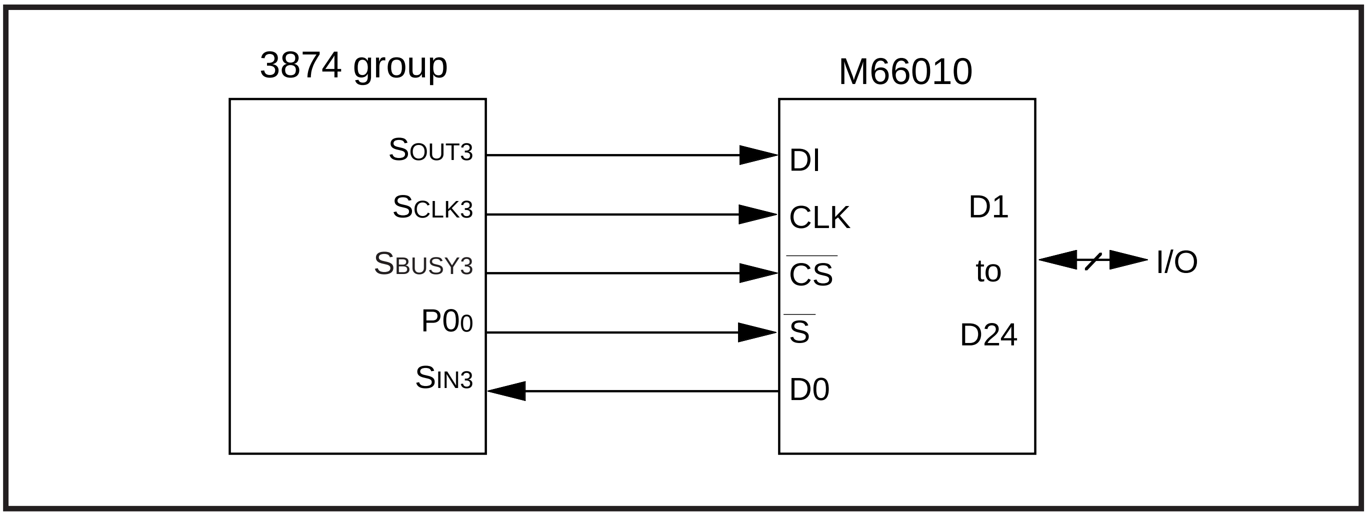


Fig. 2.5.55 Peripheral circuit example

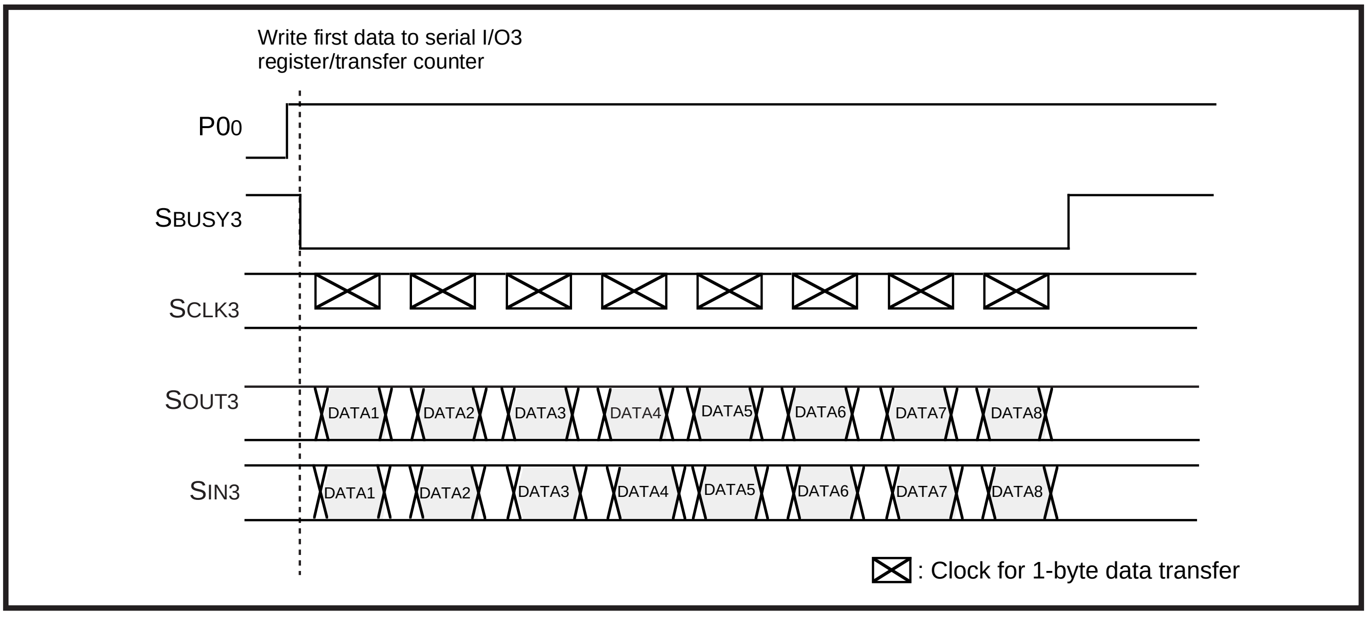

Write first data to serial I/O3 register/transfer counter
P00
SBUSY3
SCLK3
SOUT3
SIN3
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
DATA8
: Clock for 1-byte data transfer

Fig. 2.5.56 Timing diagram

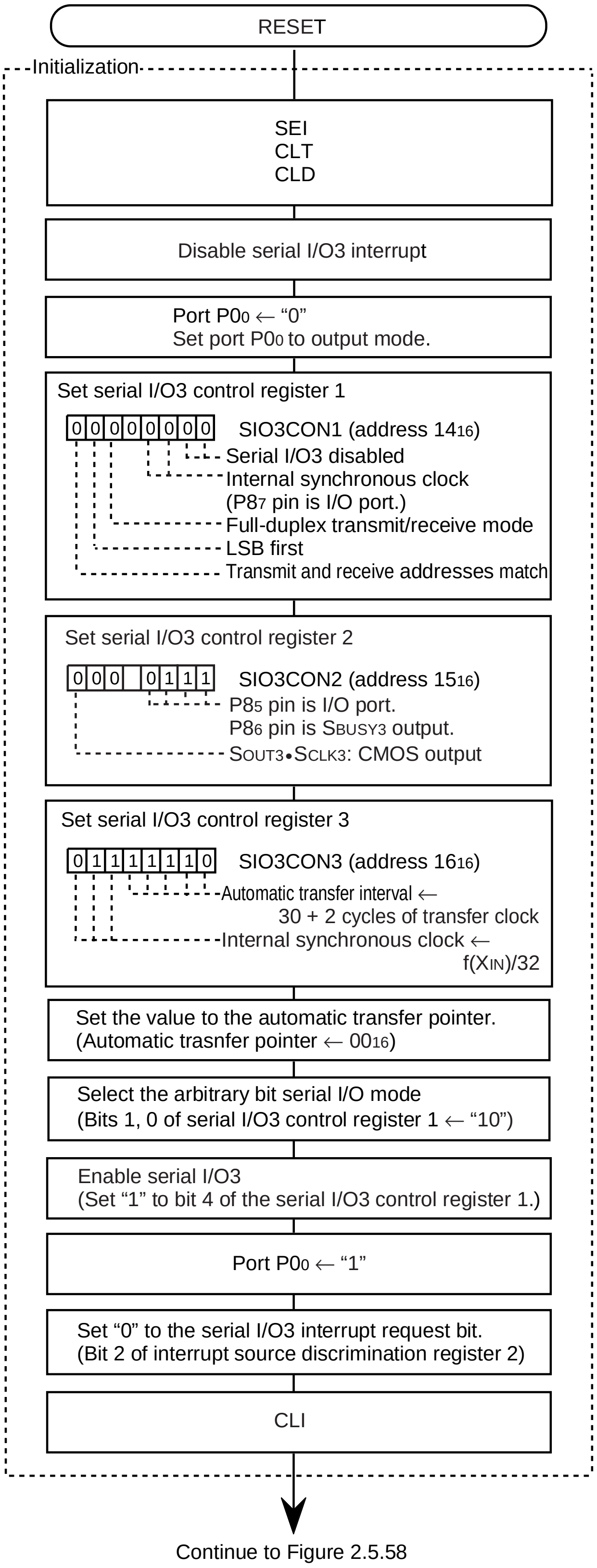
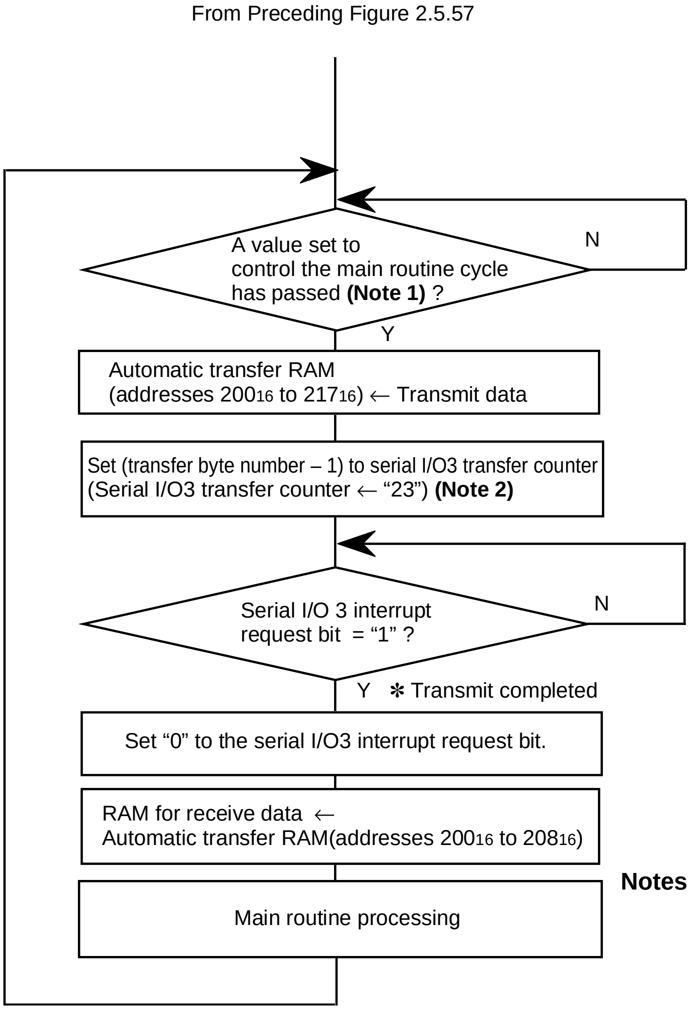

RESET
Initialization
SEI
CLT
CLD
Disable serial I/O3 interrupt
Port P00 ← “0”
Set port P00 to output mode.
Set serial I/O3 control register 1
0 0 0 0 0 0 0 0
SIO3CON1 (address 1416)
Serial I/O3 disabled
Internal synchronous clock
(P87 pin is I/O port.)
Full-duplex transmit/receive mode
LSB first
Transmit and receive addresses match
Set serial I/O3 control register 2
0 0 0 0 1 1 1
SIO3CON2 (address 1516)
P85 pin is I/O port.
P86 pin is SBUSY3 output.
SOUT3•SCLK3: CMOS output
Set serial I/O3 control register 3
0 1 1 1 1 1 1 0
SIO3CON3 (address 1616)
Automatic transfer interval ←
30 + 2 cycles of transfer clock
Internal synchronous clock ←
f(XIN)/32
Set the value to the automatic transfer pointer.
(Automatic trasnfer pointer ← 0016)
Select the arbitrary bit serial I/O mode
(Bits 1, 0 of serial I/O3 control register 1 ← “10”)
Enable serial I/O3
(Set “1” to bit 4 of the serial I/O3 control register 1.)
Port P00 ← “1”
Set “0” to the serial I/O3 interrupt request bit.
(Bit 2 of interrupt source discrimination register 2)
CLI
Continue to Figure 2.5.58

Fig. 2.5.57 Control procedure example (1)


From Preceding Figure 2.5.57
A value set to control the main routine cycle has passed (Note 1) ?
N
Y
Automatic transfer RAM (addresses 200₁₆ to 217₁₆) ← Transmit data
Set (transfer byte number – 1) to serial I/O3 transfer counter (Serial I/O3 transfer counter ← "23") (Note 2)
Serial I/O 3 interrupt request bit = "1" ?
N
Y ✻ Transmit completed
Set "0" to the serial I/O3 interrupt request bit.
RAM for receive data ← Automatic transfer RAM(addresses 200₁₆ to 208₁₆)
Main routine processing

Notes 1: A fixed period timing is generated by using the timer function, etc. Control the main routine so that it will be performed at the fixed cycle.
2: Start of serial transfer

Fig. 2.5.58 Control procedure example (2)

2.5.15 Notes on serial I/O3

(1) In all modes

■ **State of S_{OUT3} pin**

● The S_{OUT3} output control bit of the serial I/O3 control register 2 can be used to select the state of the S_{OUT3} pin when serial data is not transferred; either output active or high-impedance. However, when selecting an external synchronous clock; the S_{OUT3} pin can become the high-impedance state by setting the S_{OUT3} output control bit to “1” when S_{CLK3} input is at “H” after transfer completion.

■ **Serial I/O initialization bit**

● Set “0” to the serial I/O initialization bit of the serial I/O3 control register 1 when terminating a serial transfer during transferring.

● When writing “1” to the serial I/O initialization bit, serial I/O3 is enabled, but each register is not initialized. Set the value of each register by program.

■ **Handshake signal**

● S_{BUSY3} input signal

Input an “H” level to the S_{BUSY3} input and an “L” level signal to the $\overline{S_{BUSY3}}$ input in the initial state. When the external synchronous clock is selected, switch the input level to the S_{BUSY3} input and the $\overline{S_{BUSY3}}$ input while the S_{CLK3} input is in “H” state.

● S_{RDY3} input•output signal

When selecting the internal synchronous clock, input an “L” level to the S_{RDY3} input and an “H” level signal to the $\overline{S_{RDY3}}$ input in the initial state.

(2) 8-bit serial I/O mode

■ **When selecting external synchronous clock**

When an external synchronous clock is selected, the contents of the serial I/O3 register are being shifted continually while the transfer clock is input to S_{CLK3}. In this case, control the clock externally.

(3) In automatic transfer serial I/O mode

■ **Set of automatic transfer interval**

● When the S_{BUSY3} output is used, and the S_{BUSY3} output and the S_{STB3} output function as signal for each transfer data set by the S_{BUSY3} output•S_{STB3} output function selection bit; the transfer interval is necessary before the first data is transmitted/received, and after the last data is transmitted/received. Accordingly, regardless of the contents of the S_{BUSY3} output•S_{STB3} output function selection bit, this transfer interval becomes 2 cycles longer than the value set for each 1-byte data.

● When using the S_{STB3} output, regardless of the contents of the S_{BUSY3} output•S_{STB3} output function selection bit, this transfer interval becomes 2 cycles longer than the value set by the automatic transfer interval select bit of the serial I/O3 control register 3 for each 1-byte data.

● When using the combined output of S_{BUSY3} and S_{STB3} as the signal for each transfer data set, the transfer interval after completion of transmission/reception of the last data becomes 2 cycles longer than the value set by the automatic transfer interval select bit.

● Set the transfer interval of each 1-byte data transmission to 5 or more cycles of the internal clock ϕ after the rising edge of the last bit of a 1-byte data.

● When selecting an external clock, the automatic transfer interval cannot be set.

■ **Set of serial I/O3 transfer counter**

● Write the value decremented by 1 from the number of transfer data bytes to the serial I/O3 transfer counter.

● When selecting an external clock, after write a value to the serial I/O3 register/transfer counter, wait for 5 or more cycles of internal clock ϕ before inputting the transfer clock to the S_{CLK3} pin.

■ **Serial I/O initialization bit**

A serial I/O3 interrupt request occurs when “0” is written to the serial I/O initialization bit during an operation. Disable the interrupt enable bit as necessary by program.

(4) Arbitrary bit serial I/O mode

■ **Set of serial I/O3 transfer counter**

● Write the value decremented by 1 from the number of transfer data bits to the serial I/O3 register/transfer counter.

● When selecting an external clock, after write a value to the serial I/O3 register/transfer counter, wait for 5 or more cycles of internal clock ϕ before inputting the transfer clock to the S_{CLK3} pin.

■ **Set of automatic transfer interval**

● When selecting an external clock, the automatic interval cannot be set.

● When using the S_{BUSY3} output, the transfer interval is necessary before the first data is transmitted/received, and after the last data is transmitted/received. When using the S_{STB3} output, this transfer interval becomes 2 cycles longer than the value set for each 8-bit data. In addition, when using the combined output of S_{BUSY3} and S_{STB3}, the transfer interval after completion of transmission/reception of the last data becomes 2 cycles longer than the set value.

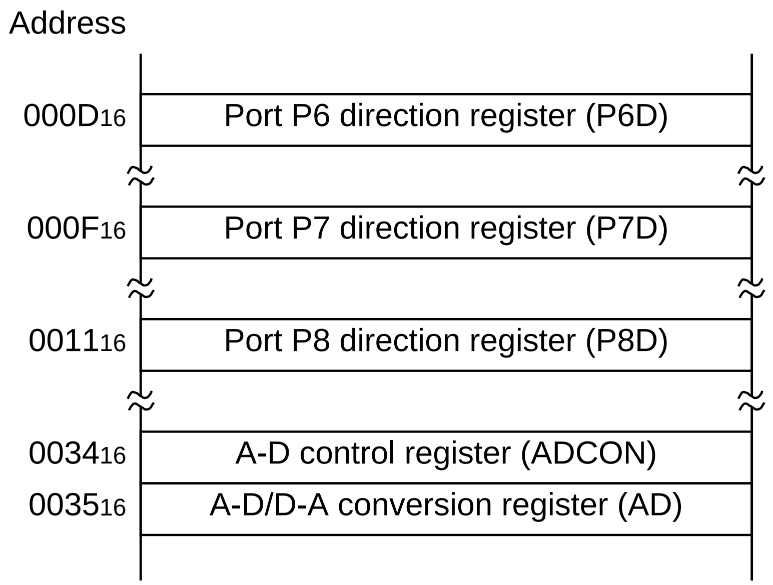
■ **Receive data**

If the last data does not fill 8 bits, the receive data stored in the serial I/O3 automatic transfer RAM becomes the closest MSB odd bit when the transfer direction select bit is set to LSB first, or the closest LSB odd bit when the transfer direction select bit is set to MSB first.

2.6 A-D converter and D-A converter

2.6.1 Memory assignment

Figure 2.6.1 shows the memory assignment of A-D converter and D-A converter relevant registers. Each of these registers is described below.



Address	
$000D_{16}$	Port P6 direction register (P6D)
$000F_{16}$	Port P7 direction register (P7D)
0011_{16}	Port P8 direction register (P8D)
0034_{16}	A-D control register (ADCON)
0035_{16}	A-D/D-A conversion register (AD)

Fig. 2.6.1 Memory assignment of A-D converter and D-A converter relevant registers

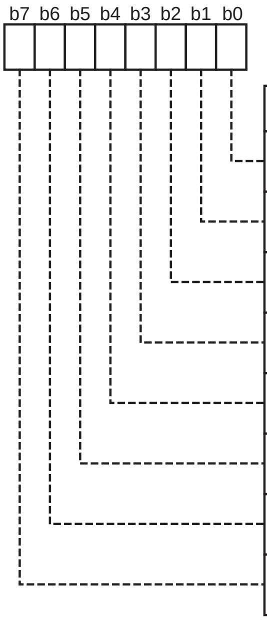
2.6.2 Relevant register

(1) Port Pi direction register (i = 6 to 8)

- To set port P6 direction register to the input mode, set “0” to bit corresponding to analog input pin to be used for A-D converter.
- When using the ADT pin input as the A-D converter start factor, set “0” to bit 7 of the port P7 direction register to the input mode.
- When using the D-A converter, Set “0” to bit 0 of the port P8 direction register to the input mode.

Figure 2.6.2 shows the structure of the port Pi direction register.

Port Pi direction register (i = 6 to 8)



Port Pi direction register
(PiD: addresses $0D_{16}$, $0F_{16}$, 11_{16})

b	Name	Functions	At reset	R	W
0	Port Pi direction register	0 : Port Pi$_0$ input mode 1 : Port Pi$_0$ output mode	0	×	○
1		0 : Port Pi$_1$ input mode 1 : Port Pi$_1$ output mode	0	×	○
2		0 : Port Pi$_2$ input mode 1 : Port Pi$_2$ output mode	0	×	○
3		0 : Port Pi$_3$ input mode 1 : Port Pi$_3$ output mode	0	×	○
4		0 : Port Pi$_4$ input mode 1 : Port Pi$_4$ output mode	0	×	○
5		0 : Port Pi$_5$ input mode 1 : Port Pi$_5$ output mode	0	×	○
6		0 : Port Pi$_6$ input mode 1 : Port Pi$_6$ output mode	0	×	○
7		0 : Port Pi$_7$ input mode 1 : Port Pi$_7$ output mode	0	×	○

● Port P6 direction register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	P6$_7$	P6$_6$	P6$_5$	P6$_4$	P6$_3$	P6$_2$	P6$_1$	P6$_0$
	AN$_7$	AN$_6$	AN$_5$	AN$_4$	AN$_3$	AN$_2$	AN$_1$	AN$_0$

● Port P7 direction register

Bit	b7
Name	P7$_7$
	ADT

● Port P8 direction register

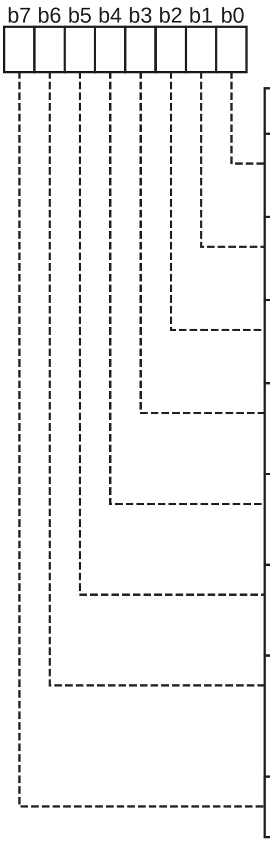
Bit	b0
Name	P8$_0$
	DA

Fig. 2.6.2 Structure of port Pi direction register (i = 6 to 8)

(2) A-D control register

The A-D control register controls A-D converter and D-A converter each. Figure 2.6.3 shows the structure of the A-D control register.

A-D control register



A-D control register
(ADCON: address 34_{16})

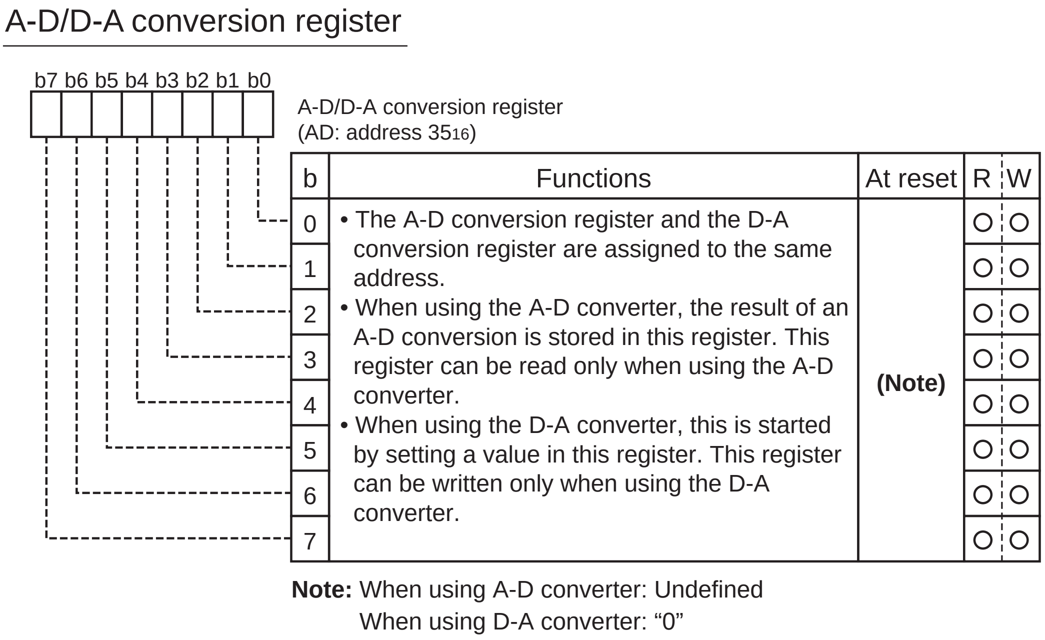
b	Name	Functions	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 000 : $P6_0/AN_0$ 001 : $P6_1/AN_1$ 010 : $P6_2/AN_2$ 011 : $P6_3/AN_3$ 100 : $P6_4/AN_4$ 101 : $P6_5/AN_5$ 110 : $P6_6/AN_6$ 111 : $P6_7/AN_7$	0	○	○
1			0	○	○
2			0	○	○
3	AD conversion completion bit **(Note)**	0: Conversion in progress 1: Conversion completed	1	○	○
4	V_{REF} input switch bit	0: OFF 1: ON	0	○	○
5	AD external trigger valid bit	0: AD external trigger invalid 1: AD external trigger valid	0	○	○
6	Interrupt source selection bit	0: Interrupt request at A-D conversion completed 1: Interrupt request at ADT input falling	0	○	○
7	DA output enable bit	0: DA output disabled 1: DA output enabled	0	○	○

Note: When setting “0” to bit 3, A-D converter is started.
Writing “0” to bit 3 is valid, but if writing “1” to bit 3, bit 3 is not set to “1”.
Accordingly, set “1” to bit 3 when a value is written to the A-D control register with bit 3 no affected.

Fig. 2.6.3 Structure of A-D control register

(3) A-D/D-A conversion register

When using the A-D converter, the result of an A-D conversion is stored in this register.
When using the D-A converter, The D-A conversion is started by setting a value in this register.
Figure 2.6.4 shows the structure of the A-D/D-A conversion register.



A-D/D-A conversion register

A-D/D-A conversion register
(AD: address 35_{16})

b	Functions	At reset	R	W
0	• The A-D conversion register and the D-A conversion register are assigned to the same address. • When using the A-D converter, the result of an A-D conversion is stored in this register. This register can be read only when using the A-D converter. • When using the D-A converter, this is started by setting a value in this register. This register can be written only when using the D-A converter.	**(Note)**	○	○
1			○	○
2			○	○
3			○	○
4			○	○
5			○	○
6			○	○
7			○	○

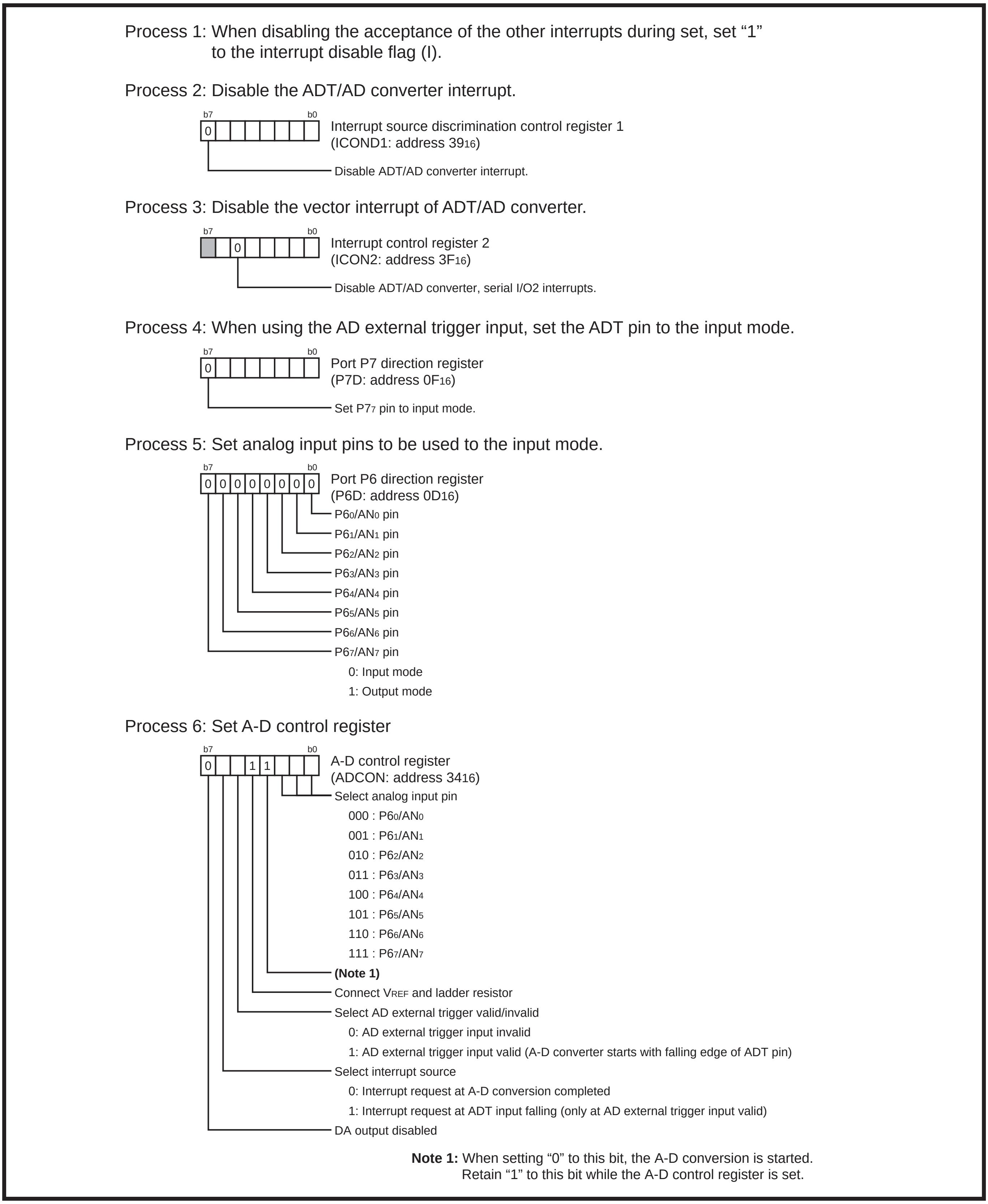
Note: When using A-D converter: Undefined
When using D-A converter: “0”

Fig. 2.6.4 Structure of A-D/D-A conversion register

2.6.3 A-D converter

(1) Setting method of A-D converter

Figures 2.6.5 and 2.6.6 show the setting method of A-D converter.



Process 1: When disabling the acceptance of the other interrupts during set, set “1” to the interrupt disable flag (I).

Process 2: Disable the ADT/AD converter interrupt.

b7
b0
0
Interrupt source discrimination control register 1
(ICOND1: address 39$_{16}$)
Disable ADT/AD converter interrupt.

Process 3: Disable the vector interrupt of ADT/AD converter.

b7
b0
0
Interrupt control register 2
(ICON2: address 3F$_{16}$)
Disable ADT/AD converter, serial I/O2 interrupts.

Process 4: When using the AD external trigger input, set the ADT pin to the input mode.

b7
b0
0
Port P7 direction register
(P7D: address 0F$_{16}$)
Set P7$_7$ pin to input mode.

Process 5: Set analog input pins to be used to the input mode.

b7
b0
0 0 0 0 0 0 0 0
Port P6 direction register
(P6D: address 0D$_{16}$)
P6$_0$/AN$_0$ pin
P6$_1$/AN$_1$ pin
P6$_2$/AN$_2$ pin
P6$_3$/AN$_3$ pin
P6$_4$/AN$_4$ pin
P6$_5$/AN$_5$ pin
P6$_6$/AN$_6$ pin
P6$_7$/AN$_7$ pin
0: Input mode
1: Output mode

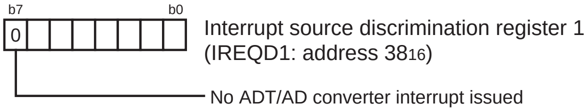
Process 6: Set A-D control register

b7
b0
0 1 1
A-D control register
(ADCON: address 34$_{16}$)
Select analog input pin
000 : P6$_0$/AN$_0$
001 : P6$_1$/AN$_1$
010 : P6$_2$/AN$_2$
011 : P6$_3$/AN$_3$
100 : P6$_4$/AN$_4$
101 : P6$_5$/AN$_5$
110 : P6$_6$/AN$_6$
111 : P6$_7$/AN$_7$
(Note 1)
Connect V$_{REF}$ and ladder resistor
Select AD external trigger valid/invalid
0: AD external trigger input invalid
1: AD external trigger input valid (A-D converter starts with falling edge of ADT pin)
Select interrupt source
0: Interrupt request at A-D conversion completed
1: Interrupt request at ADT input falling (only at AD external trigger input valid)
DA output disabled

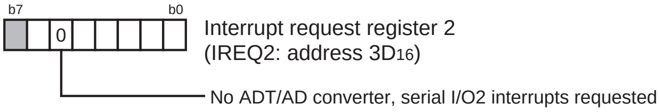
Note 1: When setting “0” to this bit, the A-D conversion is started. Retain “1” to this bit while the A-D control register is set.

Fig. 2.6.5 A-D converter setting method (1)

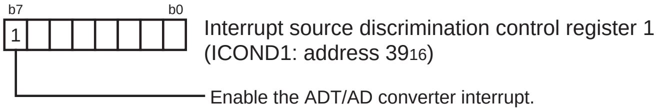
Process 7: Set “0” to the ADT/AD converter interrupt request bit.


b7
b0
0
Interrupt source discrimination register 1
(IREQD1: address 38_{16})
No ADT/AD converter interrupt issued

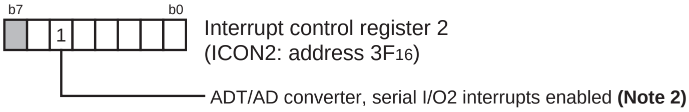
Process 8: Set “0” to the ADT/AD converter interrupt request bit of each vector.


b7
b0
0
Interrupt request register 2
(IREQ2: address $3D_{16}$)
No ADT/AD converter, serial I/O2 interrupts requested

Process 9: When using the interrupt, enable the ADT/AD converter interrupt.


b7
b0
1
Interrupt source discrimination control register 1
(ICOND1: address 39_{16})
Enable the ADT/AD converter interrupt.

Process 10: When using the interrupt, enable the ADT/AD converter interrupt of each vector.

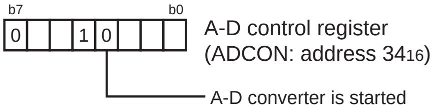

b7
b0
1
Interrupt control register 2
(ICON2: address $3F_{16}$)
ADT/AD converter, serial I/O2 interrupts enabled **(Note 2)**

Note 2: When enabling the serial I/O2 interrupt, the serial I/O2 interrupt will occur by setting “1” to the serial I/O2 interrupt enable bit.

Process 11: When the interrupt disable flag (I) is set to “1” in “process 1”, set the interrupt disable flag (I) to “0”.

Process 12: Start A-D converter

- When the AD external trigger is invalid, set “0” to the AD converter completion bit.



- When the AD external trigger is valid, the A-D converter is started by inputting the falling signal of the external pulse.

Fig. 2.6.6 A-D converter setting method (2)

(2) A-D converter application examples

● **Outline**

The voltage and temperature of a battery are detected by using A-D converter.

● **Specifications**

A-D conversion is performed at 1-second intervals and data of a battery voltage and a battery temperature is input. The voltage data or the temperature data is input in ADT/AD converter interrupt processing routine which is generated at A-D conversion completed. Analog input pins are also selected.

Figure 2.6.7 shows the peripheral circuit example, and Figures 2.6.8 and 2.6.9 show the control procedure example.

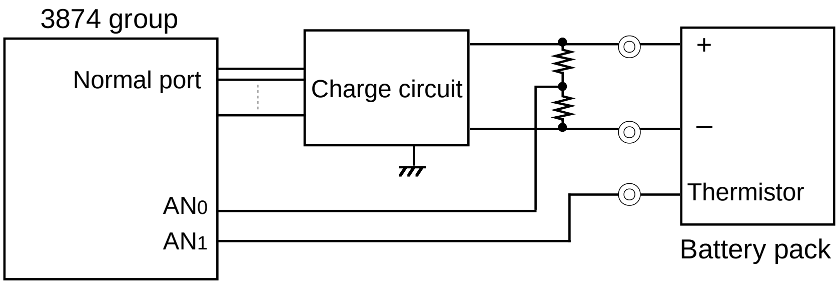
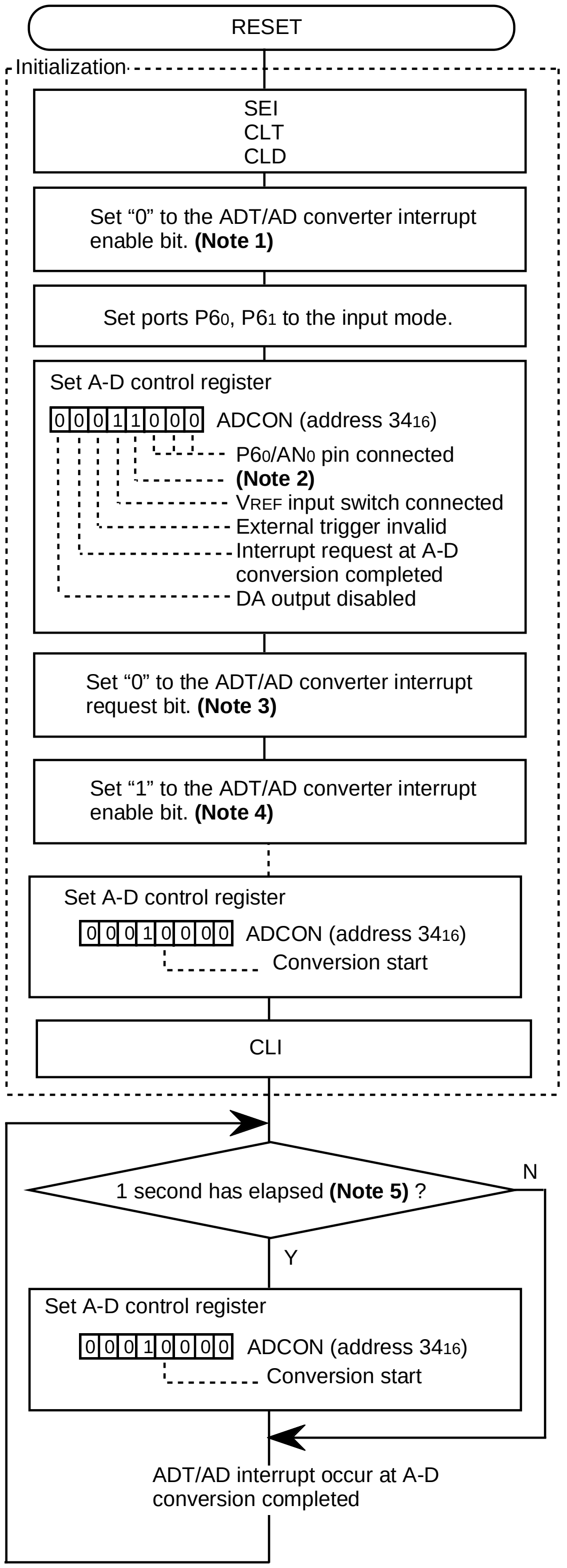


Fig. 2.6.7 Peripheral circuit example


RESET
Initialization
SEI
CLT
CLD
Set “0” to the ADT/AD converter interrupt enable bit. (Note 1)
Set ports P60, P61 to the input mode.
Set A-D control register
0 0 0 1 1 0 0 0 ADCON (address 3416)
P60/AN0 pin connected
(Note 2)
VREF input switch connected
External trigger invalid
Interrupt request at A-D conversion completed
DA output disabled
Set “0” to the ADT/AD converter interrupt request bit. (Note 3)
Set “1” to the ADT/AD converter interrupt enable bit. (Note 4)
Set A-D control register
0 0 0 1 0 0 0 0 ADCON (address 3416)
Conversion start
CLI
1 second has elapsed (Note 5) ?
N
Y
Set A-D control register
0 0 0 1 0 0 0 0 ADCON (address 3416)
Conversion start
ADT/AD interrupt occur at A-D conversion completed

Notes 1: Set the ADT/A-D converter, serial I/O2 interrupt enable bit of the interrupt control register 2 and the ADT/A-D converter interrupt enable bit of the interrupt source discrimination control register 1 to disable state.

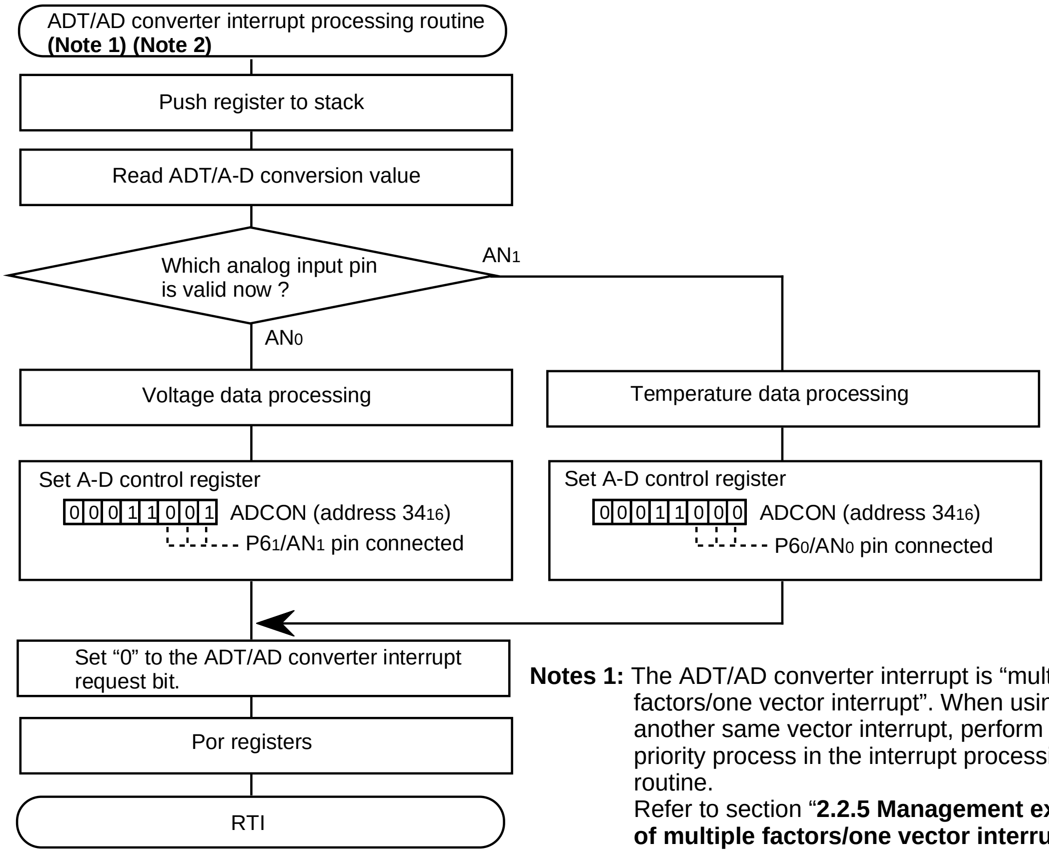
2: When writing “0” to this bit, the A-D conversion is performed.

3: Set “0” to the ADT/A-D converter, serial I/O2 interrupt request bit of the interrupt request register 2 and the ADT/A-D converter interrupt request bit of the interrupt source discrimination control register 1.

4: Set the ADT/A-D converter, serial I/O2 interrupt enable bit of the interrupt control register 2 and the ADT/A-D converter interrupt enable bit of the interrupt source discrimination control register 1 to enable state.

5: Generate 1 second with the timer.

Fig. 2.6.8 Control procedure example (1)


ADT/AD converter interrupt processing routine **(Note 1) (Note 2)**
Push register to stack
Read ADT/A-D conversion value
Which analog input pin is valid now ?
AN$_1$
AN$_0$
Voltage data processing
Temperature data processing
Set A-D control register
0 0 0 1 1 0 0 1 ADCON (address 34_{16})
P6$_1$/AN$_1$ pin connected
Set A-D control register
0 0 0 1 1 0 0 0 ADCON (address 34_{16})
P6$_0$/AN$_0$ pin connected
Set "0" to the ADT/AD converter interrupt request bit.
Por registers
RTI

Notes 1: The ADT/AD converter interrupt is "multiple factors/one vector interrupt". When using another same vector interrupt, perform the priority process in the interrupt processing routine.
Refer to section "**2.2.5 Management example of multiple factors/one vector interrupt**".

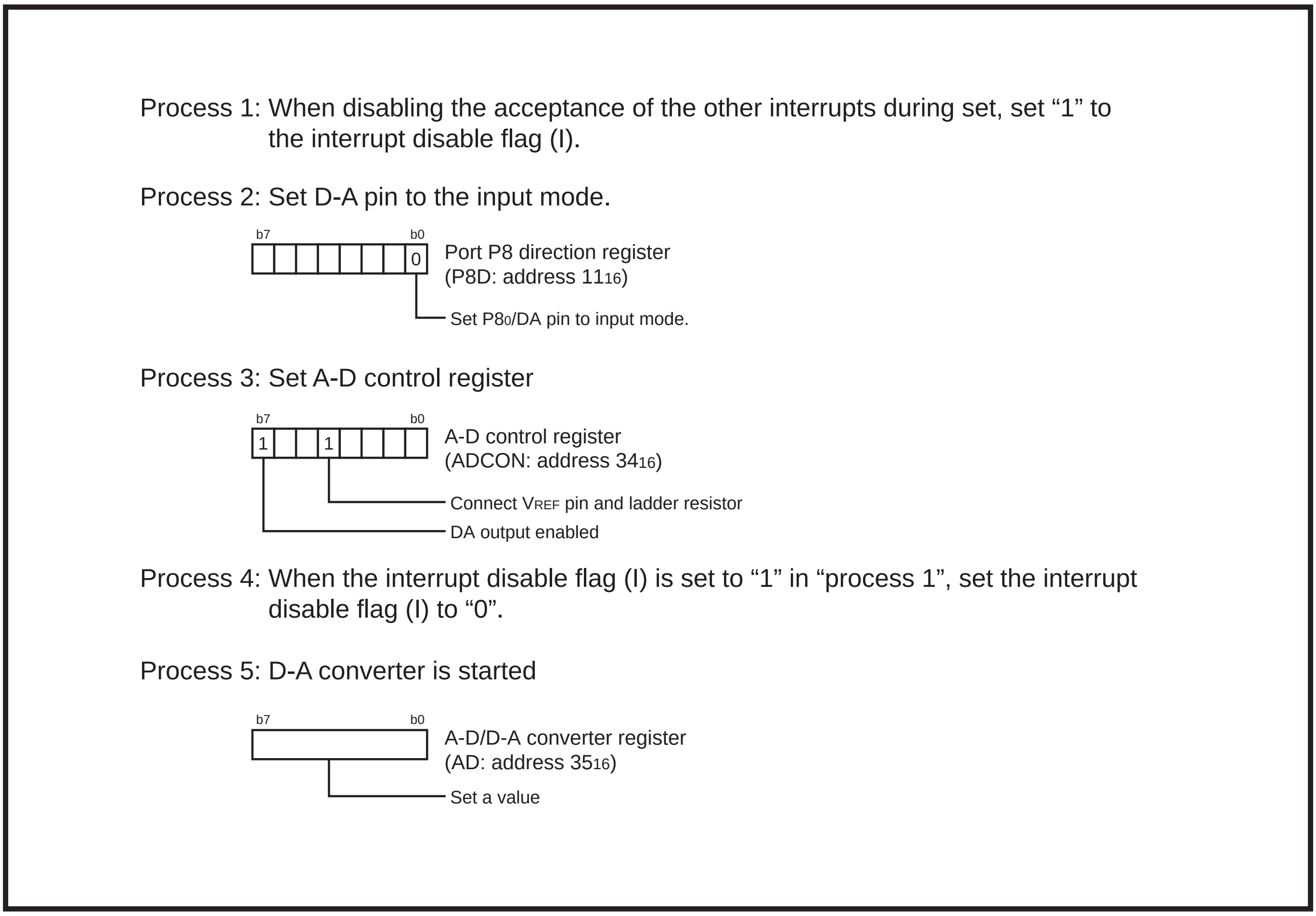
2: Confirm an A-D value, using the A-D conversion result of several times.

Fig. 2.6.9 Control procedure example (2)

2.6.4 D-A converter

(1) Setting method of D-A converter

Figure 2.6.10 shows the setting method of D-A converter.



Process 1: When disabling the acceptance of the other interrupts during set, set “1” to the interrupt disable flag (I).

Process 2: Set D-A pin to the input mode.

b7
b0
0
Port P8 direction register
(P8D: address 11_{16})
Set $P8_0$/DA pin to input mode.

Process 3: Set A-D control register

b7
b0
1
1
A-D control register
(ADCON: address 34_{16})
Connect V_{REF} pin and ladder resistor
DA output enabled

Process 4: When the interrupt disable flag (I) is set to “1” in “process 1”, set the interrupt disable flag (I) to “0”.

Process 5: D-A converter is started

b7
b0
A-D/D-A converter register
(AD: address 35_{16})
Set a value

Fig. 2.6.10 D-A converter setting method

(2) D-A converter application examples

● **Outline**

The volume of a speaker output is modulated by using D-A converter.

● **Specifications**

For a musical interval, the sound period is modulated by timer X so that the definite sound (“la”: about 440 Hz) can be output.
For the volume of sound, an amplitude of sound is modulated by D-A output value. The D-A output value is switched in the timer X interrupt processing routine.
$f(X_{IN})$ = 6 MHz is used.

Figure 2.6.11 shows the speaker output example and Figure 2.6.12 shows the peripheral circuit example.
Figures 2.6.13 and 2.6.14 shows the control procedure example.

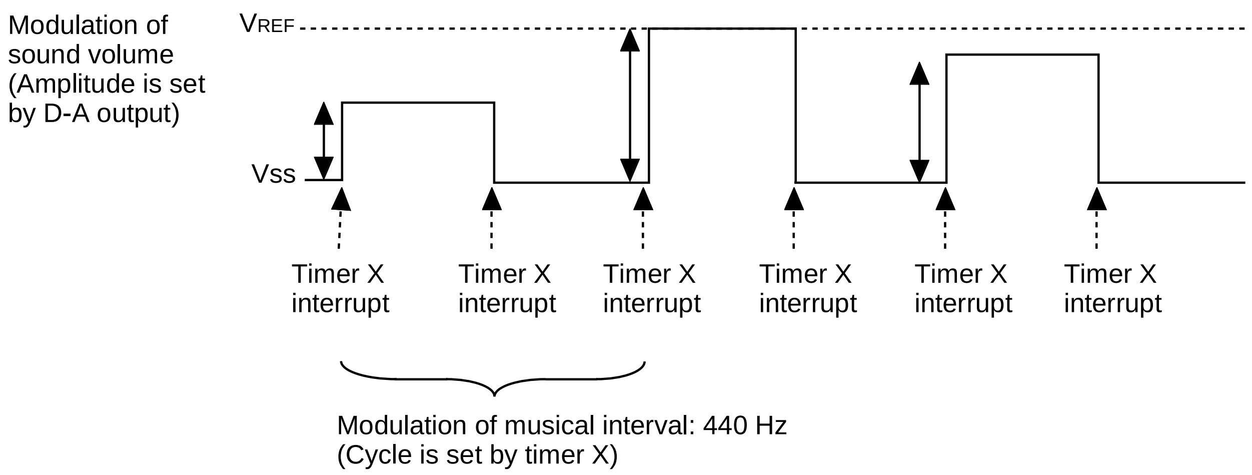


Fig. 2.6.11 Speaker output example

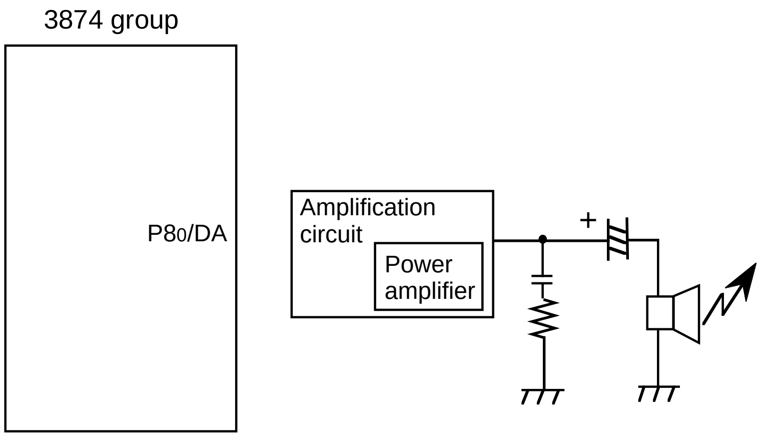
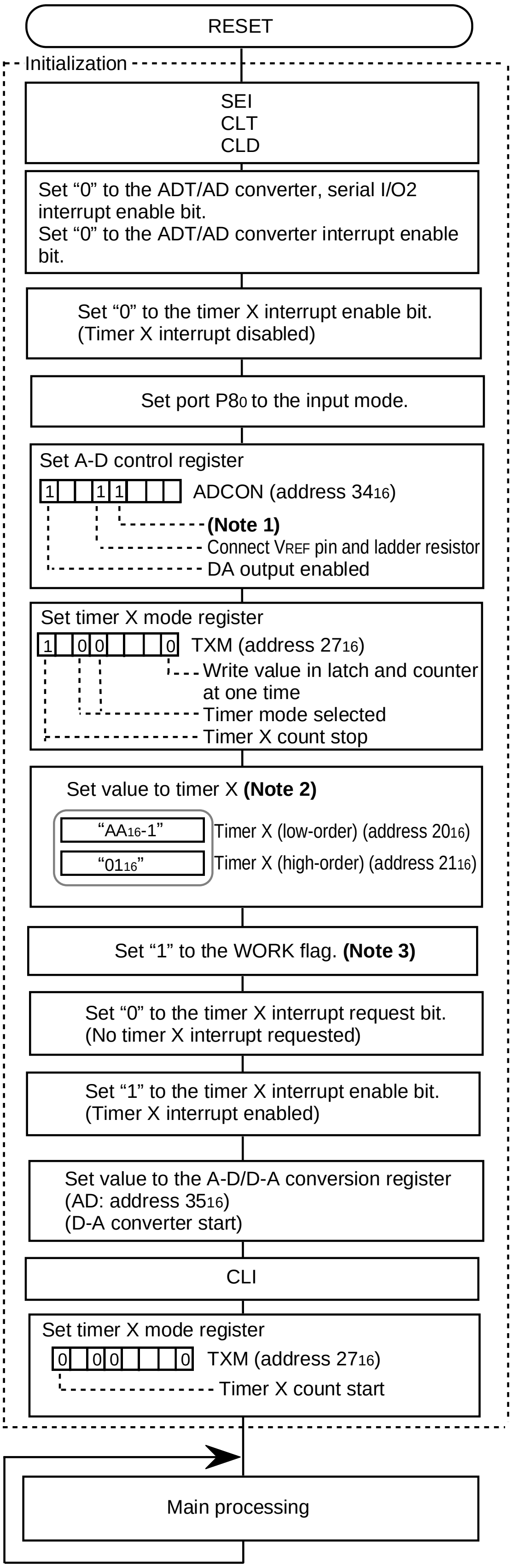


Fig. 2.6.12 Peripheral circuit example


RESET
Initialization
SEI
CLT
CLD
Set “0” to the ADT/AD converter, serial I/O2 interrupt enable bit.
Set “0” to the ADT/AD converter interrupt enable bit.
Set “0” to the timer X interrupt enable bit.
(Timer X interrupt disabled)
Set port P8₀ to the input mode.
Set A-D control register
1 1 1
ADCON (address 34₁₆)
(Note 1)
Connect VREF pin and ladder resistor
DA output enabled
Set timer X mode register
1 0 0 0
TXM (address 27₁₆)
Write value in latch and counter at one time
Timer mode selected
Timer X count stop
Set value to timer X (Note 2)
“AA₁₆-1”
“01₁₆”
Timer X (low-order) (address 20₁₆)
Timer X (high-order) (address 21₁₆)
Set “1” to the WORK flag. (Note 3)
Set “0” to the timer X interrupt request bit.
(No timer X interrupt requested)
Set “1” to the timer X interrupt enable bit.
(Timer X interrupt enabled)
Set value to the A-D/D-A conversion register
(AD: address 35₁₆)
(D-A converter start)
CLI
Set timer X mode register
0 0 0 0
TXM (address 27₁₆)
Timer X count start
Main processing

Notes 1: When writing “0” to this bit, A-D conversion is performed.

2: When setting a value to the timer, set in order of the low-order byte and the high-order byte following.

3: The WORK flag is a user flag for work. When this flag is “1”, a value except Vss is output from the DA output pin.
When this flag is “0”, Vss is output from the DA output pin.

Fig. 2.6.13 Control procedure example (1)

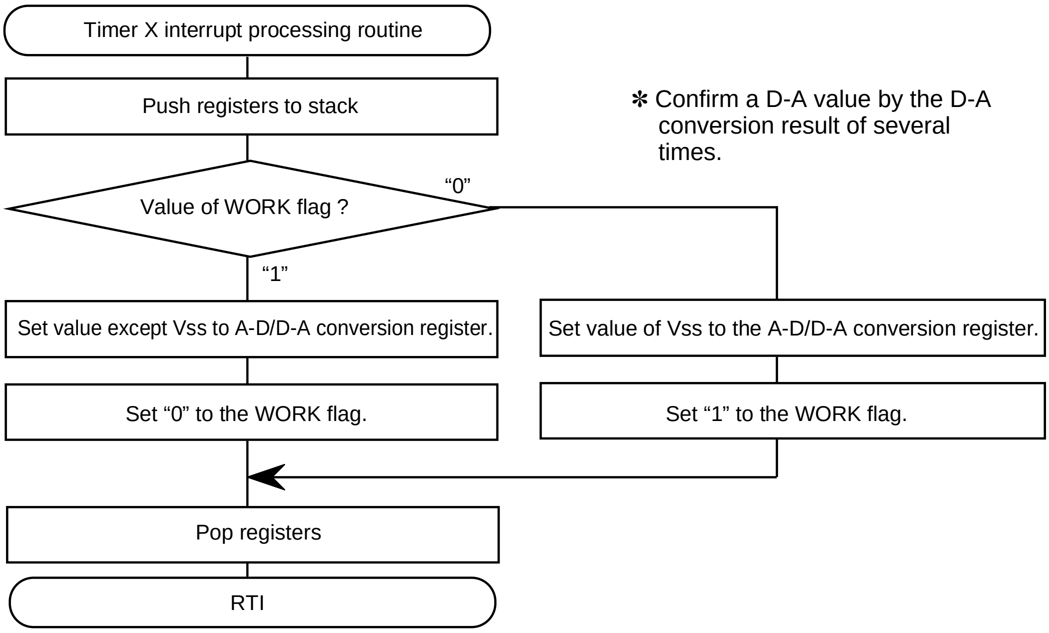

Timer X interrupt processing routine
Push registers to stack
✻ Confirm a D-A value by the D-A conversion result of several times.
Value of WORK flag ?
"0"
"1"
Set value except Vss to A-D/D-A conversion register.
Set value of Vss to the A-D/D-A conversion register.
Set "0" to the WORK flag.
Set "1" to the WORK flag.
Pop registers
RTI

Fig. 2.6.14 Control procedure example (2)

2.6.5 Notes on use

(1) Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01µF to 1µF. 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

Pins AVSS are A-D converter power source pins. Regardless of using the A-D conversion function or not, connect them as following :

• AVss : Connect to the Vss line

● **Reason**

If the AVCC and the AVSS pin are opened, the microcomputer may have a failure because of noise or others. Also, if the AVCC pin is connected to the VSS pin, current flows from AVCC to VSS.

(3) Reference voltage input pin V_{REF}

When using A-D converter or D-A converter, apply a voltage of 2 V to Vcc to the reference voltage input pin V_{REF}. Note that if the V_{REF} value is lowered, the accuracy degrades.

(4) 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 and **WIT** instruction

(5) A-D/D-A conversion register

The A-D/D-A conversion register functions as an A-D conversion register during a read and a D-A conversion during a write. Accordingly, the D-A conversion register set value cannnot be read out.

(6) Inputting falling signal to ADT pin during A-D conversion when using external trigger

When inputting the falling signal to the ADT pin during A-D conversion, A-D conversion which is in progress is stopped, and A-D conversion is started again.

(7) Set analog input pin to input mode

When using the A-D converter, set port P6 pins used as the analog input pins to an input port.
When using the D-A converter, set $P8_0$/DA pin as input port.

(8) Connecting D-A output pin to low-impedance load when using D-A converter

The DA output pin does not include a buffer, so that use an external buffer when connecting a peripheral circuit having a low-impedance load.

(9) Vcc when using D-A converter

The D-A converter accuracy when Vcc is 4.0 V or less differs from that of when Vcc is 4.0 V or more.
When using the D-A converter, we recomend that Vcc is used at 4.0 V or more.

2.7 Watchdog timer

The watchdog timer is a 20-bit down-count counter and consists of a low-order 8 bits and a high-order 12 bits. “1” is subtracted from the watchdog timer each time a count source inputs.
The watchdog timer can be also used as a 16-bit timer.

2.7.1 Memory assignment

Figure 2.7.1 shows the memory assignment of watchdog timer relevant register.

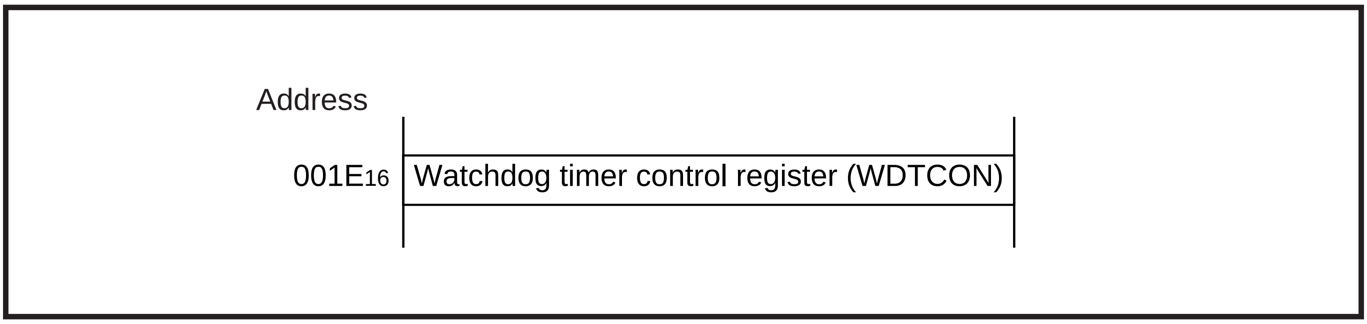


Fig. 2.7.1 Memory assignment of watchdog timer relevant register

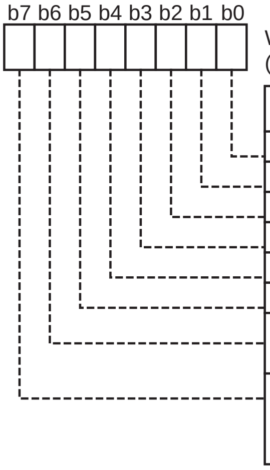
2.7.2 Relevant register

(1) Watchdog timer control register

The watchdog timer starts counting by writing an arbitrary value to the watchdog timer control register.

Figure 2.7.2 shows the structure of the watchdog timer control register.

Watchdog timer control register



Watchdog timer control register
(WDTCON: address $1E_{16}$)

b	Name	Functions	At reset	R	W
0	Watchdog timer H (high-order 6 bits of reading exclusive)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0: STP instruction enabled 1: STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0: Watchdog timer L underflow 1: f(X_{IN})/16 or f(X_{CIN})/16	0	○	○

Fig. 2.7.2 Structure of watchdog timer control register

2.7.3 Watchdog timer application example

● **Outline**

When a program runs away, the watchdog timer make the microcomputer return to the reset state.

● **Specifications**

•When the watchdog timer H underflows, it is judged as unexpected program, and the microcomputer is returned to the reset state.
•Bit 7 of the watchdog timer control register is set to “0” at 1-cycle intervals in the main routine before underflow of the watchdog timer H. (Initialization of watchdog timer value)
•f(X_{IN})/2 (high-speed mode) is used as the system clock.
•Underflow of the watchdog timer L is used as the count source of the watchdog timer H.
When 2.6 s(at f(X_{IN}) = 6.4 MHz) has passed, the watchdog timer L underflow.
Figure 2.7.3 shows the connection of watchdog timer and the setting of the division ratio.
Figure 2.7.4 shows the control procedure example.

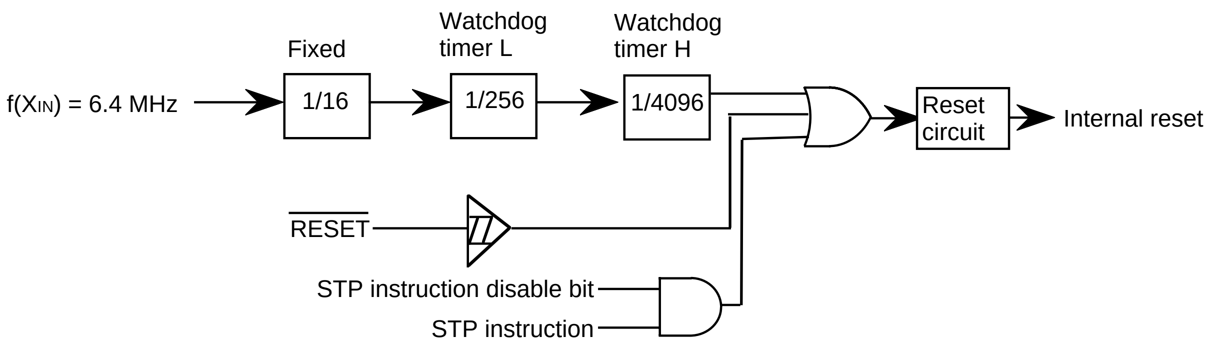


Fig. 2.7.3 Connection of watchdog timer and setting of division ratio

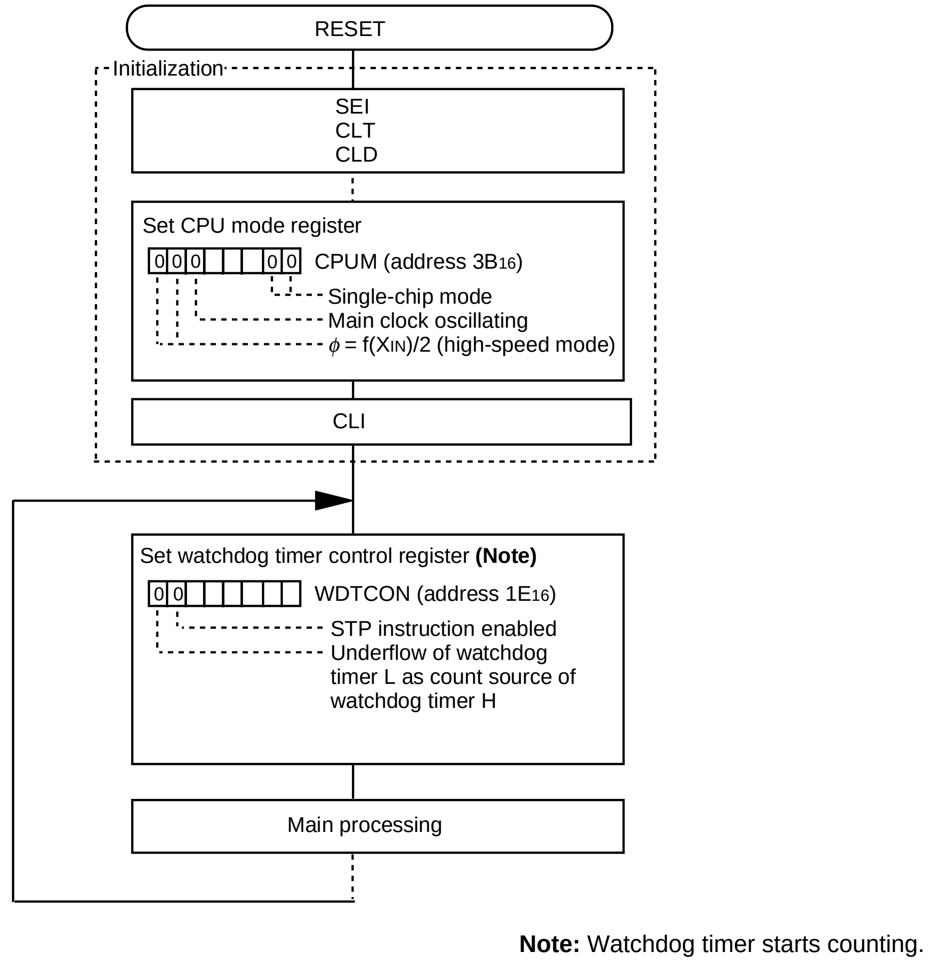


Fig. 2.7.4 Control procedure example

2.7.4 Notes on use

- The watchdog timer continues to count even while waiting for Stop release. Accordingly, make sure that watchdog timer does not underflow during this term.
- Once a “1” is written to the STP instruction disable bit of the watchdog timer control register, it cannot be programmed to “0” again.

2.8 Reset circuit

The reset state is caused by applying an "L" level to the $\overline{\text{RESET}}$ pin. After that, the reset state is released by applying an "H" level to the $\overline{\text{RESET}}$ pin, so that the program is executed in the middle-speed mode from the contents of the reset vector address.

2.8.1 Connection example of reset IC

Figure 2.8.1 shows the example of power-on reset circuit. Figure 2.8.2 shows the system example which switches to the RAM back-up mode by detecting a drop of the system power source voltage with the INT interrupt.

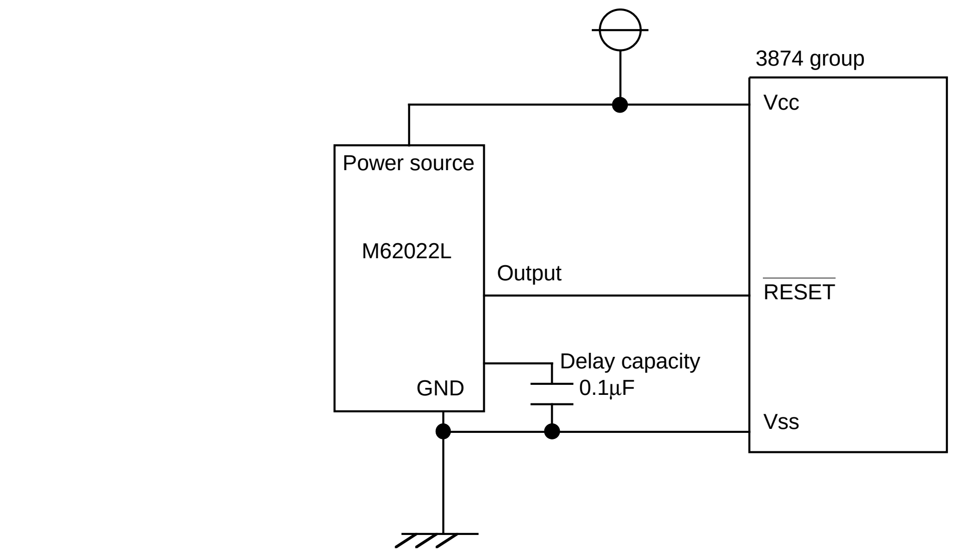


Fig. 2.8.1 Example of power-on reset circuit

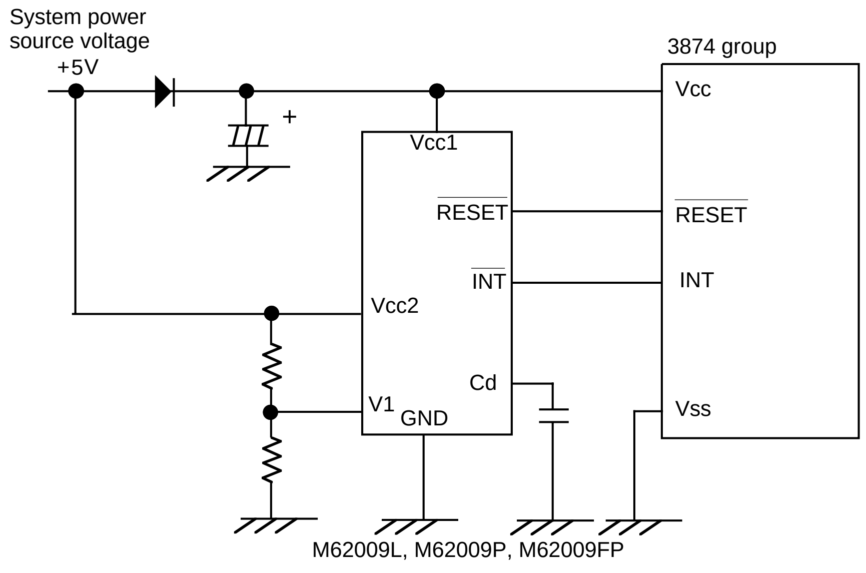


Fig. 2.8.2 RAM back-up system example

2.8.2 Notes on use

(1) Reset input voltage control

Make sure that the reset input voltage is 0.6 V or less for Vcc of 3.0 V.

(2) 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 Vss pin. And 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.

2.9 Clock generating circuit

The 3874 group has two built-in oscillation circuits to obtain clocks required for operation.

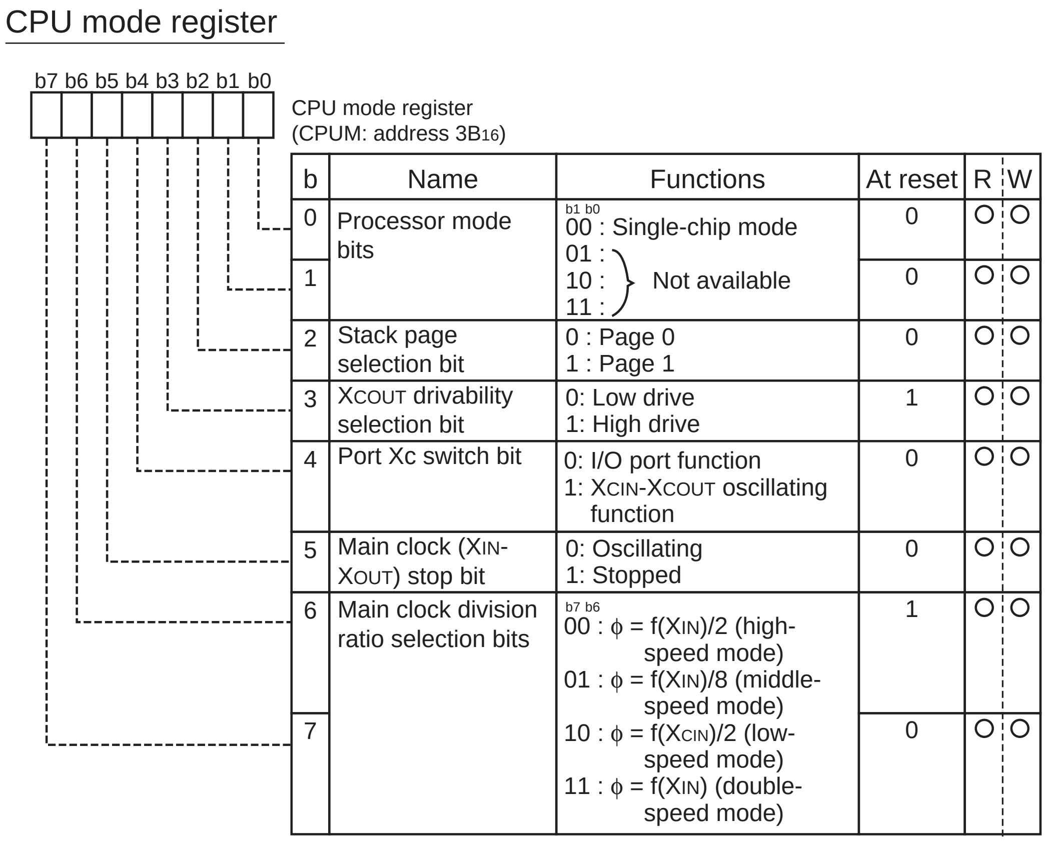
- X_{IN}-X_{OUT} oscillation circuit
  Main clock (X_{IN} input) oscillation circuit
- X_{CIN}-X_{COUT} oscillation circuit
  Sub-clock (X_{CIN}) oscillation circuit

In the following modes, CPU can operate and function in the low power dissipation.

- Stop mode by STP instruction execution
- Wait mode by WIT instruction execution

2.9.1 Relevant register

Figure 2.9.1 shows the structure of the CPU mode register.



b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : 10 : } Not available 11 :	0	○	○
1			0	○	○
2	Stack page selection bit	0 : Page 0 1 : Page 1	0	○	○
3	X_{COUT} drivability selection bit	0: Low drive 1: High drive	1	○	○
4	Port Xc switch bit	0: I/O port function 1: X_{CIN}-X_{COUT} oscillating function	0	○	○
5	Main clock (X_{IN}-X_{OUT}) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bits	b7 b6 00 : ϕ = f(X_{IN})/2 (high-speed mode) 01 : ϕ = f(X_{IN})/8 (middle-speed mode) 10 : ϕ = f(X_{CIN})/2 (low-speed mode) 11 : ϕ = f(X_{IN}) (double-speed mode)	1	○	○
7			0	○	○

Fig. 2.9.1 Structure of CPU mode register

2.9.2 Setting method

The operation modes of the 3874 group are classified as follows:

- Reset
- High-speed mode (ϕ = f(X_{IN})/2)
- Middle-speed mode (ϕ = f(X_{IN})/8)
- Low-speed mode (ϕ = f(X_{CIN})/2)
- Double-speed mode (ϕ = f(X_{IN}))
- Stop mode (by STP instruction execution)
- Wait mode (by WIT instruction execution)

For the state transitions of system clock, refer to **"Figure 71"** of "**CHAPTER 1. HARDWARE"**.

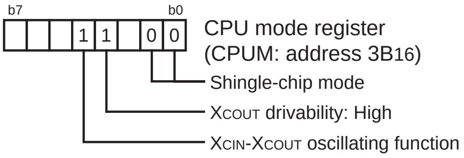
(1) Setting method when using low-speed mode

Figure 2.9.2 shows the switch method from the high-/middle-/double-speed mode to the low-speed mode.

Figure 2.9.3 shows the switch method from the low-speed mode to the high-/middle-speed mode. Figure 2.9.4 shows the switch method from the low-speed mode to the double-speed mode.

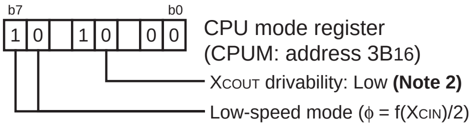
- Switch from high-/middle-/double-speed mode to low-speed mode

Process 1: Set CPU mode register



Process 2: Generate the oscillating stabilizing wait time for f(X_{CIN}) by software. **(Note 1)**

Process 3: Switch the operation mode to the low-speed mode.

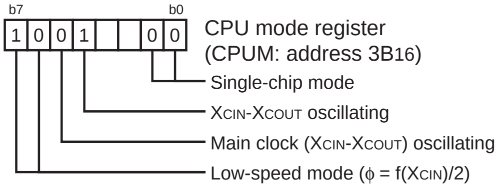

b7
b0
1 0 1 0 0 0
CPU mode register
(CPUM: address 3B16)
XCOUT drivability: Low (Note 2)
Low-speed mode (ϕ = f(XCIN)/2)

Notes 1: For the oscillation stabilizing wait time, ask the resonator manufacturer for information.
2: Set "0" to this bit as necessity.

Fig. 2.9.2 Switch method from high-/middle-/double-speed mode to low-speed mode

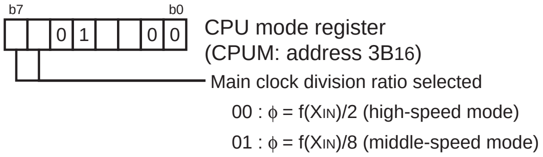
● Switch from low-speed mode to high-/middle-speed mode

Process 1: Set CPU mode register


b7
b0
1 0 0 1 0 0
CPU mode register
(CPUM: address $3B_{16}$)
Single-chip mode
X_{CIN}-X_{COUT} oscillating
Main clock (X_{CIN}-X_{COUT}) oscillating
Low-speed mode (φ = f(X_{CIN})/2)

Process 2: Generate the oscillating stabilizing wait time for f(X_{IN}) by software. **(Note)**

Process 3: Switch the operation mode to the high-/middle-speed mode.

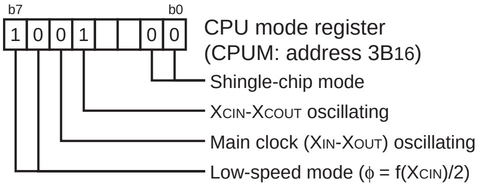

b7
b0
0 1 0 0
CPU mode register
(CPUM: address $3B_{16}$)
Main clock division ratio selected
00 : φ = f(X_{IN})/2 (high-speed mode)
01 : φ = f(X_{IN})/8 (middle-speed mode)

Note : For the oscillation stabilizing wait time, ask the resonator manufacter for information.

Fig. 2.9.3 Switch method from low-speed mode to high-/middle-speed mode

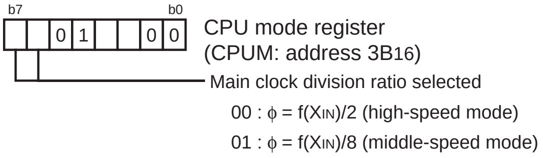
● Switch from low-speed mode to double-speed mode

Process 1: Set CPU mode register


b7
b0
1 0 0 1 0 0
CPU mode register
(CPUM: address $3B_{16}$)
Shingle-chip mode
X_{CIN}-X_{COUT} oscillating
Main clock (X_{IN}-X_{OUT}) oscillating
Low-speed mode (φ = f(X_{CIN})/2)

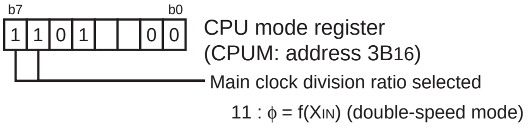
Process 2: Generate the oscillating stabilizing wait time for f(X_{IN}) by software. **(Note)**

Process 3: Switch the operation mode to the high-/middle-speed mode.


b7
b0
0 1 0 0
CPU mode register
(CPUM: address $3B_{16}$)
Main clock division ratio selected
00 : φ = f(X_{IN})/2 (high-speed mode)
01 : φ = f(X_{IN})/8 (middle-speed mode)

Process 4: Generate the clock switch timing wait time by software. (4 machine cycles)

Process 5: Switch the operation mode to the double-speed mode.


b7
b0
1 1 0 1 0 0
CPU mode register
(CPUM: address $3B_{16}$)
Main clock division ratio selected
11 : φ = f(X_{IN}) (double-speed mode)

Note : For the oscillation stabilizing wait time, ask the resonator manufacter for information.

Fig. 2.9.4 Change method from low-speed mode to double-speed mode

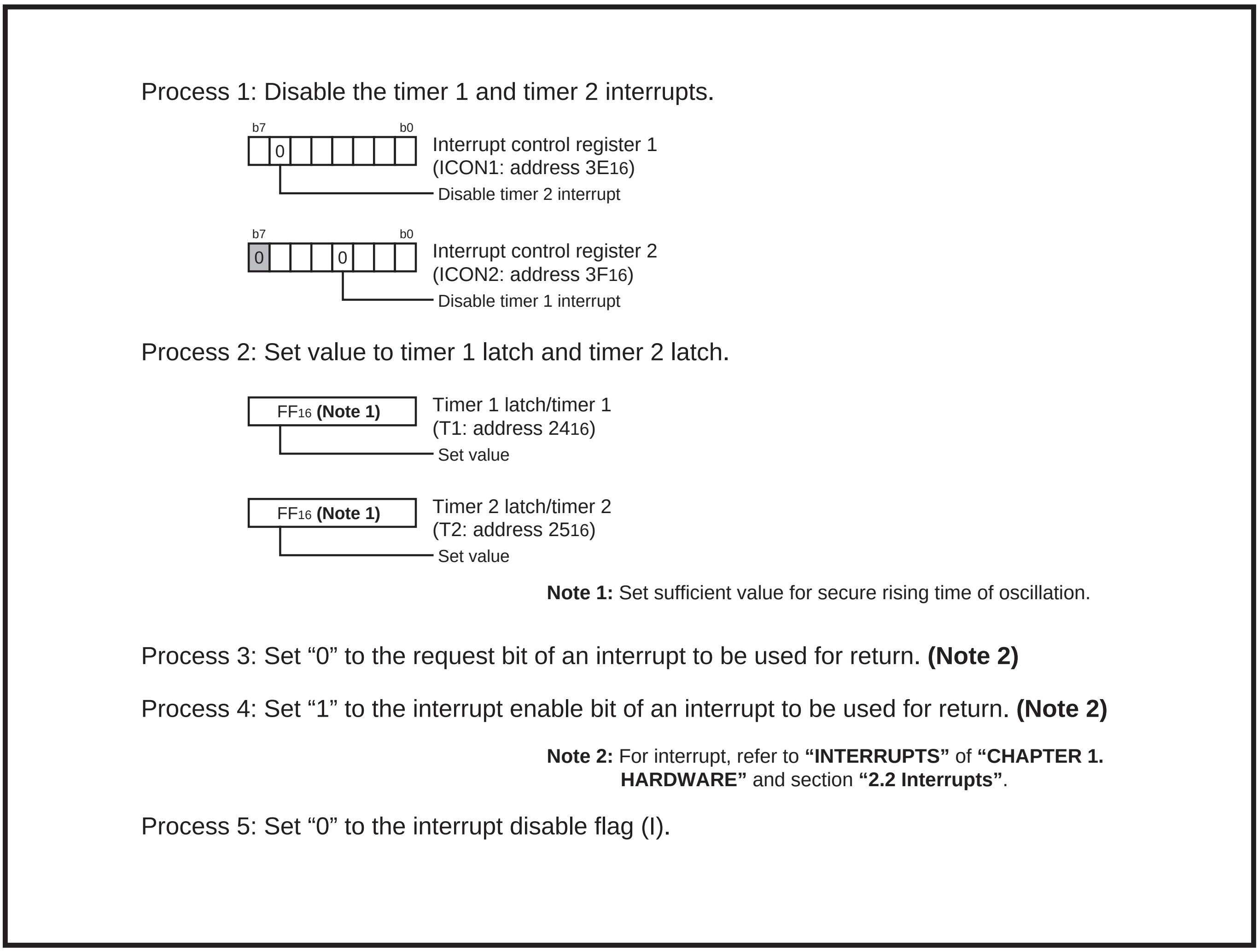
(2) Stop mode release method

The transition from the high-/middle-/low-/double-speed mode to the stop mode can be performed by execution the STP instruction. However, when the STP instruction disable bit of the watchdog timer control register (address 001E$_{16}$) is “1”; if the STP instruction is executed, it is managed as an undefined instruction and reset occurs.

The stop mode is released by reset input or interrupt request occurrence. The interrupt sources to be used for return are shown below.

- ● INT$_0$ to INT$_5$
- ● CNTR$_0$, CNTR$_1$
- ● Serial I/O using external clock
- ● Timer (timer X, timer Y) using external clock
- ● Key input (key-on wake-up)
- ● A-D converter start trigger input

When the above interrupt sources are used for return from the stop mode, perform the setting to enable interrupt to be used shown in Figure 2.9.5 before STP instruction execution.



Process 1: Disable the timer 1 and timer 2 interrupts.

b7
b0
0
Interrupt control register 1
(ICON1: address 3E16)
Disable timer 2 interrupt
b7
b0
0
0
Interrupt control register 2
(ICON2: address 3F16)
Disable timer 1 interrupt

Process 2: Set value to timer 1 latch and timer 2 latch.

Note 1: Set sufficient value for secure rising time of oscillation.

Process 3: Set “0” to the request bit of an interrupt to be used for return. **(Note 2)**

Process 4: Set “1” to the interrupt enable bit of an interrupt to be used for return. **(Note 2)**

Note 2: For interrupt, refer to **“INTERRUPTS”** of **“CHAPTER 1. HARDWARE”** and section **“2.2 Interrupts”**.

Process 5: Set “0” to the interrupt disable flag (I).

Fig. 2.9.5 Return method by interrupt (at STP instruction execution)

(3) Wait mode setting method

The transition from the high-/middle-/low-/double-speed mode to the wait mode can be performed by execution the WIT instruction.

The wait mode is released by reset input or an interrupt request occurrence.

When an interrupt source is used for return from the wait mode, set interrupt to be used to the enable state by performing the setting shown in Figure 2.9.6 before the WIT instruction execution.

Process 1: Set “0” to the request bit to be used from return. **(Note)**

Process 2: Set “1” to the interrupt enable bit of an interrupt to be used for return. **(Note)**

Note: For interrupt, refer to **“INTERRUPTS”** of **“CHAPTER 1. HARDWARE”** and section **“2.2 Interrupts”**.

Process 3: Set “0” to the interrupt disable flag (I).

Fig. 2.9.6 Return method by interrupt (at WIT instruction execution)

2.9.3 Clock generating circuit application example

(1) Application example 1 (state transition at power stoppage)

● **Outline**

At power stoppage, the clock is counted up 1-second intervals by using the timer interrupt.

● **Specifications**

•The power dissipation is reduced as low as possible as, maintaining the clock function.
•Clock: f(X_{IN}) = 4.19 MHz (4.194304M = 2^{22}M)
•Sub-clock: f(X_{CIN}) = 32.768 kHz

Figure 2.9.7 shows the peripheral circuit example and Figure 2.9.8 shows the state transition at power stoppage.
Figure 2.9.9 shows the control procedure example.

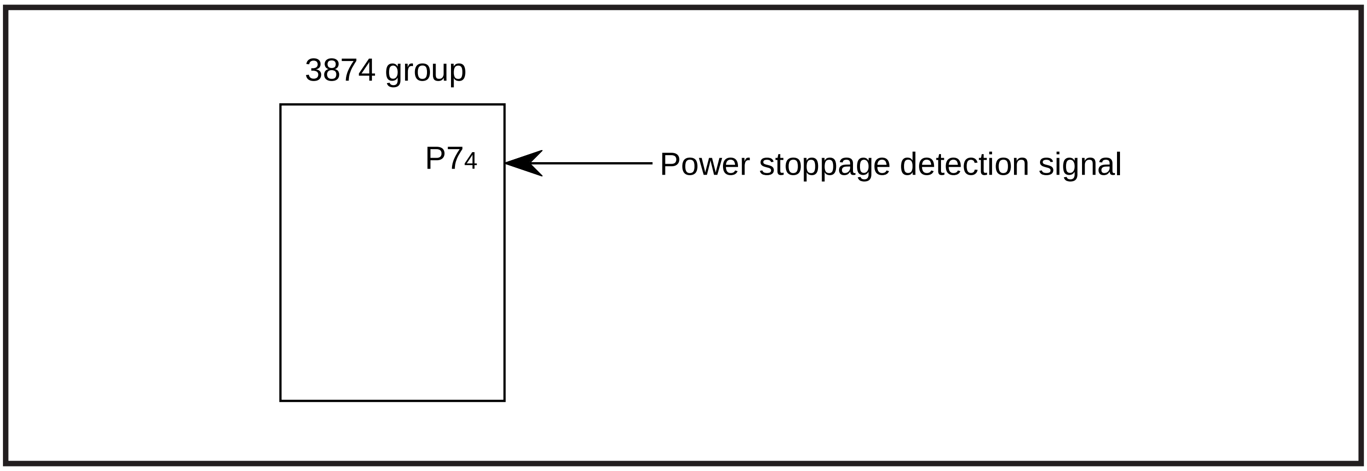


Fig. 2.9.7 Peripheral circuit example

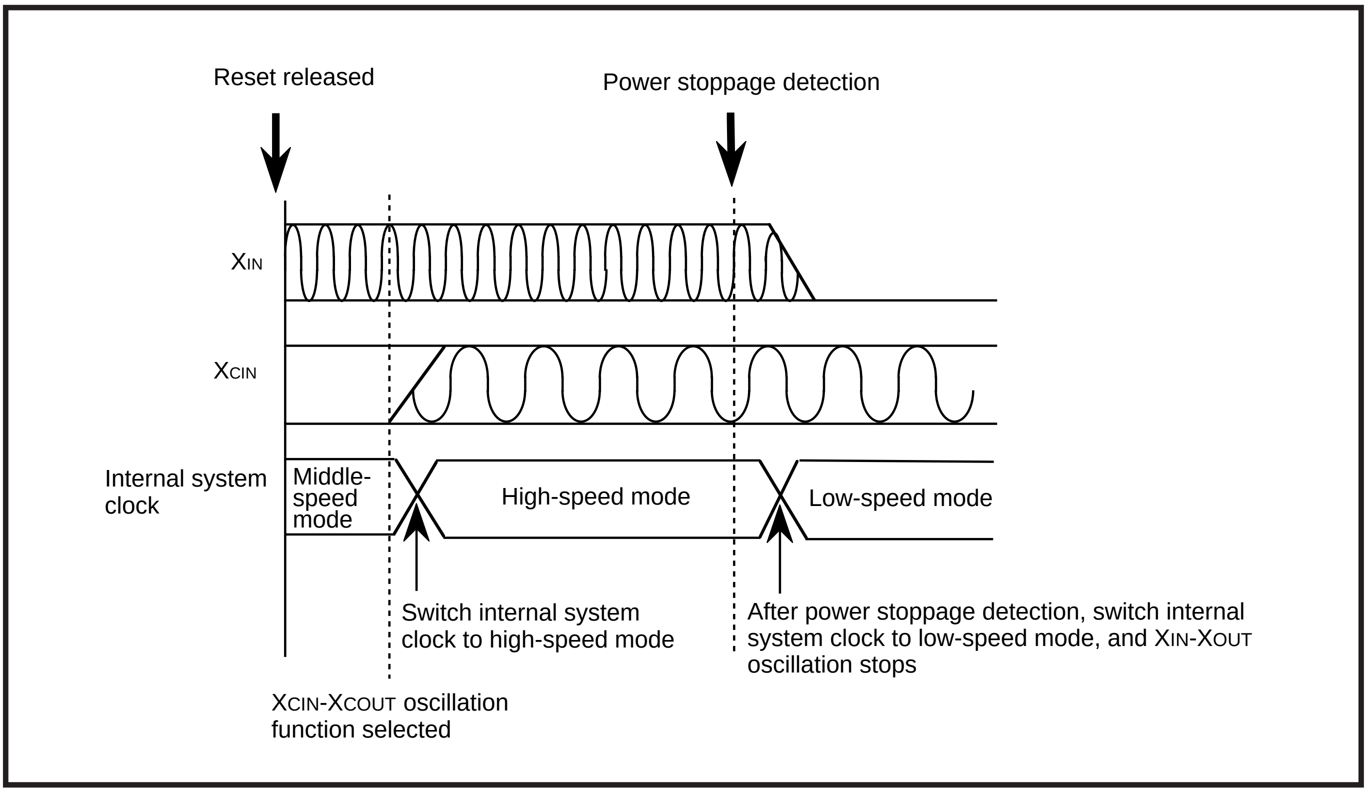
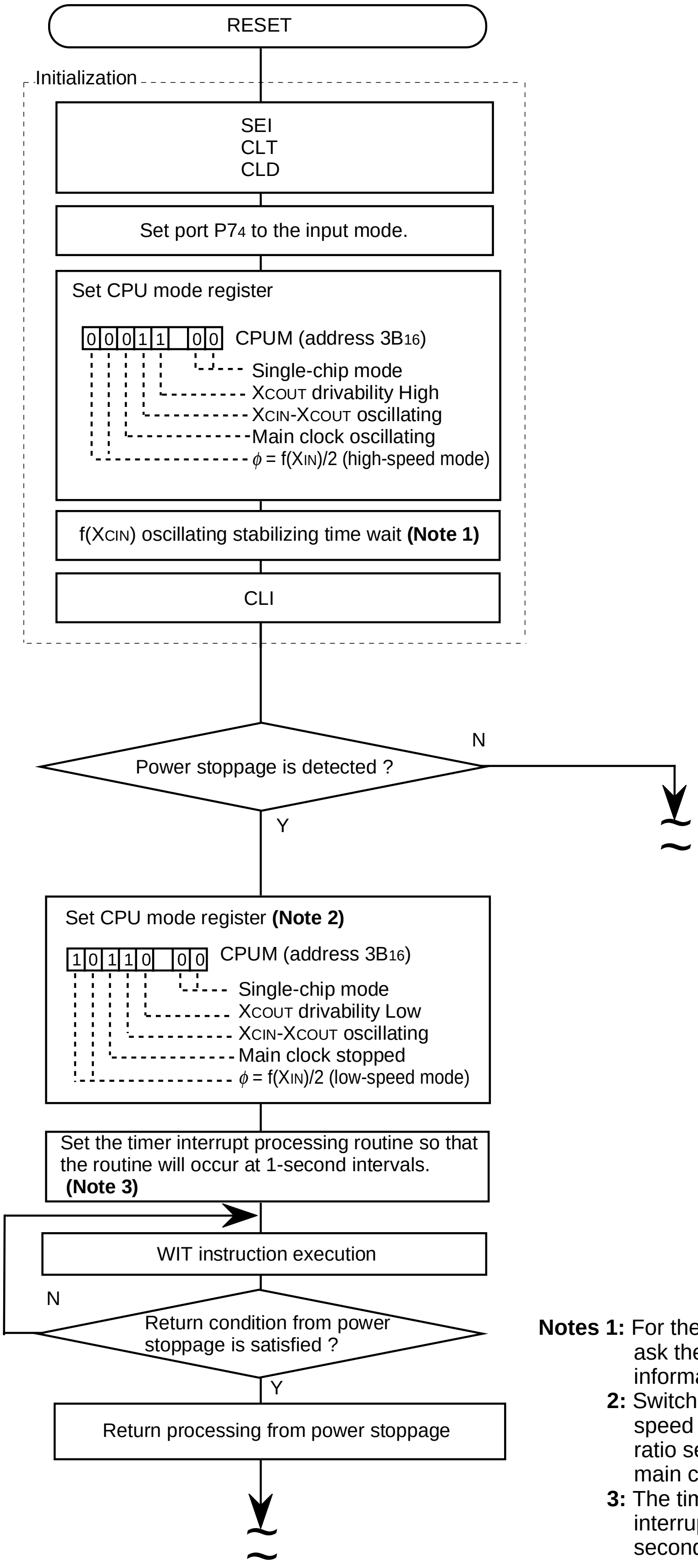

Reset released
Power stoppage detection
XIN
XCIN
Internal system clock
Middle-speed mode
High-speed mode
Low-speed mode
Switch internal system clock to high-speed mode
After power stoppage detection, switch internal system clock to low-speed mode, and XIN-XOUT oscillation stops
XCIN-XCOUT oscillation function selected

Fig. 2.9.8 State transition at power stoppage


RESET
Initialization
SEI
CLT
CLD
Set port $P7_4$ to the input mode.
Set CPU mode register
0 0 0 1 1 0 0
CPUM (address $3B_{16}$)
Single-chip mode
XCOUT drivability High
XCIN-XCOUT oscillating
Main clock oscillating
ϕ = f(XIN)/2 (high-speed mode)
f(XCIN) oscillating stabilizing time wait (Note 1)
CLI
Power stoppage is detected ?
N
Y
Set CPU mode register (Note 2)
1 0 1 1 0 0 0
CPUM (address $3B_{16}$)
Single-chip mode
XCOUT drivability Low
XCIN-XCOUT oscillating
Main clock stopped
ϕ = f(XIN)/2 (low-speed mode)
Set the timer interrupt processing routine so that the routine will occur at 1-second intervals. (Note 3)
WIT instruction execution
N
Return condition from power stoppage is satisfied ?
Y
Return processing from power stoppage

Notes 1: For the oscillation stabilizing wait time, ask the resonator manufacturer for information.

2: Switch the operation mode to the low-speed mode by the main clock division ratio selection bit before stopping the main clock (XIN-XOUT).

3: The timer counts up in the timer interrupt processing routine (every 1 second) at power stoppage.

Fig. 2.9.9 Control procedure example

(2) Application example 2 (Correct count method of timer at power stoppage)

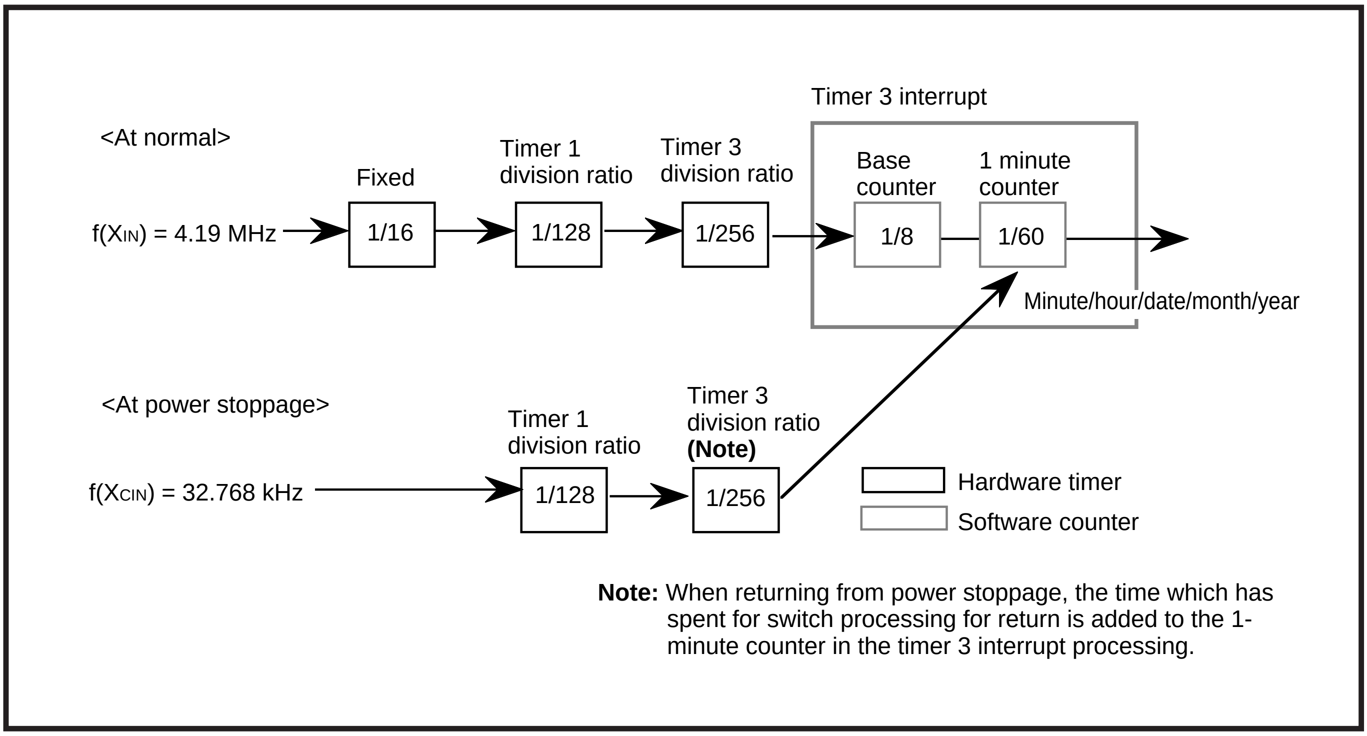
● **Outline**

The correct count of the timer is maintained at power stoppage.

● **Specifications**

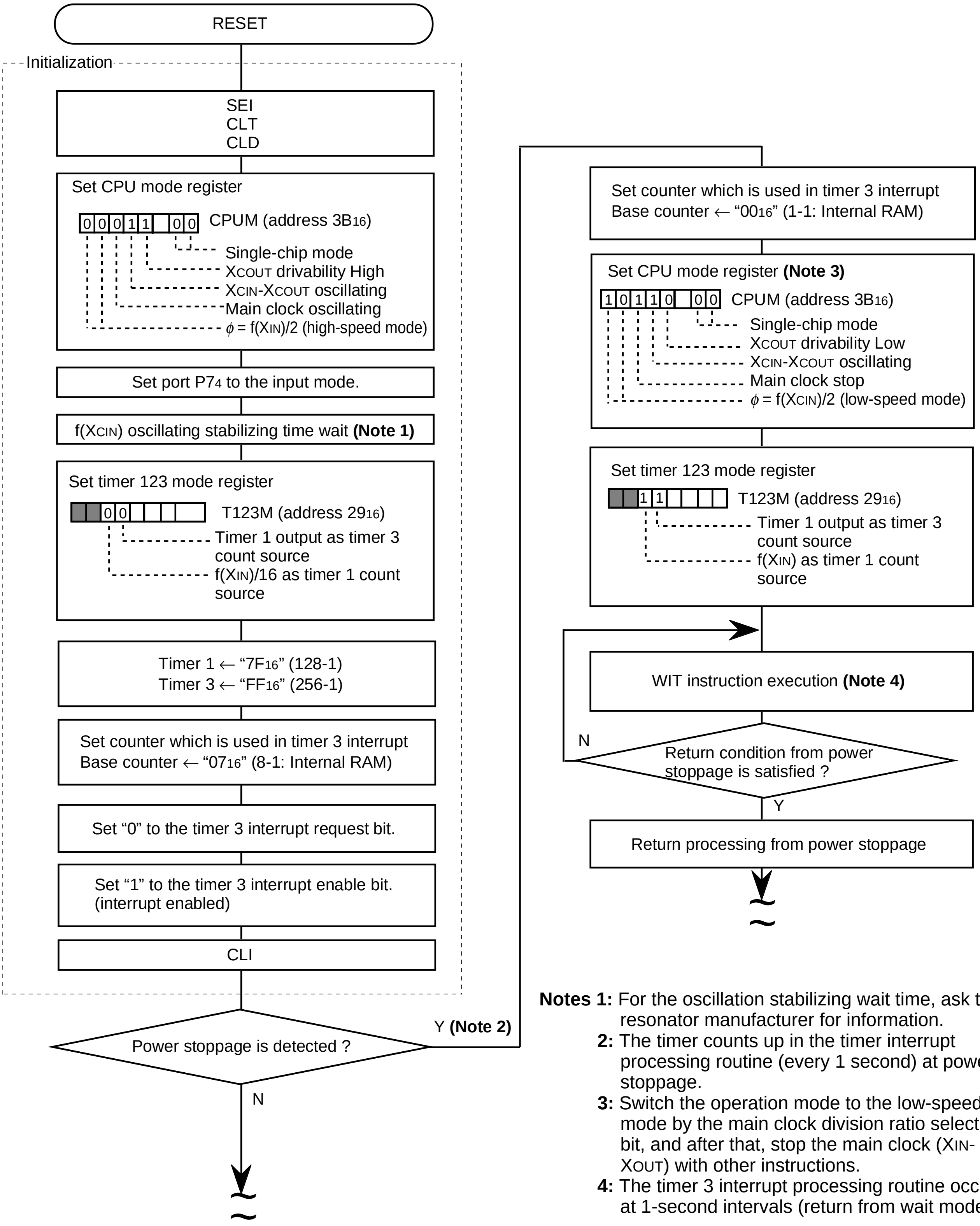
•The power dissipation is reduced as low as possible as, maintaining the clock function.
•Clock: $f(X_{IN})$ = 4.19 MHz (4.194304M = 2^{22}M)
•Sub-clock: $f(X_{CIN})$ = 32.768 kHz
•Timer 3 interrupt used

For the peripheral circuit example and the state transition at power stoppage, refer to Figures 2.9.7 and 2.9.8. Figure 2.9.10 shows the structure of the timer counter and Figure 2.9.11 shows the control procedure example.



Note: When returning from power stoppage, the time which has spent for switch processing for return is added to the 1-minute counter in the timer 3 interrupt processing.

Fig. 2.9.10 Structure of timer counter


RESET
Initialization
SEI
CLT
CLD
Set CPU mode register
0 0 0 1 1 0 0
CPUM (address 3B16)
Single-chip mode
XCOUT drivability High
XCIN-XCOUT oscillating
Main clock oscillating
φ = f(XIN)/2 (high-speed mode)
Set port P74 to the input mode.
f(XCIN) oscillating stabilizing time wait (Note 1)
Set timer 123 mode register
0 0
T123M (address 2916)
Timer 1 output as timer 3 count source
f(XIN)/16 as timer 1 count source
Timer 1 ← "7F16" (128-1)
Timer 3 ← "FF16" (256-1)
Set counter which is used in timer 3 interrupt
Base counter ← "0716" (8-1: Internal RAM)
Set "0" to the timer 3 interrupt request bit.
Set "1" to the timer 3 interrupt enable bit. (interrupt enabled)
CLI
Power stoppage is detected ?
Y (Note 2)
N
Set counter which is used in timer 3 interrupt
Base counter ← "0016" (1-1: Internal RAM)
Set CPU mode register (Note 3)
1 0 1 1 0 0 0
CPUM (address 3B16)
Single-chip mode
XCOUT drivability Low
XCIN-XCOUT oscillating
Main clock stop
φ = f(XCIN)/2 (low-speed mode)
Set timer 123 mode register
1 1
T123M (address 2916)
Timer 1 output as timer 3 count source
f(XIN) as timer 1 count source
WIT instruction execution (Note 4)
Return condition from power stoppage is satisfied ?
N
Y
Return processing from power stoppage

Notes 1: For the oscillation stabilizing wait time, ask the resonator manufacturer for information.
2: The timer counts up in the timer interrupt processing routine (every 1 second) at power stoppage.
3: Switch the operation mode to the low-speed mode by the main clock division ratio selection bit, and after that, stop the main clock (XIN-XOUT) with other instructions.
4: The timer 3 interrupt processing routine occurs at 1-second intervals (return from wait mode).

Fig. 2.9.11 Control procedure example

2.9.4 Notes on use

(1) In all mode

- Use the circuit constants recommended by the resonator manufacturer.
- No external resistor is needed between pins X_{IN} and X_{OUT} because a feed-back resistor is included. However, an external feed-back resistor is needed between pins X_{CIN} and X_{COUT}.
- Use the oscillation circuit of the XCIN side in the condition that the pull-up circuit between pins XCIN and XCOUT is valid.
- If switching the mode between low-speed and double-speed, switch the mode to middle/high-speed first, and then switch the mode to double-speed by program. Do not switch the mode from low-speed to double-speed directly. 1 to 4 machine cycles are required for switching from low-speed mode to other mode. Insert “clock switch timing wait” for switching the mode to middle/high-speed certainly, and then switch the mode to double-speed.
  Table 2.9.1 lists the recommended transition process for system clock switch.
  Figure 2.9.12 shows the program example.
- When switching between the modes, set the frequency under the condition of $f(X_{IN}) > 3f(X_{CIN})$.

Table 2.9.1 Clock switch combination

Recommended transition process	
Low-speed→High-speed	Middle-speed→High-speed
Low-speed→Middle-speed	Middle-speed→Middle-speed
Double-speed→High-speed	Middle-speed→Low-speed
Double-speed→Middle-speed	High-speed→Double-speed
Double-speed→Low-speed	High-speed→MIddle-speed
	High-speed→Low-speed

```
Low-speed mode → Middle/High-speed mode → Double-speed mode switch

LDM  xx, CPUM  •••Low-speed mode → Middle/High-speed mode switch
NOP          )     Clock switch timing wait
NOP          )     (1 to 4 machine cycles are required for switching mode.)
LDM  yy, CPUM  •••Switch mode to double-speed

     Note: CPUM = CPU mode register (address 003B16)
```

Fig. 2.9.12 Program example

- Use the LDM, STA, etc. instructions to modify the division ratio of the internal clock ϕ. (Do not use read-modify-write instructions such as CLB, SEB, etc.)
- When oscillation of the main clock X_{IN} is restarted, after setting the main clock stop bit to “0” sufficient time to stabilize oscillation by program.

(2) In low-speed mode

- The sub-clock X_{CIN}-X_{COUT} oscillating circuit can not use input clocks externally. Accordingly, make sure to use an external resonator.

(3) In stop mode

- Set the interrupt enable bits of timer 1 and timer 2 to the disabled state (“0”) before executing the STP instruction.
- Oscillation restarts at reset or when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU until timer 2 underflows. It is because that retains time to stabilize oscillation using a ceramic resonator, etc. Set values for stabilizing oscillation to the timer 1 latch and the timer 2 latch before executing the STP instruction.
- When using the external interrupt input pin sharing with an I/O port, set “0” (input mode) to the direction register of a pin to be used before execution of the STP instruction.
- When using an interrupt for release from the stop mode, set the interrupt to the enable state before execution of the STP instruction.

(4) In wait mode

- When using an interrupt for release from the wait mode, set the interrupt to the enable state before execution of the WIT instruction.
- When using the external interrupt input pin sharing with an I/O port, set “0” (input mode) to the direction register of a pin to be used before execution of the WIT instruction.

2.10 Application circuit example

Figure 2.10.1 shows the application example to a low-speed LAN bus.

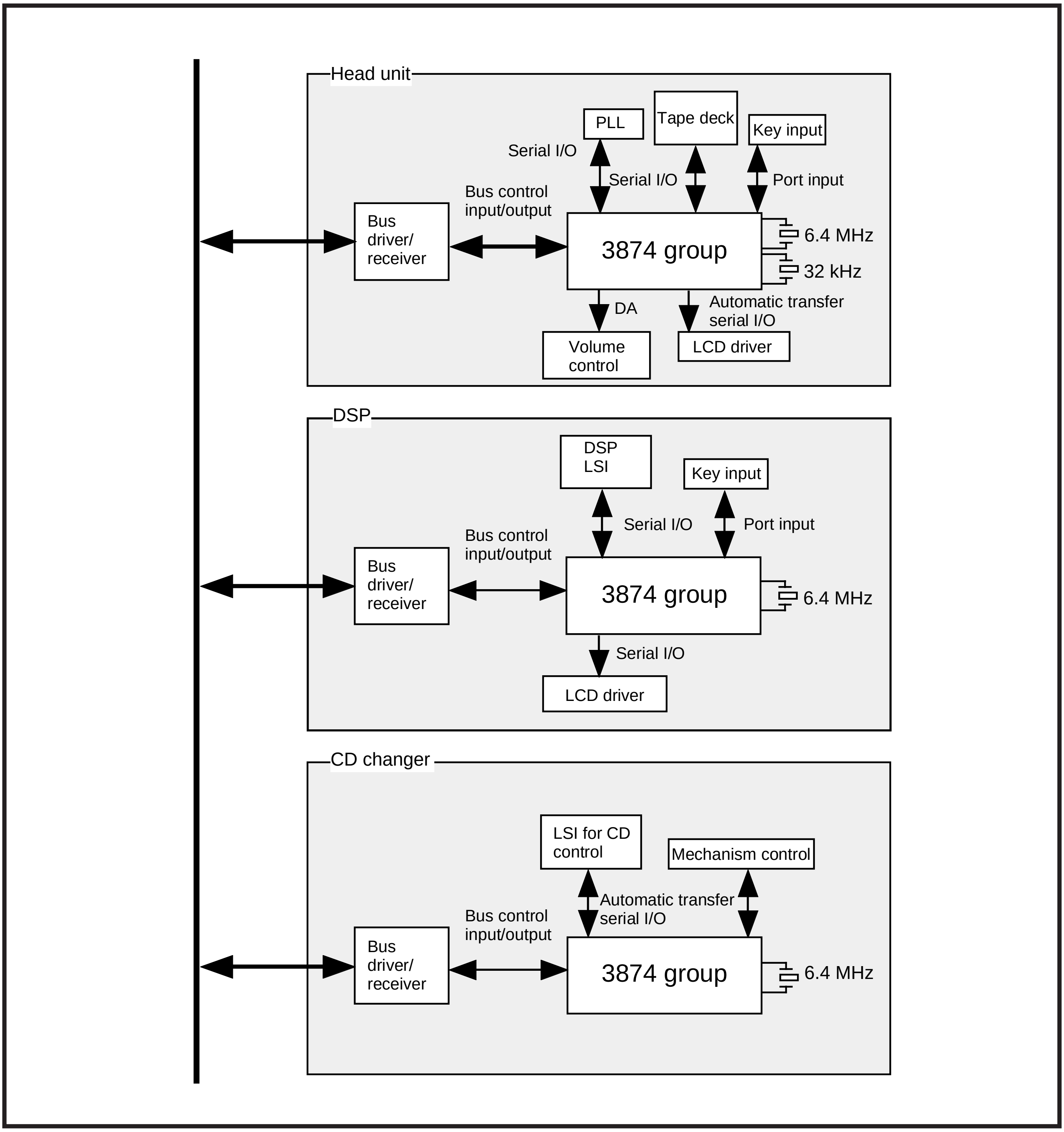


Fig. 2.10.1 Application circuit example to low-speed LAN bus

MEMORANDUM

CHAPTER 3
APPENDIX

3.1 Electrical characteristics
3.2 Standard characteristics
3.3 Notes on programming
3.4 Countermeasures against noise
3.5 Control registers
3.6 Package outline
3.7 List of instruction codes
3.8 Machine instructions
3.9 SFR allocation
3.10 Pin configuration

3.1 Electrical characteristics

Table 3.1.1 Absolute maximum ratings

(extended operating temperature version and automotive version)

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltage	All voltages are based on Vss. Output transistors are cut off.	–0.3 to 7.0	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, V_{REF}		–0.3 to Vcc +0.3	V
V_I	Input voltage $\overline{\text{RESET}}$, X_{IN}		–0.3 to Vcc +0.3	V
V_I	Input voltage $P9_7$		–0.3 to Vcc +0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, X_{OUT}		–0.3 to Vcc +0.3	V
P_d	Power dissipation	Ta = 25°C	500	mW
T_{opr}	Operating temperature		–40 to 85	°C
T_{stg}	Storage temperature		–60 to 150	°C

Table 3.1.2 Recommended operating conditions

(extended operating temperature version and automotive version, Vcc = 3.0 to 5.5 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter		Power source voltage			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage	At operating data link layer communication control circuit	4.0	5.0	5.5	V
		Double-speed mode	4.0	5.0	5.5	V
		High-speed mode	4.0	5.0	5.5	V
		Middle-speed mode	3.0	5.0	5.5	V
		Low-speed mode	3.0	5.0	5.5	V
V_{SS}	Power source voltage			0		V
V_{REF}	Analog reference voltage (when A-D converter is used)		2.0		V_{CC}	V
	Analog reference voltage (when D-A converter is used)		3.0		V_{CC}	V
AV_{SS}	Analog power source voltage			0		V
V_{IA}	Analog input voltage AN_0 to AN_7		AV_{SS}		V_{CC}	V
V_{IH}	"H" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_7$		$0.8V_{CC}$		V_{CC}	V
V_{IH}	"H" input voltage $\overline{\text{RESET}}$, X_{IN}		$0.8V_{CC}$		V_{CC}	V
V_{IL}	"L" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_7$		0		$0.2V_{CC}$	V
V_{IL}	"L" input voltage $\overline{\text{RESET}}$		0		$0.2V_{CC}$	V
V_{IL}	"L" input voltage X_{IN}		0		$0.16V_{CC}$	V

Table 3.1.3 Recommended operating conditions (1)

(extended operating temperature version and automotive version, Vcc = 3.0 to 5.5 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$\Sigma I_{OH(peak)}$	"H" total peak output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$			–80	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$			–80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$			80	mA
$\Sigma I_{OH(avg)}$	"H" total average output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$			–40	mA
$\Sigma I_{OH(avg)}$	"H" total average output current $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$			–40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$			40	mA
$I_{OH(peak)}$	"H" peak output current **(Note 2)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			–10	mA
$I_{OL(peak)}$	"L" peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			10	mA
$I_{OH(avg)}$	"H" average output current **(Note 3)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			–5.0	mA
$I_{OL(avg)}$	"L" average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			5.0	mA
$f(CNTR_0)$ $f(CNTR_1)$	Timer X, timer Y input oscillation frequency (at duty cycle of 50%)			2.5	MHz
$f(X_{IN})$	Main clock input oscillation frequency **(Note 4)**			6.4	MHz
$f(X_{CIN})$	Sub-clock input oscillation frequency **(Notes 4, 5)**		32.768	50	kHz

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 $I_{OL(avg)}$, $I_{OH(avg)}$ in an average value measured over 100 ms.

4: Choose an external oscillator which ensures no warps in the oscillation waveform as well as sufficient amplitude for the main clock oscillation circuit. Use according to the manufacturer's recommended conditions.

Table 3.1.4 Recommended operating conditions (2) (EPROM or One Time PROM version)

(Vcc = 3.0 to 5.5 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$f(X_{IN})$	Main clock input oscillation frequency	High-speed mode/Middle-speed mode			6.4	MHz
		Double-speed mode ($4.0 \leq V_{CC} < 4.5V$)			$2.8V_{CC}$–6.2	MHz
		Double-speed mode ($4.5 \leq V_{CC} \leq 5.5V$)			6.4	MHz

Note 5: When using the microcomputer in the low-speed mode, set the sub-clock input oscillation frequency on condition that $f(X_{CIN}) < f(X_{IN})/3$.

Table 3.1.5 Electrical characteristics

(extended operating temperature version and automotive version, Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	"H" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P8_0$–$P8_7$ **(Note)**	I_{OH} = –10 mA V_{CC} = 4.0–5.5 V	V_{CC}–2.0			V
		I_{OH} = –1 mA V_{CC} = 3.0–5.5 V	V_{CC}–1.0			V
V_{OL}	"L" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	I_{OL} = 10 mA V_{CC} = 4.0–5.5 V			2.0	V
		I_{OL} = 1.0 mA V_{CC} = 3.0–5.5 V			1.0	V
V_T^+–V_T^-	Hysteresis INT_0–INT_5, ADT, $CNTR_0$, $CNTR_1$			0.5		V
V_T^+–V_T^-	Hysteresis RxD, S_{CLK1}, S_{IN2}, S_{CLK2}, $P2_0$–$P2_7$	Valid hysteresis only when these pins is used as the function		0.5		V
V_T^+–V_T^-	Hysteresis $\overline{RESET}$			0.5		V
I_{IH}	"H" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	V_I = V_{CC}			5.0	µA
I_{IH}	"H" input current $P9_7$	V_I = V_{CC}			5.0	µA
I_{IH}	"H" input current $\overline{RESET}$	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 $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	V_I = V_{SS}			–5.0	µA
I_{IL}	"L" input current $P9_7$	V_I = V_{SS}			–5.0	µA
I_{IL}	"L" input current $\overline{RESET}$	V_I = V_{SS}			–5.0	µA
I_{IL}	"L" input current X_{IN}	V_I = V_{SS}		–4.0		µA
V_{RAM}	RAM hold voltage	When clock stopped	2.0		5.5	V

Note: When $P4_5$/TxD, $P7_1$/S_{OUT2}, and $P7_2$/S_{CLK2} are CMOS output states (when not P-channel output disable states)

Table 3.1.6 Electrical characteristics

(extended operating temperature version and automotive version, Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
ICC	Power source current	Double-speed mode, at operating data link layer communication control circuit f(XIN) = 6.29 MHz f(XCIN) = 32 kHz Output transistors “off” During A-D conversion			18.0	24.0	mA
		Double-speed mode, at stopping data link layer communication control circuit f(XIN) = 6.29 MHz f(XCIN) = 32 kHz Output transistors “off” During A-D conversion			12.0	18.0	mA
		Double-speed mode, at stopping data link layer communication control circuit f(XIN) = 6.29 MHz (in WIT state) f(XCIN) = 32 kHz Output transistors “off” During A-D conversion			2.0	3.5	mA
		High-speed mode, at operating data link layer communication control circuit f(XIN) = 6.29 MHz f(XCIN) = 32 kHz Output transistors “off” During A-D conversion			12.0	19.0	mA
		High-speed mode, at stopping data link layer communication control circuit f(XIN) = 6.29 MHz f(XCIN) = 32 kHz Output transistors “off” During A-D conversion			8.0	12.0	mA
		High-speed mode, at stopping data link layer communication control circuit f(XIN) = 6.29 MHz (in WIT state) f(XCIN) = 32 kHz Output transistors “off” During A-D conversion			2.0	3.5	mA
		Low-speed mode (VCC = 3.0 V) f(XIN) = stopped f(XCIN) = 32 kHz Low power dissipation mode (CM 5 = 0) Output transistors “off”			60	200	µA
		Low-speed mode (VCC = 3.0 V) f(XIN) = stopped f(XCIN) = 32 kHz (in WIT state) Low power dissipation mode (CM5 = 0) Output transistors “off”			20	40	µA
		All oscillation stopped (in STP state) Output transistors “off”	Ta = 25 °C **(Note)**		0.1	1.0	µA
			Ta = 85 °C **(Note)**			10	µA

Note: The A-D conversion is inactive. (The A–D conversion complete.) VREF current is not included.

3.1 Electrical characteristics

Table 3.1.7 A-D converter characteristics

(extended operating temperature version and automotive version, V_{CC} = 4.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, V_{REF} = 2.0 V to V_{CC}, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy (excluding quantization error)			±1	±2.5	LSB
t_{CONV}	Conversion time				50	tc(ϕ)
R_{LADDER}	Ladder resistor		12	35	100	kΩ
I_{VREF}	Reference power source input current	V_{REF} = 5.0 V	50	150	200	µA
$I_{I(AD)}$	Analog port input current			0.5	5.0	µA

Table 3.1.8 D-A converter characteristics

(extended operating temperature version and automotive version, V_{CC} = 4.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, V_{REF} = 2.0 V to V_{CC}, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
tsu	Setting time				3.0	µs
RO	Output resistor		1	2.5	4.0	kΩ
I_{VREF}	Reference power source input current				3.2	mA

TIMING REQUIREMENTS

Table 3.1.9 Timing requirements

(extended operating temperature version and automotive version, V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W}(\overline{RESET})$	Reset input “L” pulse width	2			µs
$t_{C}(X_{IN})$	External clock input cycle time	159			ns
$t_{WH}(X_{IN})$	External clock input “H” pulse width	63			ns
$t_{WL}(X_{IN})$	External clock input “L” pulse width	63			ns
$t_{C}(CNTR)$	CNTR$_0$, CNTR$_1$ input cycle time	200			ns
$t_{WH}(CNTR)$	CNTR$_0$, CNTR$_1$ input “H” pulse width	80			ns
$t_{WH}(INT)$	INT$_0$ to INT$_5$ input “H” pulse width	80			ns
$t_{WL}(CNTR)$	CNTR$_0$, CNTR$_1$ input “L” pulse width	80			ns
$t_{WL}(INT)$	INT$_0$ to INT$_5$ input “L” pulse width	80			ns
$t_{C}(S_{CLK1})$	Serial I/O1 clock input cycle time **(Note)**	800			ns
$t_{C}(S_{CLK2})$	Serial I/O2 clock input cycle time	1000			ns
$t_{C}(S_{CLK3})$	Serial I/O3 clock input cycle time	1000			ns
$t_{WH}(S_{CLK1})$	Serial I/O1 clock input “H” pulse width **(Note)**	370			ns
$t_{WH}(S_{CLK2})$	Serial I/O2 clock input “H” pulse width	400			ns
$t_{WH}(S_{CLK3})$	Serial I/O3 clock input “H” pulse width	400			ns
$t_{WL}(S_{CLK1})$	Serial I/O1 clock input “L” pulse width **(Note)**	370			ns
$t_{WL}(S_{CLK2})$	Serial I/O2 clock input “L” pulse width	400			ns
$t_{WL}(S_{CLK3})$	Serial I/O3 clock input “L” pulse width	400			ns
$t_{su}(R_{X}D\text{-}S_{CLK1})$	Serial I/O1 input setup time	220			ns
$t_{su}(S_{IN2}\text{-}S_{CLK2})$	Serial I/O2 input setup time	200			ns
$t_{su}(R_{IN3}\text{-}S_{CLK3})$	Serial I/O3 input setup time	200			ns
$t_{h}(S_{CLK1}\text{-}R_{X}D)$	Serial I/O1 input hold time	100			ns
$t_{h}(S_{CLK2}\text{-}S_{IN2})$	Serial I/O2 input hold time	200			ns
$t_{h}(S_{CLK3}\text{-}S_{IN3})$	Serial I/O3 input hold time	200			ns

Note : When bit 6 of address 001A$_{16}$ is “1” (clock synchronous).
Divide this value by four when bit 6 of address 001A$_{16}$ is “0” (UART).

Table 3.1.10 Switching characteristics

(extended operating temperature version and automotive version, V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{WH} (S_{CLK1})	Serial I/O1 clock output "H" pulse width	t_C(S_{CLK1})/2–30			ns
t_{WH} (S_{CLK2})	Serial I/O2 clock output "H" pulse width **(Note 1)**	t_C(S_{CLK2})/2–30			ns
t_{WH} (S_{CLK3})	Serial I/O3 clock output "H" pulse width **(Note 5)**	t_C(S_{CLK3})/2–30			ns
t_{WL} (S_{CLK1})	Serial I/O1 clock output "L" pulse width	t_C(S_{CLK1})/2–30			ns
t_{WL} (S_{CLK2})	Serial I/O2 clock output "L" pulse width **(Note 1)**	t_C(S_{CLK2})/2–30			ns
t_{WL} (S_{CLK3})	Serial I/O3 clock output "L" pulse width **(Note 5)**	t_C(S_{CLK3})/2–30			ns
t_d (S_{CLK1}-T_XD)	Serial I/O1 output delay time **(Note 3)**			140	ns
t_d (S_{CLK2}-S_{OUT2})	Serial I/O2 output delay time **(Notes 1, 2)**			140	ns
t_d (S_{CLK3}-S_{OUT3})	Serial I/O3 output delay time **(Notes 5, 6)**			140	ns
t_V (S_{CLK1}-T_XD)	Serial I/O1 output valid time **(Note 3)**	–30			ns
t_V (S_{CLK2}-S_{OUT2})	Serial I/O2 output valid time **(Notes 1, 2)**	0			ns
t_V (S_{CLK3}-S_{OUT3})	Serial I/O3 output valid time **(Notes 5, 6)**	0			ns
t_r (S_{CLK1})	Serial I/O1 clock output rising time		10	30	ns
t_f (S_{CLK1})	Serial I/O1 clock output falling time		10	30	ns
t_r (S_{CLK2})	Serial I/O2 clock output rising time **(Note 1)**		10	30	ns
t_f (S_{CLK2})	Serial I/O2 clock output falling time **(Note 1)**		10	30	ns
t_r (S_{CLK3})	Serial I/O3 clock output rising time **(Note 5)**		10	30	ns
t_f (S_{CLK3})	Serial I/O3 clock output falling time **(Note 5)**		10	30	ns
t_r (CMOS)	CMOS output rising time **(Note 4)**		10	30	ns
t_f (CMOS)	CMOS output falling time **(Note 4)**		10	30	ns

Notes 1: When P7$_2$/S_{CLK2} is CMOS output.
2: When P7$_1$/S_{OUT2} is CMOS output.
3: When P4$_5$/T_XD is CMOS output.
4: The X_{OUT} pin is excluded.
5: When P8$_4$/S_{CLK3} is CMOS output.
6: When P8$_2$/S_{OUT3} is CMOS output.

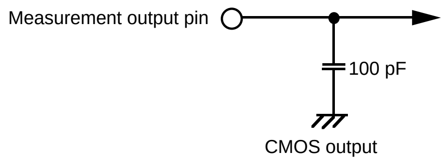


Fig. 3.1.1 Circuit for measuring output switching characteristics

Timing diagram

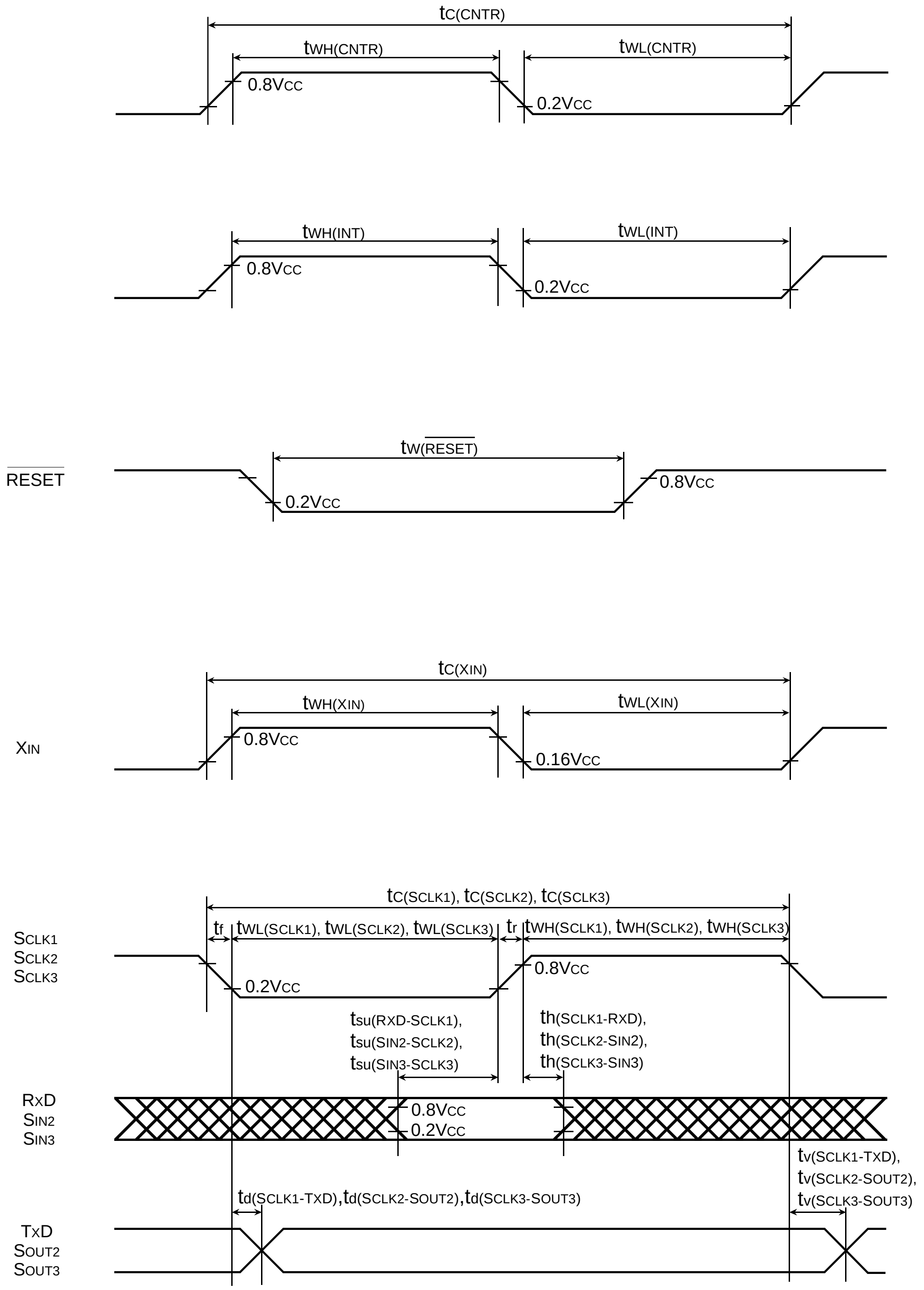


Fig. 3.1.2 Timing diagram (in single-chip mode)

3.2 Standard characteristics

3.2.1 Power source current standard characteristics (Icc–Vcc characteristics)

Figures 3.2.1 to 3.2.3 show the Icc – Vcc characteristics.

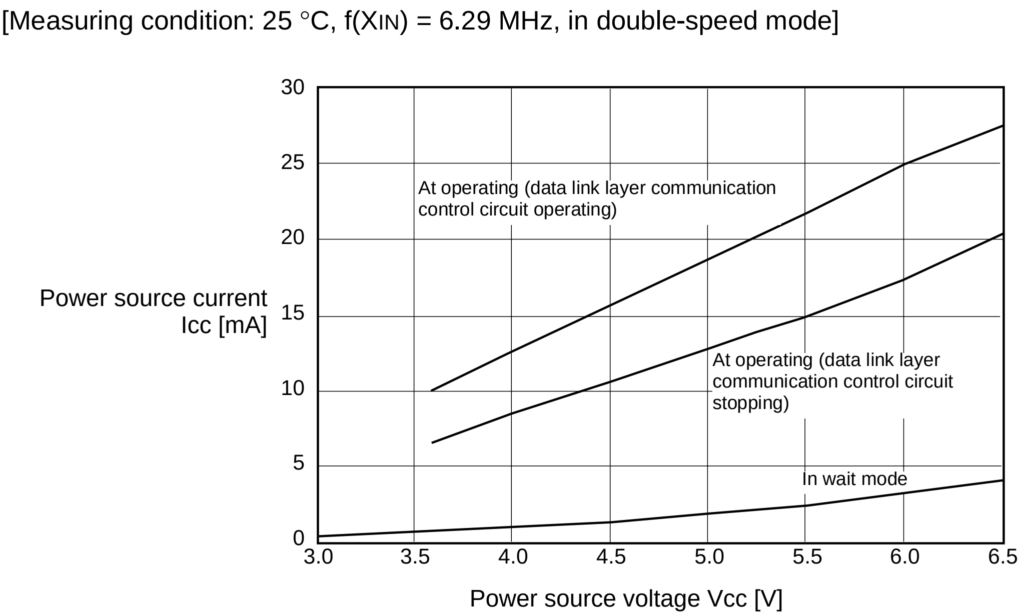


Fig. 3.2.1 Icc – Vcc characteristics (in double-speed mode)

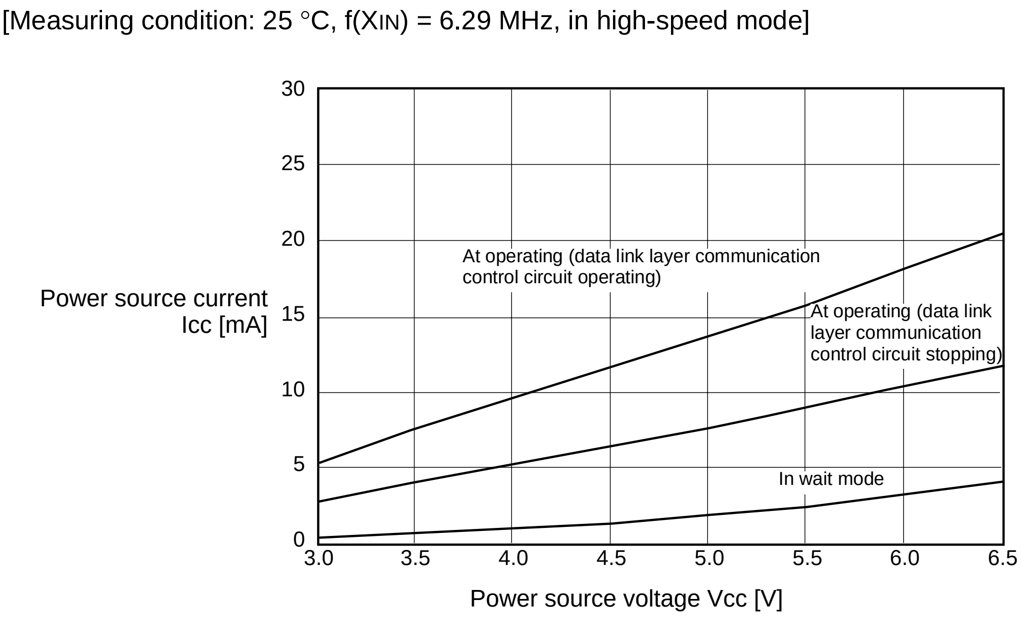


Fig. 3.2.2 Icc – Vcc characteristics (in high-speed mode)

3.2 Standard characteristics

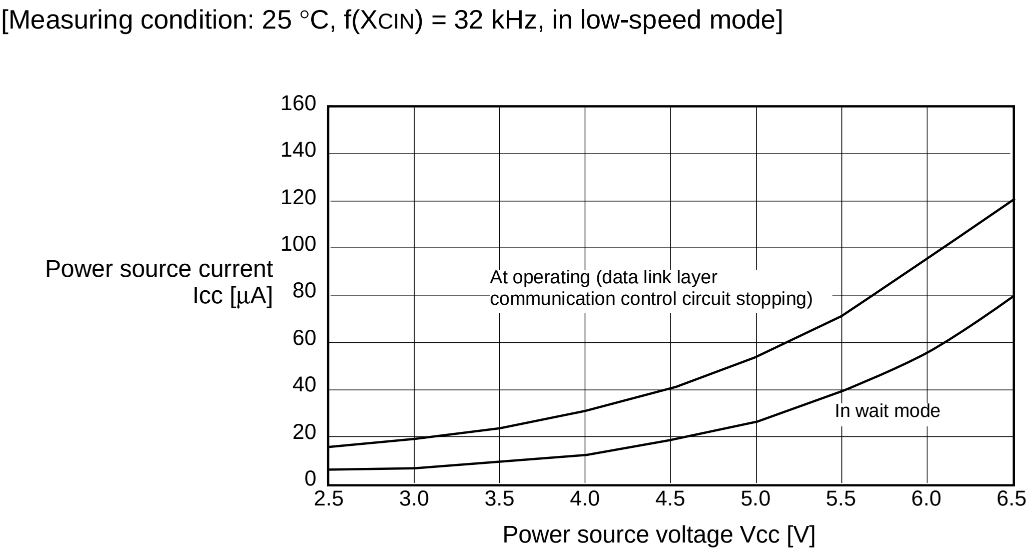


Fig. 3.2.3 Icc – Vcc characteristics (in low-speed mode)

3.2.2 Current dissipation frequency characteristics (Icc–f(X_{IN}) characteristics)

Figures 3.2.4 and 3.2.5 show the Icc – f(X_{IN}) characteristics.

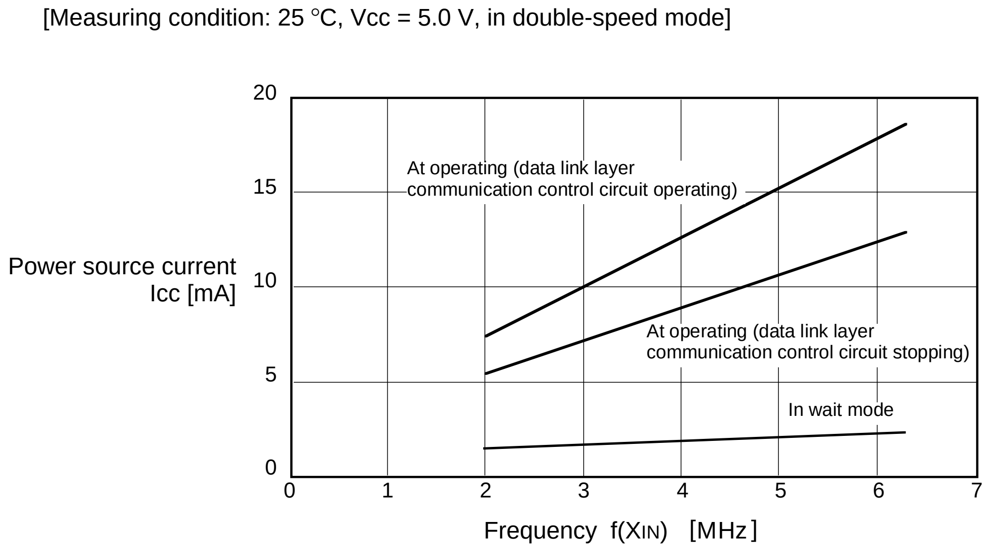


Fig. 3.2.4 Icc – f(X_{IN}) characteristics (in double-speed mode)

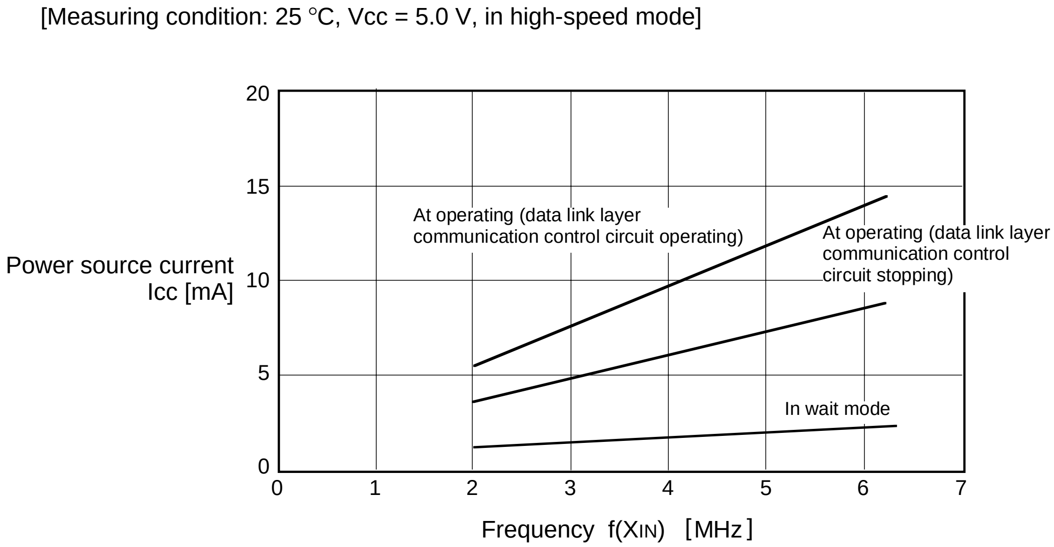


Fig. 3.2.5 Icc – f(X_{IN}) characteristics (in high-speed mode)

3.2 Standard characteristics

3.2.3 Port standard characteristics

Figures 3.2.6 to 3.2.8 show the port standard characteristics.

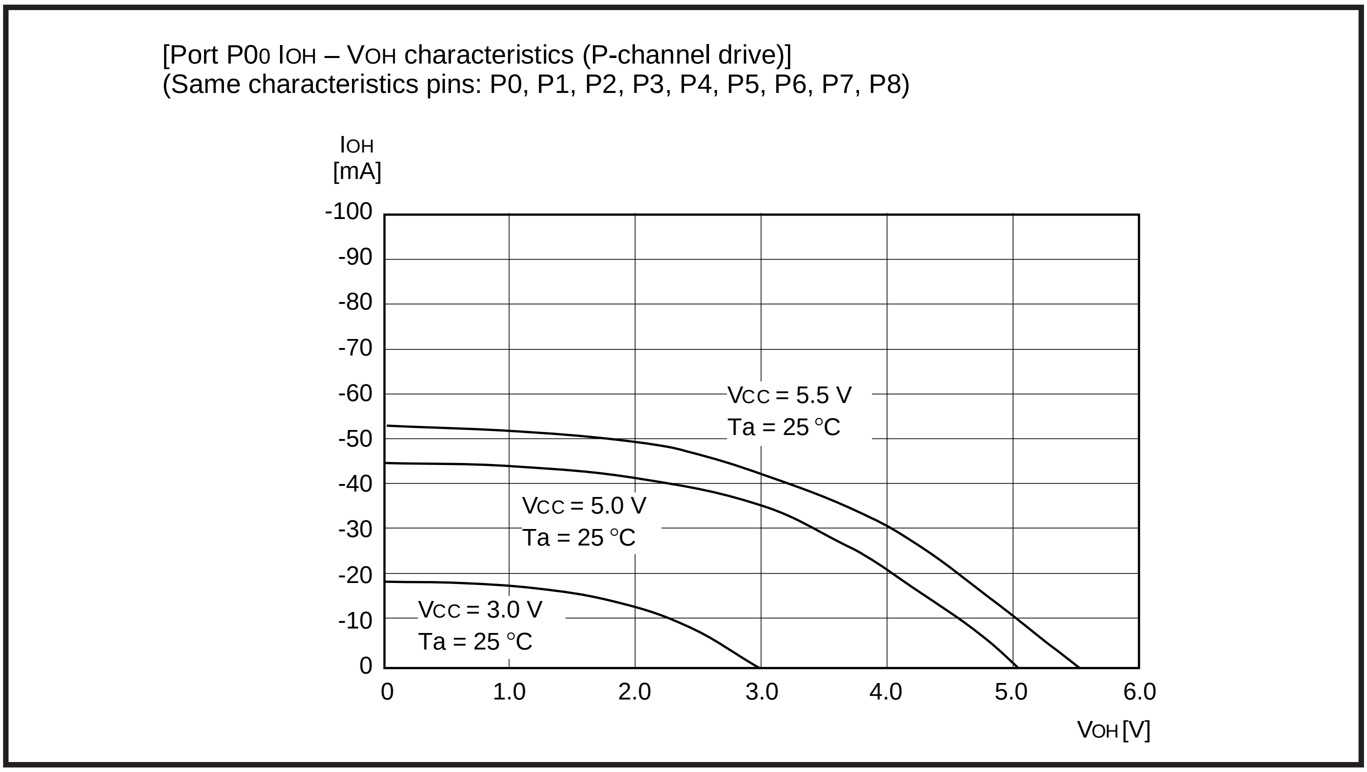


Fig. 3.2.6 Programmable I/O port (CMOS output) P-channel side I_{OH} – V_{OH} characteristics

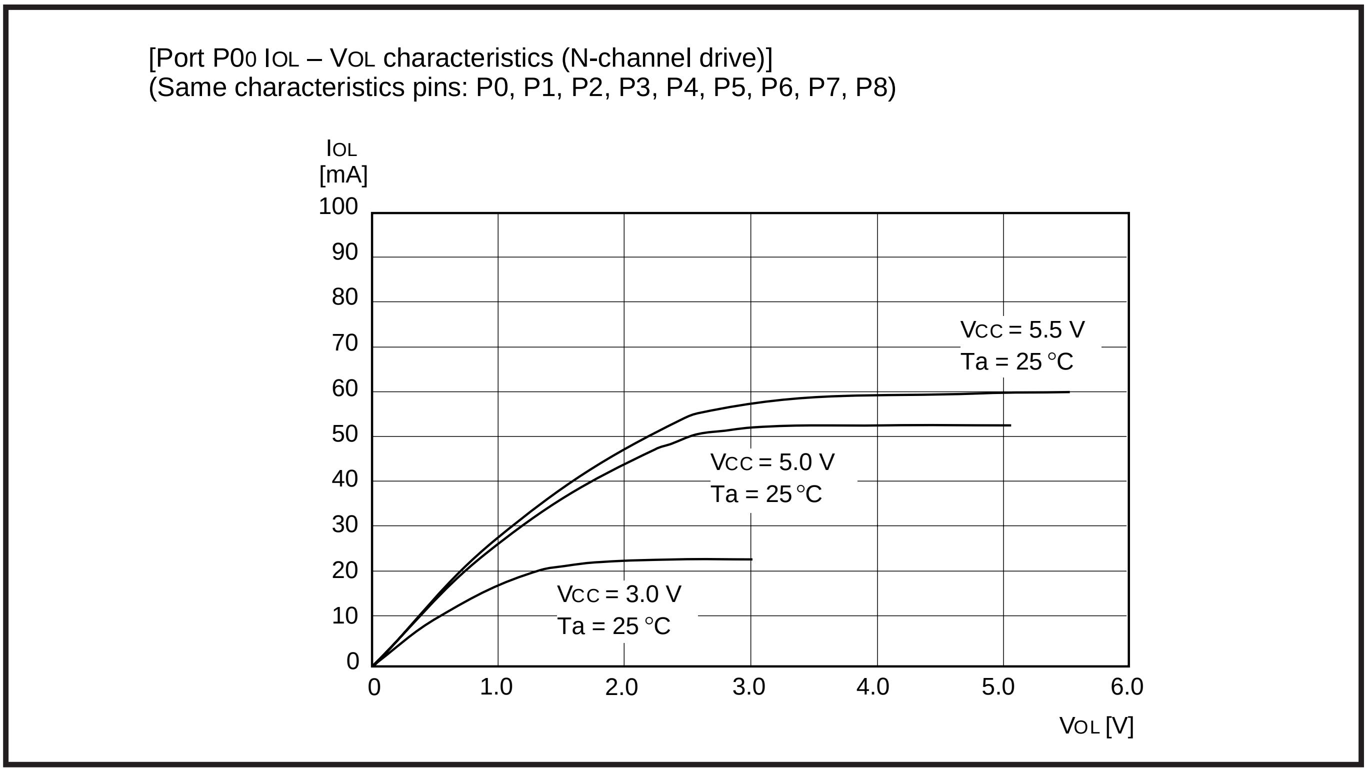


Fig. 3.2.7 Programmable I/O port (CMOS output) N-channel side I_{OL} – V_{OL} characteristics

[Port P2₀ I_{IL} – V_I]
(Same characteristics pin: P2)

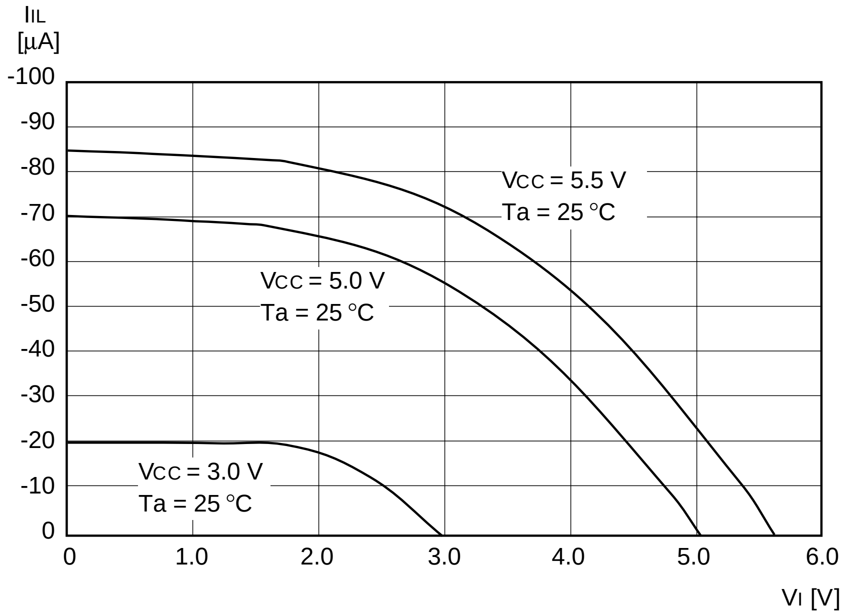


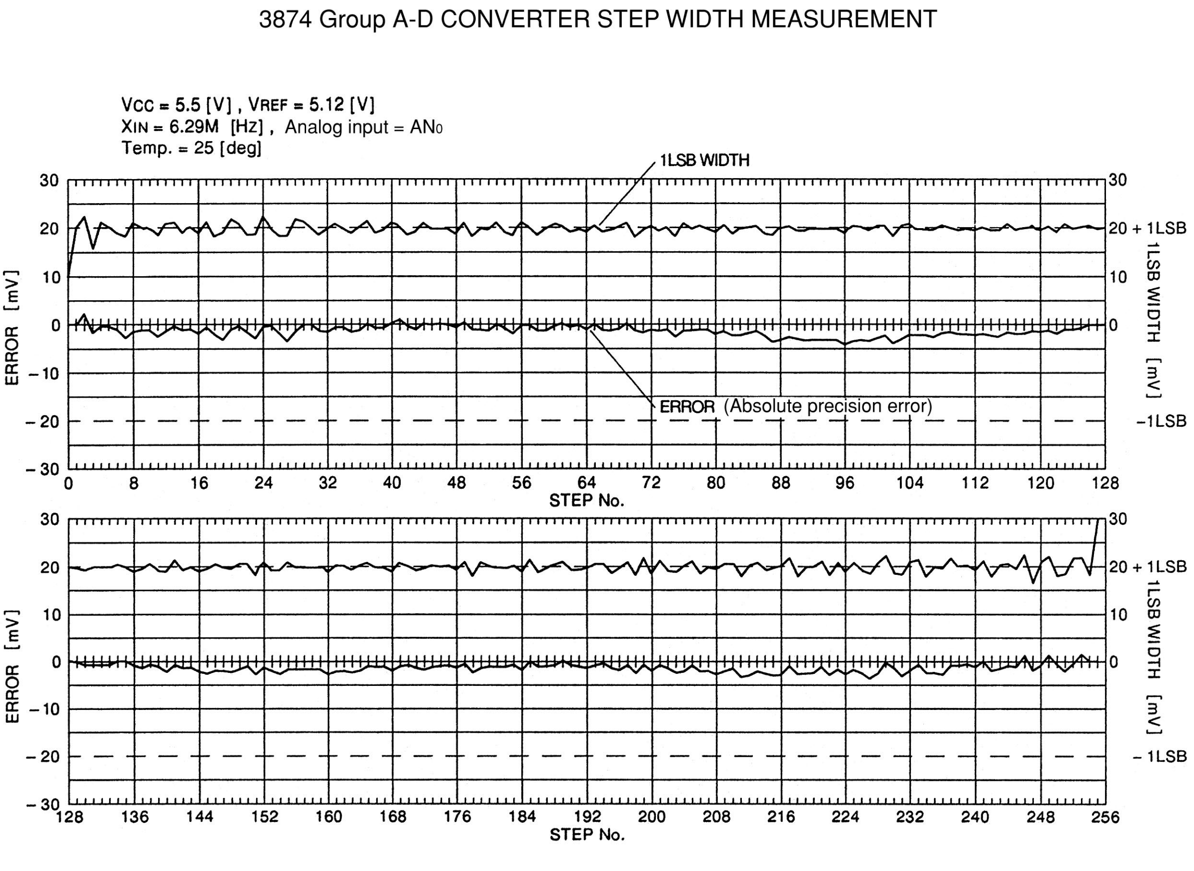
Fig. 3.2.8 Programmable I/O port (CMOS output) I_{IL} – V_I characteristics when connecting pull-up transistor

3.2.4 A-D conversion standard characteristics

Figure 3.2.9 shows the A-D conversion standard characteristics.

The lower line on the graph indicates the absolute precision error. It expresses the deviation from the ideal value. For example, the conversion of output code from 0 to 1 occurs ideally at the point of AN0 = 10 mV, but the measured value is 0 mV. Accordingly, the measured point of conversion is defined as “10 – 0 = 10 mV”.

The upper line on the graph indicates the width of input voltages equivalent to output codes. For example, the measured width of the input voltage for output code 13 is 22 mV, so that the differential nonlinear error is defined as “22 – 20 = 2 mV (0.1 LSB)”.



Note: This is measured when the power source voltage is stabilize in the high-speed mode.

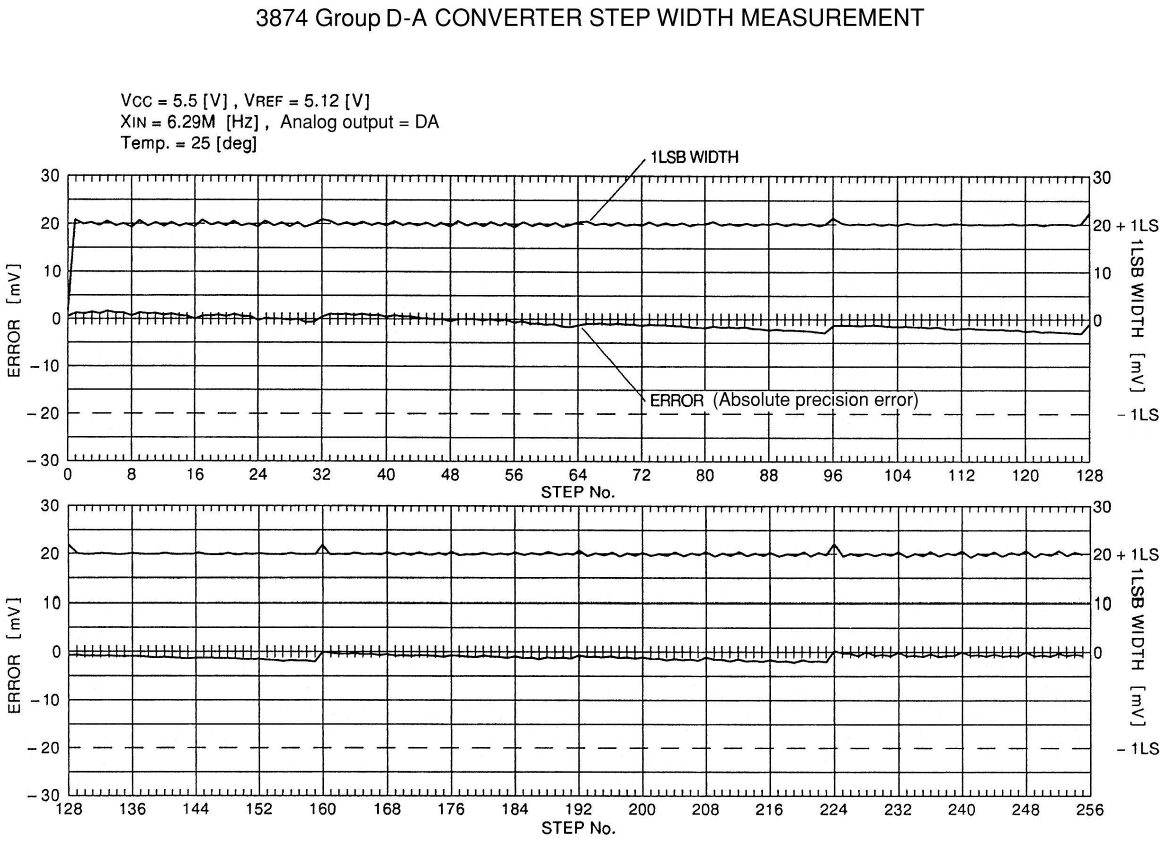
Fig. 3.2.9 A-D conversion standard characteristics

3.2.5 D-A conversion standard characteristics

Figure 3.2.10 shows the D-A conversion standard characteristics.

The lower line on the graph indicates the absolute precision error. That expresses the difference between the ideal analog output value for an input code and the measured value.

The upper line on the graph indicates the change width of output analog value toward a one-bit change of input code.



Note: This is measured when the power source voltage is stabilize in the high-speed mode.

Fig. 3.2.10 D-A conversion standard characteristics

3.3 Notes on use

3.3.1 Notes on interrupts

(1) Switching external interrupt detection edge

For the products able to switch the external interrupt detection edge, switch it as the following sequence.

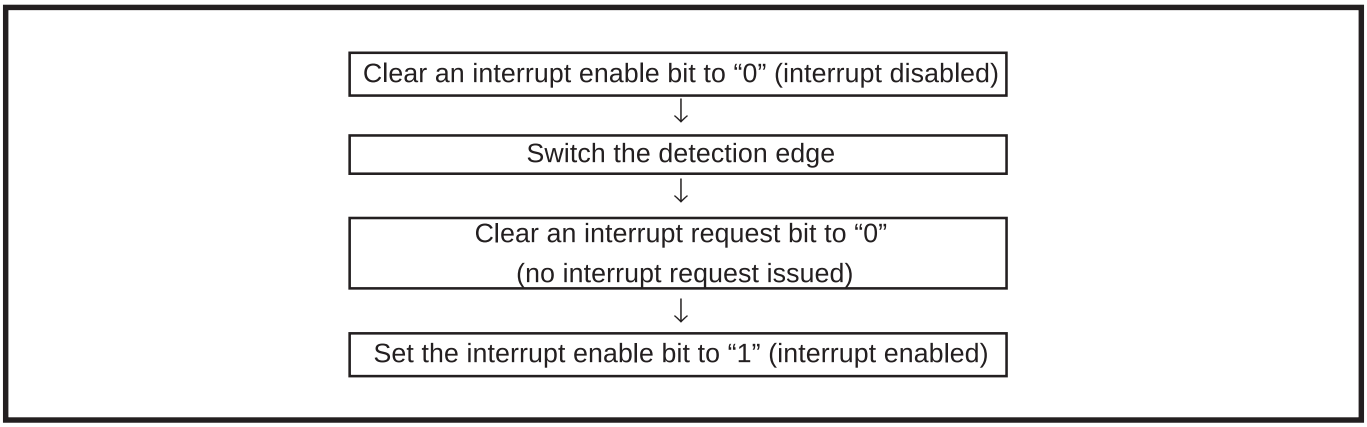

Clear an interrupt enable bit to “0” (interrupt disabled)
Switch the detection edge
Clear an interrupt request bit to “0”
(no interrupt request issued)
Set the interrupt enable bit to “1” (interrupt enabled)

Fig. 3.3.1 Sequence of switch detection edge

■ **Reason**

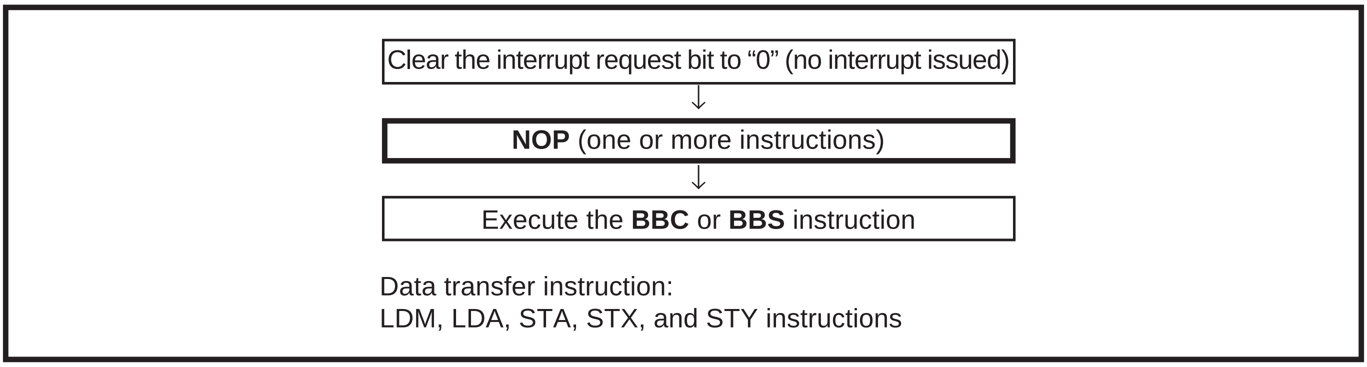
The interrupt circuit recognizes the switching of the detection edge as the change of external input signals. This may cause an unnecessary interrupt.

(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” by using a data transfer instruction, execute one or more instructions before executing the **BBC** or **BBS** instruction.

■ **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.


Clear the interrupt request bit to “0” (no interrupt issued)
NOP (one or more instructions)
Execute the BBC or BBS instruction

Data transfer instruction:
LDM, LDA, STA, STX, and STY instructions

Fig. 3.3.2 Sequence of check of interrupt request bit

- The request bits of interrupt source discrimination registers are not atuomatically cleared when an interrupt occurs. After an interrupt source has been discriminated and before execution of the RTI or CLI instruction; set “0” to the request bits of interrupt source discrimination registers by the user’s program.

(3) Setting of interrupt source discrimination registers

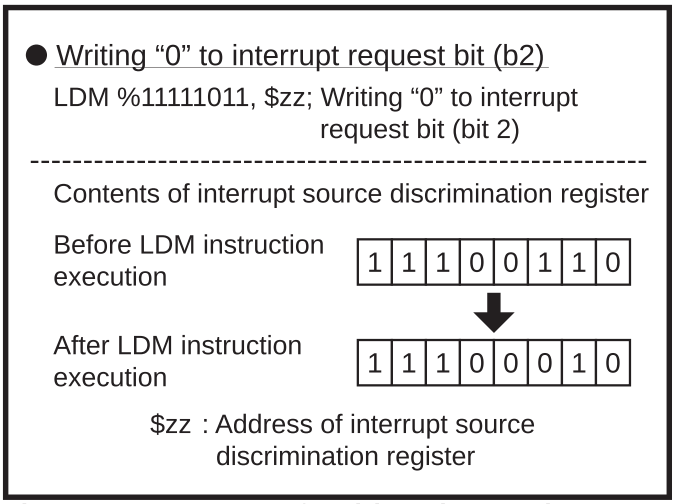
When “1” is written (or transferred) to an interrupt request bit, previous data is kept. Use the data transfer instructions✻1 to set “0” to the interrupt request bit of the interrupt source discrimination register without using the bit managing instructions✻2.

✻1 Data transfer instructions: LDM, LDA, STA, STX, and STY instructions
✻2 Bit managing instruction: CLB instruction

● **Reason**

If an interrupt request occurs at the same timing that the contents of the interrupt source discrimination register is modified by a bit managing instruction, the interrupt request bits of this register may not be set correctly.

Figure 3.3.3 shows an example of setting an interrupt request bit of an interrupt source discrimination.



● Writing “0” to interrupt request bit (b2)

```
LDM %11111011, $zz; Writing “0” to interrupt
                    request bit (bit 2)
```

Contents of interrupt source discrimination register

Before LDM instruction execution: 1 1 1 0 0 1 1 0

After LDM instruction execution: 1 1 1 0 0 0 1 0

$zz : Address of interrupt source discrimination register

Fig. 3.3.3 Example of writing data to interrupt source discrimination register

(4) Structure of interrupt control register 2

Fix the bit 7 of the interrupt control register 2 to “0”. Figure 3.3.4 shows the structure of the interrupt control register 2.

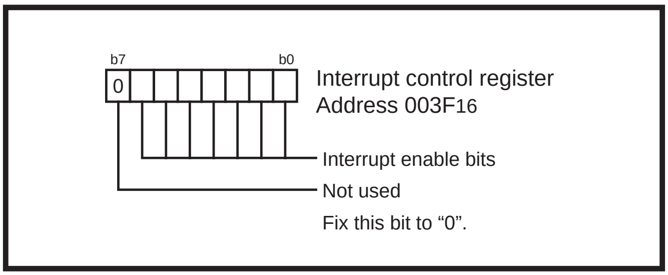


Fig. 3.3.4 Structure of interrupt control register 2

(5) Interrupt occurrence timing assigned to interrupt source discrimination registers

The interrupt (multiple factors / one vector interrupt) assigned to the interrupt source discrimination registers requires one instruction execution cycle (2 to 16 cycles of internal system clock) until starting the interrupt sequence more than that of “one factor / one vector interrupt”.

3.3.2 Notes on timer X

(1) Common to all modes

■ **Reading/Writing of timer X**

When reading or writing to timer X, be sure to execute to both timer X (high-order) and timer X (low-order). When reading a value from timer X, read it in order of timer X (high-order) and timer X (low-order) following. When writing a value to timer X, write in order of timer X (low-order) and timer X (high-order) following. If the following operations are performed to timer X, unexpected operation may occur.

•Write operation before execution of timer X (low-order) reading
•Read operation before execution of timer X (high-order) writing
•When writing for the latch only, if writing timing for the high-order latch is the almost same as the underflow timing, unexpected value may be set in the high-order counter.

■ **Division ratio of timer X**

When a value n (0 to $FFFF_{16}$) is written to timer X, the division ratio is 1/(n+1).

■ **Select of $CNTR_0$ interrupt active edge**

Setting the $CNTR_0$ active edge switch bit affects the active edge of an interrupt. Consequently, a $CNTR_0$ interrupt request may occur by setting the $CNTR_0$ active edge switch bit. As a countermeasure against the above, switch the active edge after disabling the $CNTR_0$ interrupt, and set “0” to the $CNTR_0$ interrupt request bit following.

(2) Pulse output mode

● Set “1” to bit 4 ($CNTR_0$ pin) of the port P5 direction register (output mode).
● When bit 4 ($CNTR_0$ pin) of the port P5 register is read, the value of the port register are not read out but the output value of the pin is read out.

(3) Event counter mode

● Set “0” to bit 4 ($CNTR_0$ pin) of the port P5 direction register (input mode).

(4) Pulse width measurement mode

● Set “0” to bit 4 ($CNTR_0$ pin) of the port P5 direction register (input mode).
● When reading bit 4 of port P5, the value is “1” at “H” level input or “0” at “L” level input regardless of the value of the $CNTR_0$ active edge switch bit of the timer X mode register.

(5) Real time port function

● Port $P5_6$ (RTP_0 pin) and port $P5_7$ (RTP_1 pin) function as a normal I/O port after reset released. When using ports $P5_6$ and $P5_7$ as real time port function pins, set “1” to the corresponding port direction register to be the output mode.
● Do not switch the pins which is used as the real time port to the input mode during operation.

3.3.3 Notes on timer Y

(1) Common to all modes

● When reading or writing to timer Y, be sure to execute to both timer Y (high-order) and timer Y (low-order).

● When reading a value from timer Y, read it in order of timer Y (high-order) and timer Y (low-order) following. When writing a value to timer Y, write in order of timer Y (low-order) and timer Y (high-order) following.

● If the following operations are performed to timer Y, expected operation may occur.
•Write operation before execution of timer Y (low-order) reading
•Read operation before execution of timer Y (high-order) writing

■ **Division ratio of timer Y**

When a value n (0 to $FFFF_{16}$) is written to timer Y, the division ratio is 1/(n+1).

■ **Select of CNTR1 interrupt active edge**

Setting the $CNTR_1$ active edge switch bit affects the active edge of an interrupt. Consequently, a $CNTR_1$ interrupt request may occur by setting the $CNTR_1$ active edge switch bit. As a countermeasure against the above, switch the active edge after disabling the $CNTR_1$ interrupt, and set “0” to the $CNTR_1$ interrupt request bit following.

(2) Period measurement mode

● Set bit 5 (corresponding to the $CNTR_1$ pin) of the port P5 direction register to “0” (input mode).

(3) Event counter mode

● Set bit 5 (corresponding to the $CNTR_1$ pin) of the port P5 direction register to “0” (input mode).

(4) Pulse width HL continuously measurement mode

● Set bit 5 (corresponding to the $CNTR_1$ pin) of the port P5 direction register to be the input mode.

● The $CNTR_1$ interrupt request occurs at both edges of a input pulse regardless of the contents of the $CNTR_1$ active edge switch bit of the timer Y mode register.

3.3.4 Notes on timers 1, 2, 3

(1) Notes on timer 1 to timer 3

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

● In case of the following, a short pulse occurs on counted input signals, so that the timer count value may change greatly. Accordingly, after setting their count sources, set values in order of timer 1, timer 2, and timer 3:
•When the count sources of timers 1 to 3 are switched
•When the timer 1 output signal is selected as a count source of timer 2 or timer 3 and data is written to timer 1

(2) Notes on timer 2

● When writing to the latch only is selected, the value written into timer 2 is written only in the latch for reloading and the contents of timer 2 is not changed. This written value is transferred to timer 2 at the first underflow after writing. Normally, a value is written in both the latch and timer at one time. That is, when a value is written to the timer, it is set in both the latch and the timer.

● In the T_{OUT} output enabled state, a signal whose polarity is reversed each time timer 2 underflows is output from the T_{OUT} pin. In this case, set port $P5_0$ (sharing with the T_{OUT} pin) to the output mode.

3.3.5 Notes on serial I/O1

(1) Notes when selecting clock synchronous serial I/O

① **Stop of transmission operation**

As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the serial I/O1 enable bit and the transmit enable bit to “0” (serial I/O1 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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, 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 TxD pin and an operation failure occurs.

② **Stop of receive operation**

As for the serial I/O1 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/O1 enable bit to “0” (serial I/O1 disabled).

③ **Stop of transmit/receive operation**

As for the serial I/O1 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/O1 enable bit to “0” (serial I/O1 disabled) (refer to (1) ①).

(2) Notes when selecting clock asynchronous serial I/O

① **Stop of transmission operation**
As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the transmit enable bit to “0” (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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK1, and $\overline{\mathrm{SRDY1}}$ 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 TxD pin and an operation failure occurs.

② **Stop of receive operation**
As for the serial I/O1 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).

③ **Stop of transmit/receive operation**
● **Only transmission operation is stopped**
As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the transmit enable bit to “0” (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 TxD, RxD, SCLK1, and $\overline{\mathrm{SRDY1}}$ 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/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

● **Only receive operation is stopped**
As for the serial I/O1 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).

(3) $\overline{\mathrm{SRDY1}}$ output of reception side
When signals are output from the $\overline{\mathrm{SRDY1}}$ 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 $\overline{\mathrm{SRDY1}}$ output enable bit, and the transmit enable bit to “1” (transmit enabled).

(4) Setting serial I/O control register again
Set the serial I/O 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.”

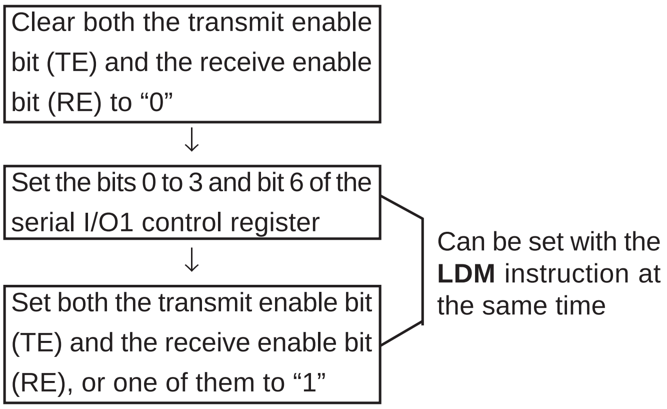

Clear both the transmit enable bit (TE) and the receive enable bit (RE) to “0”
Set the bits 0 to 3 and bit 6 of the serial I/O1 control register
Set both the transmit enable bit (TE) and the receive enable bit (RE), or one of them to “1”
Can be set with the **LDM** instruction at the same time

Fig. 3.3.5 Sequence of setting serial I/O1 control register again

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

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 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

The transmit interrupt request bit is set and the interrupt request occurs even when selecting timing that either of the following flags is set to “1” as timing where the transmit interrupt occurs.

- Transmit buffer empty flag is set to “1”
- Transmit shift register completion flag is set to “1”

Accordingly, when the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as the following sequence.

① Transmit enable bit is set to “1”

② Transmit interrupt request bit is set to “0”

● **Reason**

When the transmit enable bit is set to “1”, the transmit buffer empty flag and transmit shift register completion flag are set to “1”.

(8) Use of TxD pin

The $P4_5$/TxD channel output disable bit of the UART control register is valid both when using as a normal port and when using as the TxD pin. However, do not apply a voltage of Vcc + 0.3 V or more to the $P4_5$/TxD pin even when it is used as N-channel open drain output.

In the serial I/O1, after transmit is completed, the TxD pin continue latching and outputting the last bit.

3.3.6 Notes on serial I/O2

(1) When using external clock

● The serial I/O2 interrupt request bit is set to “1” by counting eight times of the transfer clock when the synchronous clock is the internal clock or the external clock. However, when using the external clock, the contents of the serial I/O2 register is being shifted continuously while the transfer clock is input to the serial I/O2’s circuit. Stop the transfer clock at 8 times. (When using the internal clock, the transfer clock automatically stops.)

● When using the external clock, the S_{OUT2} pin does not become the high-impedance state after the data transfer is completed. Set “1” to the S_{OUT2} output control bit of the serial I/O2 control register after the data transfer is completed.
When using the internal clock, the S_{OUT2} pin automatically becomes the high-impedance state after the data transfer is completed.

● When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLK2 input level. Also, write data to the serial I/O2 register at “H” of the SCLK2 input level.

3.3.7 Notes on serial I/O3

(1) In all modes

■ **State of S_{OUT3} pin**

● The S_{OUT3} output control bit of the serial I/O3 control register 2 can be used to select the state of the S_{OUT3} pin when serial data is not transferred; either output active or high-impedance. However, when selecting an external synchronous clock; the S_{OUT3} pin can become the high-impedance state by setting the S_{OUT3} output control bit to “1” when S_{CLK3} input is at “H” after transfer completion.

■ **Serial I/O initialization bit**

● Set “0” to the serial I/O initialization bit of the serial I/O3 control register 1 when terminating a serial transfer during transferring.

● When writing “1” to the serial I/O initialization bit, serial I/O3 is enabled, but each register is not initialized. Set the value of each register by program.

■ **Handshake signal**

● S_{BUSY3} input signal
Input an “H” level to the S_{BUSY3} input and an “L” level signal to the $\overline{S_{BUSY3}}$ input in the initial state. When the external synchronous clock is selected, switch the input level to the S_{BUSY3} input and the $\overline{S_{BUSY3}}$ input while the S_{CLK3} input is in “H” state.

● S_{RDY3} input•output signal
When selecting the internal synchronous clock, input an “L” level to the S_{RDY3} input and an “H” level signal to the $\overline{S_{RDY3}}$ input in the initial state.

(2) 8-bit serial I/O mode

■ **When selecting external synchronous clock**

When an external synchronous clock is selected, the contents of the serial I/O3 register are being shifted continually while the transfer clock is input to S_{CLK3}. In this case, control the clock externally.

(3) In automatic transfer serial I/O mode

■ **Set of automatic transfer interval**

- ● When the S_{BUSY3} output is used, and the S_{BUSY3} output and the S_{STB3} output function as signal for each transfer data set by the S_{BUSY3} output•S_{STB3} output function selection bit: the transfer interval is necessary before the first data is transmitted/received, and after the last data is transmitted/received. Accordingly, regardless of the contents of the S_{BUSY3} output•S_{STB3} output function selection bit, this transfer interval becomes 2 cycles longer than the value set for each 1-byte data.
- ● When using the S_{STB3} output, regardless of the contents of the S_{BUSY3} output•S_{STB3} output function selection bit, this transfer interval becomes 2 cycles longer than the value set by the automatic transfer interval set bit of the serial I/O3 control register 3 for each 1-byte data.
- ● When using the combined output of S_{BUSY3} and S_{STB3} as the signal for each transfer data set, the transfer interval after completion of transmission/reception of the last data becomes 2 cycles longer than the value set by the automatic transfer interval set bit.
- ● Set the transfer interval of each 1-byte data transmission to 5 or more cycles of the internal clock ϕ after the rising edge of the last bit of a 1-byte data.
- ● When selecting an external clock, the automatic transfer interval cannot be set.

■ **Set of serial I/O3 transfer counter**

- ● Write the value decremented by 1 from the number of transfer data bytes to the serial I/O3 transfer counter.
- ● When selecting an external clock, write a value to the serial I/O3 register/transfer counter, wait for 5 or more cycles of internal clock ϕ before inputting the transfer clock to the S_{CLK3} pin.

■ **Serial I/O initialization bit**

A serial I/O3 interrupt request occurs when “0” is written to the serial I/O initialization bit during an operation. Disable the interrupt enable bit as necessary by program.

(4) Arbitrary bit serial I/O mode

■ **Set of serial I/O3 transfer counter**

- ● Write the value decremented by 1 from the number of transfer data bytes to the serial I/O3 transfer counter.
- ● When selecting an external clock, write a value to the serial I/O3 register/transfer counter, wait for 5 or more cycles of internal clock ϕ before inputting the transfer clock to the S_{CLK3} pin.

■ **Set of automatic transfer interval**

- ● When selecting an external clock, the automatic interval cannot be set.
- ● When using the S_{BUSY3} output, the transfer interval is necessary before the first data is transmitted/received, and after the last data is transmitted/received. When using the S_{STB3} output, this transfer interval becomes 2 cycles longer than the value set for each 8-bit data. In addition, when using the combined output of S_{BUSY3} and S_{STB3}, the transfer interval after completion of transmission/reception of the last data becomes 2 cycles longer than the set value.

■ **Receive data**

If the last data does not fill 8 bits, the receive data stored in the serial I/O3 automatic transfer RAM becomes the closest MSB odd bit when the transfer direction select bit is set to LSB first, or the closest LSB odd bit when the transfer direction select bit is set to MSB first.

3.3.8 Notes on data link layer communication control circuit

(1) Start and mode switch of data link layer communication control circuit

● The data link layer communication control is stopped after reset. Write "00XXXXX1_2" to the communicatiton mode register to start the data link layer communication control circuit or switch the mode. Bits 4 and 5 are read exclusive bits.

(2) $P7_5/\overline{BUS_{OUT}}$ pin and $P7_6/\overline{BUS_{IN}}$ pin

● The $P7_5/\overline{BUS_{OUT}}$ pin and the $P7_6/\overline{BUS_{IN}}$ pin function as normal ports after reset released. When selecting as normal ports, these pins can be switched to either input or output by the direction register.

3.3.9 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 0.01μF to 1μF. 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

Pins AVss are A-D converter power source pins. Regardless of using the A-D conversion function or not, connect them as following :

• AVss : Connect to the Vss line

● **Reason**

If the AVss pin are opened, the microcomputer may have a failure because of noise or others.

(3) Reference voltage input pin V_{REF}

When using A-D converter or D-A converter, apply a voltage of 2 V to Vcc to the reference voltage input pin V_{REF}. Note that if the V_{REF} value is lowered, the accuracy degrades.

(4) 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 and **WIT** instruction

(5) A-D/D-A conversion register

The A-D/D-A conversion register functions as an A-D conversion register during a read and a D-A conversion during a write. Accordingly, the D-A conversion register set value cannot be read out.

(6) Inputting falling signal to ADT pin during A-D conversion when using external trigger

When inputting the falling signal to the ADT pin during A-D conversion, A-D conversion which is in progress is stopped, and A-D conversion is started again.

(7) Set analog input pin to input mode

When using the A-D converter, set port P6 pins used as the analog input pins to an input port. When using the D-A converter, set $P8_0$/DA pin as input port.

(8) Connecting D-A output pin to low-impedance load when using D-A converter

The DA output pin does not include a buffer, so that use an external buffer when connecting a peripheral circuit having a low-impedance load.

(9) Vcc when using D-A converter

The D-A converter accuracy when Vcc is 4.0 V or less differs from that of when Vcc is 4.0 V or more. When using the D-A converter, we recomend that Vcc is used at 4.0 V or more.

3.3.10 Notes on watchdog timer

- ● The watchdog timer continues to count even while waiting for Stop release. Accordingly, make sure that watchdog timer does not underflow during this term.
- ● Once a “1” is written to the STP instruction disable bit of the watchdog timer control register, it cannot be programmed to “0” again.

3.3.11 Notes on reset circuit

(1) Reset input voltage control

Make sure that the reset input voltage is 0.6 V or less for Vcc of 3.0 V.

(2) 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. And 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.

3.3.12 Notes on clock generating circuit

(1) In all mode

- Use the circuit constants recommended by the resonator manufacturer.
- No external resistor is needed between pins X_{IN} and X_{OUT} because a feed-back resistor is included. However, an external feed-back resistor is needed between pins X_{CIN} and X_{COUT}.
- Use the oscillation circuit of the XCIN side in the condition that the pull-up circuit between pins XCIN and XCOUT is valid.
- If switching the mode between low-speed and double-speed, switch the mode to middle/high-speed first, and then switch the mode to double-speed by program. Do not switch the mode from low-speed to double-speed directly. 1 to 4 machine cycles are required for switching from low-speed mode to other mdoe. Insert “clock switch timing wait” for switching the mode to middle/high-speed certainly, and then switch the mode to double-speed.

  Table 3.3.1 lists the recommended transition process for system clock switch.

  Figure 3.3.6 shows the program example.

Table 3.3.1 Clock switch combination

Recommended transition process	
Low-speed→High-speed	Middle-speed→High-speed
Low-speed→Middle-speed	Middle-speed→Middle-speed
Double-speed→High-speed	Middle-speed→Low-speed
Double-speed→Middle-speed	High-speed→Double-speed
Double-speed→Low-speed	High-speed→MIddle-speed
	High-speed→Low-speed

```
Low-speed mode → Middle/High-speed mode → Double-speed mode switch
   LDM  xx, CPUM  •••Low-speed mode → Middle/High-speed mode switch
   NOP          )      Clock switch timing wait
   NOP          )      (1 to 4 machine cycles are required for switching mode.)
   LDM  yy, CPUM  •••Switch mode to double-speed

          Note: CPUM = CPU mode register (address 003B16)
```

Fig. 3.3.6 Program example

- When switching between the modes, set the frequency under the condition of $f(X_{IN}) > 3f(X_{CIN})$.
- Use the LDM, STA, etc. instructions to modify the division ratio of the internal clock ϕ. (Do not use read-modify-write instructions such as CLB, SEB, etc.)
- When oscillation of the main clock X_{IN} is restarted, after setting the main clock stop bit to “0”, set sufficient time to stabilize oscillation by program.

(2) In low-speed mode

- The sub-clock X_{CIN}-X_{COUT} oscillating circuit can not use input clocks externally. Accordingly, make sure to use an external resonator.

(3) In stop mode

- Set the interrupt enable bits of timer 1 and timer 2 to the disabled state (“0”) before executing the STP instruction.
- Oscillation restarts at reset or when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU until timer 2 underflows. It is because that retains time to stabilize oscillation using a ceramic resonator etc. Set values for stabilizing oscillation to the timer 1 latch and the timer 2 latch before executing the STP instruction.
- When using the external interrupt input pin sharing with an I/O port, set “0” (input mode) to the direction register of a pin to be used before execution of the STP instruction.
- When using an interrupt for release from the stop mode, set the interrupt to the enable state before execution of the STP instruction.

(4) In wait mode

- When using an interrupt for release from the wait mode, set the interrupt to the enable state before execution of the WIT instruction.
- When using the external interrupt input pin sharing with an I/O port, set “0” (input mode) to the direction register of a pin to be used before execution of the WIT instruction.

3.3.13 Notes on input and output pins

(1) Notes in stand-by state

In stand-by state*1 for low-power dissipation, do not make input levels of an input port and an I/O port “undefined”, especially for I/O ports of the N-channel open-drain.

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**

Even when setting as an output port with its direction register, in the following state :

- P-channel......when the content of the port latch is “0”
- N-channel......when the content of the port latch is “1”

the transistor becomes the OFF state, which causes the ports to be the high-impedance state. Note that the level becomes “undefined” depending on external circuits.

Accordingly, the potential which is input to the input buffer in a microcomputer is unstable in the state that input levels of a input port and an I/O port are “undefined”. This may cause power source current.

*1 stand-by state : the stop mode by executing the **STP** instruction
the wait mode by executing the **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*2, the value of the unspecified bit may be changed.

● **Reason**

The bit managing instructions are read-modify-write form instructions for reading and writing data by a byte unit. Accordingly, when these instructions are executed on a bit of the port latch of an I/O port, the following is executed to all bits of the port latch.

- As for a bit which is set for an input port :
  The pin state is read in the CPU, and is written to this bit after bit managing.
- As for a bit which is set for an output port :
  The bit value of the port latch is read in the CPU, and is written to this bit after bit managing.

Note the following :

- Even when a port which is set as an output port is changed for an input port, its port latch holds the output data.
- As for a bit of the port latch which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its port latch contents.

*2 bit managing instructions : **SEB**, and **CLB** instructions

3.3.14 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”.

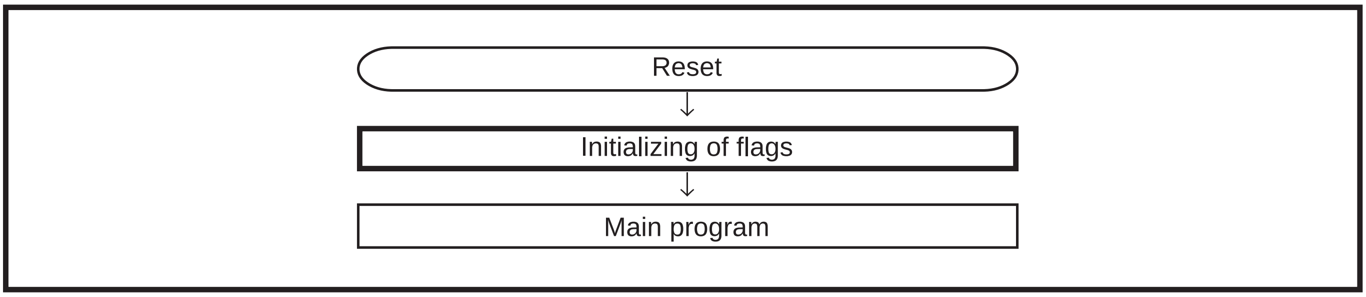


Fig. 3.3.7 Initialization of processor status register

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.

A **NOP** instruction should be executed after every **PLP** instruction.

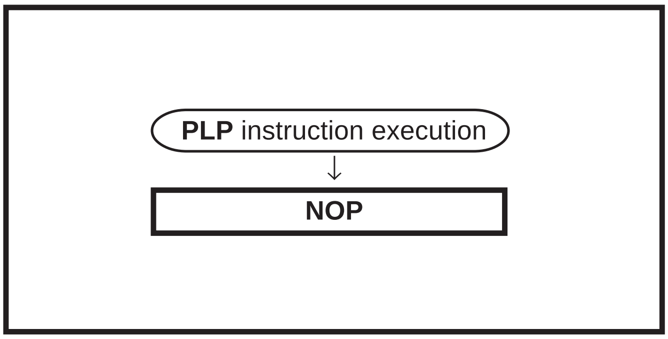


Fig. 3.3.8 Sequence of PLP instruction execution

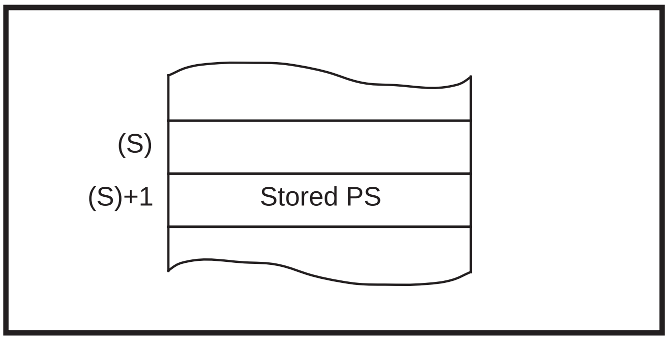


Fig. 3.3.9 Stack memory contents after PHP instruction execution

(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.

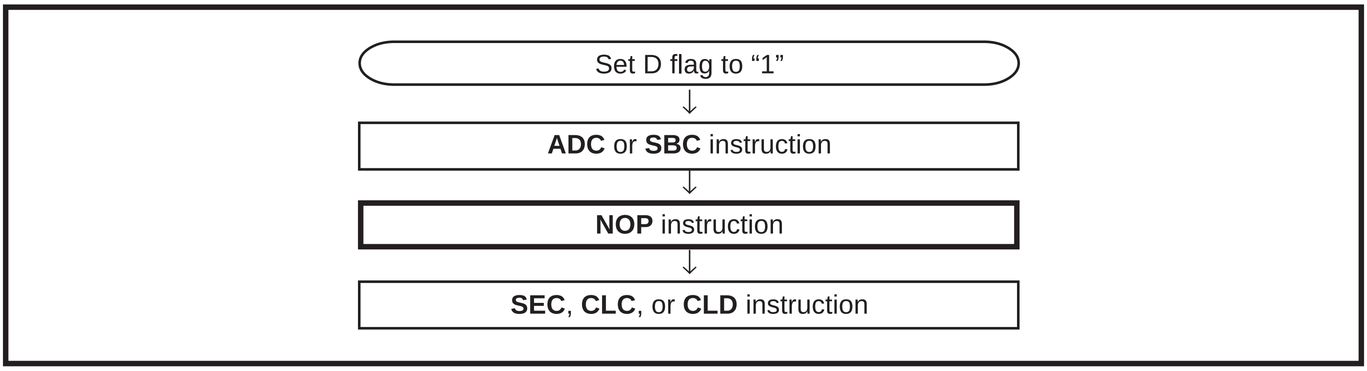


Fig. 3.3.10 Status flag at decimal calculations

(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.

3.3.15 Notes on EPROM version

(1) ERPOM version

■ **At erasing**

- ● Contents of the windowed EPROM are erased through an ultraviolet light source of the wavelength 2537 Ångstrom. At least 15 W•sec/cm^2 are required to erase EPROM contents.
- ● Sunloght and fluorescent light include light that may erase the information written in the built-in PROM. When using the EPROM version in the read mode, be sure to cover the transparent glass portion with a seal.
- ● The seal to cover the transparent glass portion in preparedn our side. Be careful not to bring the seal into contact with the microcomputer lead wires when covering the portion with the seal because this seal is made of metal (aluminum).
- ● Before erasing data, the transparent glass. If any stain or seal adhesive is stuck to the transparent glass, this prevents ultraviolet rays from passing, thereby affecting the erase characteristic adversely.
- ● When erasing user’s program, the firmware is elased too at the same time.

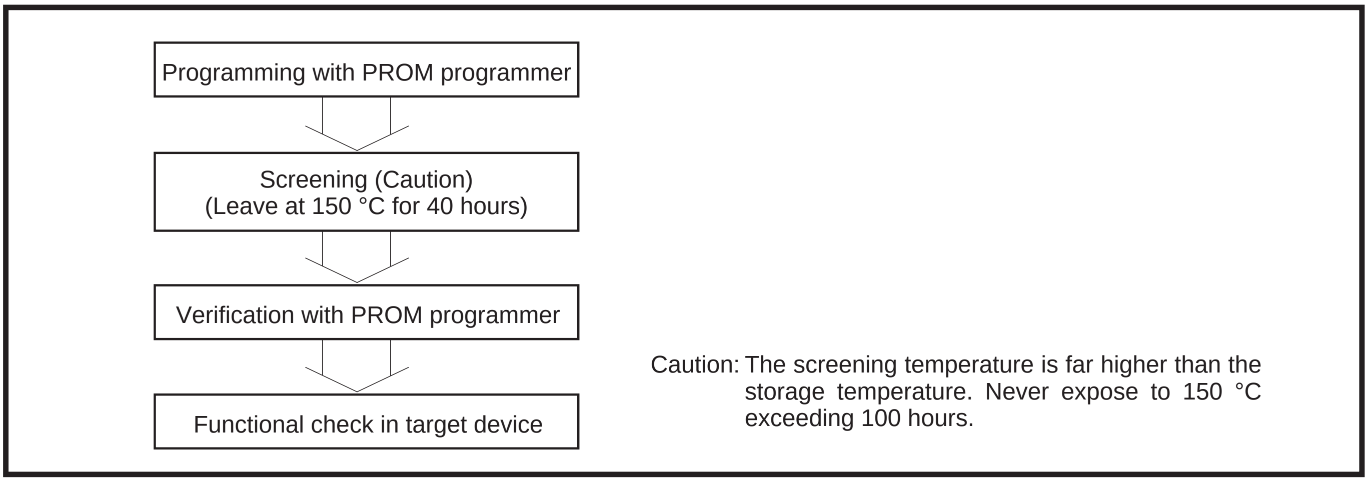
■ **At mounting**

- ● To mount the EPROM version for a purpose other than evaluation, use a suitable mounting socket. When mounting a ceramic package on the socket, fix it securely with silicone resin.

(2) Programming and test of built-in PROM version

As for in the One Time PROM version (shipped in blank) and the built-in EPROM version, their built-in PROM can be read or programmed with a general-purpose PROM programmer using a special programming adapter.

The built-in EPROM version is available only for program development and on-chip program evaluation. The programming test and screening for PROM of the One Time PROM version (shipped in blank) are not performed in the assembly process and the following processes. To ensure reliability after programming, performing programming and test according to the Figure 3.3.11 before actual use are recommended.


Programming with PROM programmer
Screening (Caution)
(Leave at 150 °C for 40 hours)
Verification with PROM programmer
Functional check in target device

Caution: The screening temperature is far higher than the storage temperature. Never expose to 150 °C exceeding 100 hours.

Fig. 3.3.11 Programming and testing of One Time PROM version

3.3.16 Sub-ROM number

Fill in ROM number same as an experimental evaluation on sub-ROM number of the mask ROM confirmation form and the ROM programming confirmation form.

3.3.17 Termination of unused pins

(1) Terminate unused pins

① Output ports : Open

② Input ports :

Connect each pin to Vss through each resistor of 1 kΩ to 10 kΩ.

As for pins whose potential affects to operation modes such as pins INT or others, select the Vcc pin or the Vss pin according to their operation mode.

③ I/O ports :

- Set the I/O ports for the input mode and connect them to 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.

(2) Termination remarks

① Input ports and 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 and ③ 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 microcomputer pins.

3.4 Countermeasures against noise

Countermeasures against noise are described below. The following countermeasures are effective against noise in theory, however, it is necessary not only to take measures as follows but to evaluate before actual use.

3.4.1 Shortest wiring length

The wiring on a printed circuit board can function as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

(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 20 mm)

● **Reason**

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

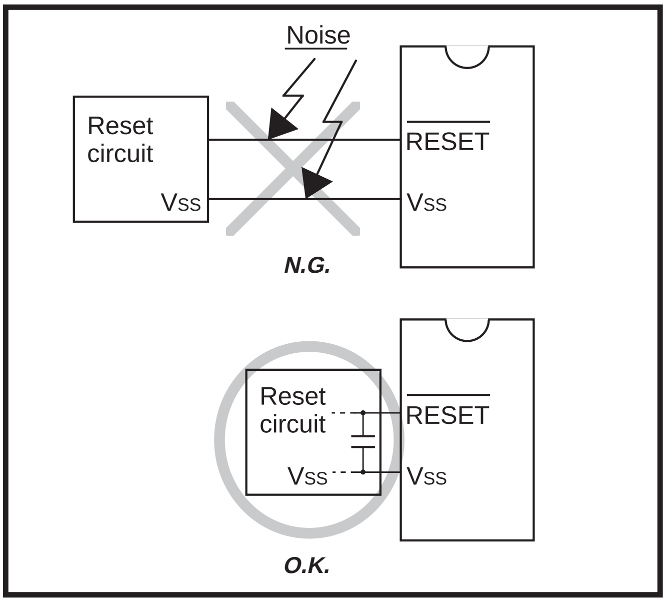


Fig. 3.4.1 Wiring for the $\overline{\text{RESET}}$ pin

(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.

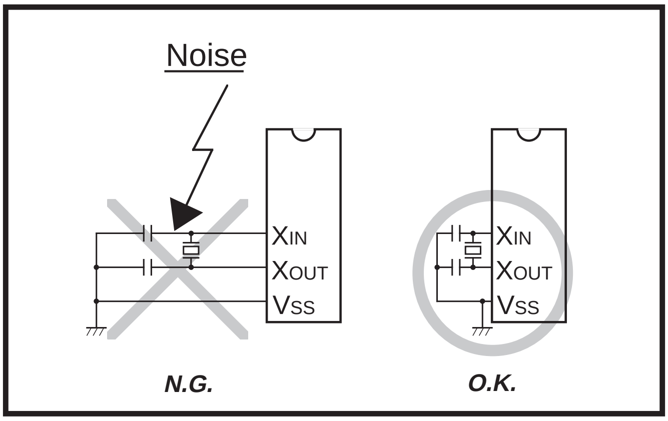


Fig. 3.4.2 Wiring for clock I/O pins

(3) Wiring to V_{PP} pin of One Time PROM version and EPROM version

Connect an approximately 5 kΩ resistor to the V_{PP} pin the shortest possible in series. When not connecting the resistor, make the length of wiring the V_{PP} pin the shortest possible.

Note: Even when a circuit which included an approximately 5 kΩ resistor is used in the Mask ROM version, the microcomputer operates correctly.

● **Reason**

The V_{PP} pin of the One Time PROM and the EPROM version is the power source input pin for the built-in PROM. When programming in the built-in PROM, the impedance of the V_{PP} pin is low to allow the electric current for writing flow into the PROM. Because of this, noise can enter easily. If noise enters the V_{PP} pin, abnormal instruction codes or data are read from the built-in PROM, which may cause a program runaway.

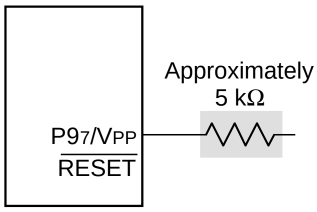


Fig. 3.4.3 Wiring for the V_{PP} pin of the One Time PROM and the EPROM version

3.4.2 Connection of bypass capacitor across V_{SS} line and V_{CC} line

Connect an approximately 0.1 μF bypass capacitor across the V_{SS} line and the V_{CC} line as follows:

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

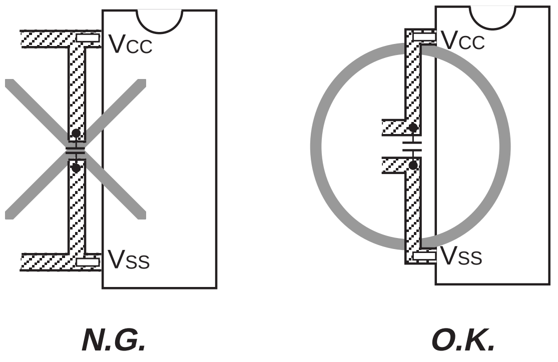


Fig. 3.4.4 Bypass capacitor across the V_{SS} line and the V_{CC} line

3.4.3 Wiring to analog input pins

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

● **Reason**

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

If a capacitor between an analog input pin and the Vss pin is grounded at a position far away from the Vss pin, noise on the GND line may enter a microcomputer through the capacitor.

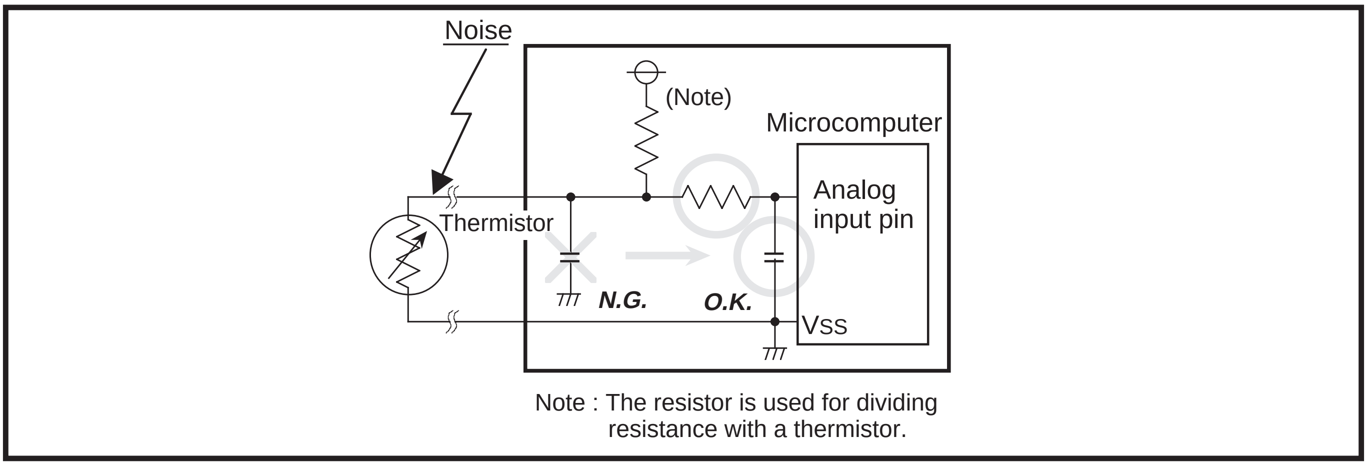

Noise
(Note)
Microcomputer
Analog input pin
Thermistor
N.G.
O.K.
Vss
Note : The resistor is used for dividing resistance with a thermistor.

Fig. 3.4.5 Analog signal line and a resistor and a capacitor

3.4.4 Oscillator concerns

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

(1) Keeping oscillator away from large current signal lines

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

● **Reason**

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

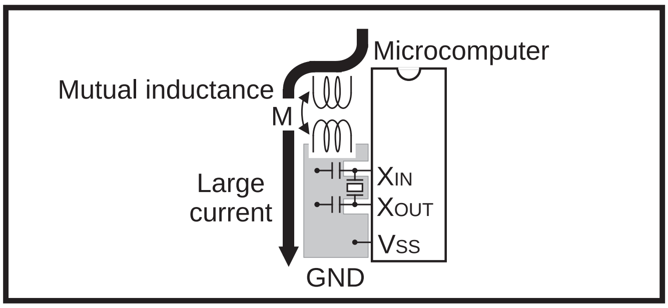


Fig. 3.4.6 Wiring for a large current signal line

(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.

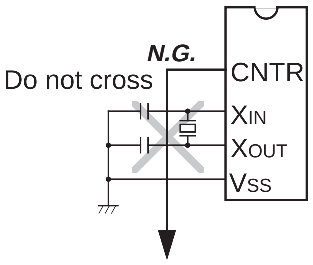


Fig. 3.4.7 Wiring to a signal line where potential levels change frequently

(3) Oscillator protection using Vss pattern

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

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

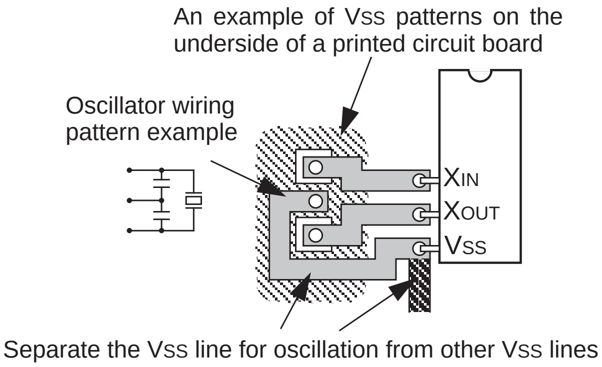

An example of Vss patterns on the underside of a printed circuit board
Oscillator wiring pattern example
XIN
XOUT
VSS
Separate the Vss line for oscillation from other Vss lines

Fig. 3.4.8 Vss pattern on the underside of an oscillator

3.4.5 Setup for I/O ports

Setup I/O ports using hardware and software as follows:

<Hardware>

- Connect a resistor of 100 Ω or more to an I/O port in series.

<Software>

- As for an input port, read data several times by a program for checking whether input levels are equal or not.
- As for an output port, since the output data may reverse because of noise, rewrite data to its port latch at fixed periods.
- Rewrite data to direction registers and pull-up control registers at fixed periods.

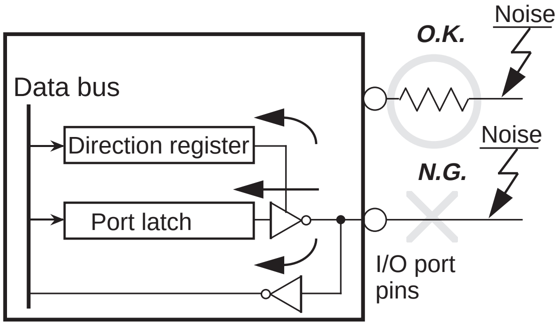


Fig. 3.4.9 Setup for I/O ports

3.4.6 Providing of watchdog timer function by software

If a microcomputer runs away because of noise or others, it can be detected by a software watchdog timer and the microcomputer can be reset to normal operation. This is equal to or more effective than program runaway detection by a hardware watchdog timer. The following shows an example of a watchdog timer provided by software.

In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt processing routine and the interrupt processing routine detects errors of the main routine.

This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

<The main routine>

- Assigns a single byte of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:
  N+1 ≥ (Counts of interrupt processing executed in each main routine)
  As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.
- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
  If the SWDT contents do not change after interrupt processing.

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
  If the SWDT contents are not initialized to the initial value N but continued to decrement and if they reach 0 or less.

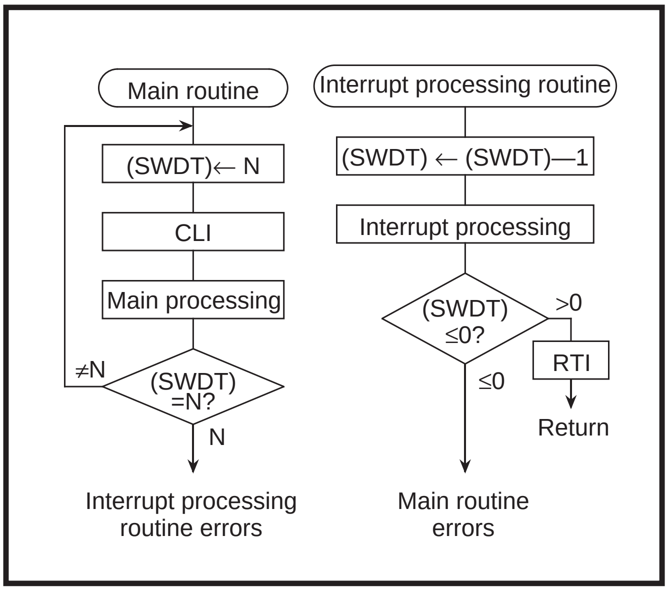
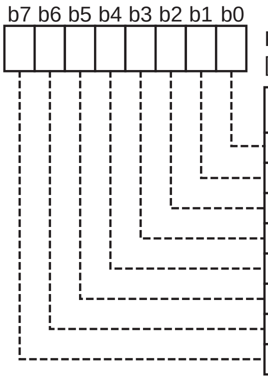


Fig. 3.4.10 Watchdog timer by software

3.5 Control Registers

Port Pi

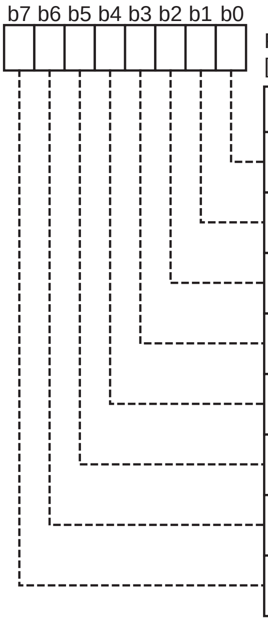


Port Pi (Pi)(i = 0 to 9)
[Addresses 00$_{16}$, 02$_{16}$, 04$_{16}$, 06$_{16}$, 08$_{16}$, 0A$_{16}$, 0C$_{16}$, 0E$_{16}$, 10$_{16}$, 12$_{16}$]

b	Name	Functions	At reset	R	W
0	Port Pi$_0$	●In output mode Write •••••••• Port latch Read •••••••• Port latch ●In input mode Write •••••••• Port latch Read •••••••• Value of pin	0	○	○
1	Port Pi$_1$		0	○	○
2	Port Pi$_2$		0	○	○
3	Port Pi$_3$		0	○	○
4	Port Pi$_4$		0	○	○
5	Port Pi$_5$		0	○	○
6	Port Pi$_6$		0	○	○
7	Port Pi$_7$		0	○	○

Fig. 3.5.1 Structure of port Pi (i = 0 to 9)

Port Pi direction register

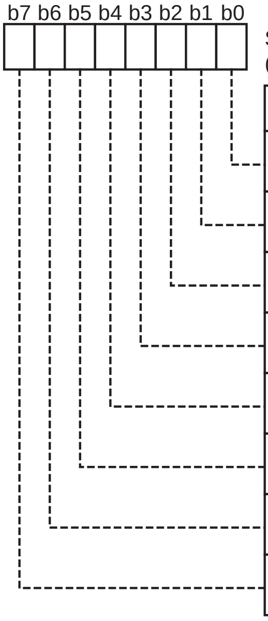


Port Pi direction register (PiD) (i = 0 to 8)
[Addresses 01$_{16}$, 03$_{16}$, 05$_{16}$, 07$_{16}$, 09$_{16}$, 0B$_{16}$, 0D$_{16}$, 0F$_{16}$, 11$_{16}$]

b	Name	Functions	At reset	R	W
0	Port Pi direction register	0 : Port Pi$_0$ input mode 1 : Port Pi$_0$ output mode	0	×	○
1		0 : Port Pi$_1$ input mode 1 : Port Pi$_1$ output mode	0	×	○
2		0 : Port Pi$_2$ input mode 1 : Port Pi$_2$ output mode	0	×	○
3		0 : Port Pi$_3$ input mode 1 : Port Pi$_3$ output mode	0	×	○
4		0 : Port Pi$_4$ input mode 1 : Port Pi$_4$ output mode	0	×	○
5		0 : Port Pi$_5$ input mode 1 : Port Pi$_5$ output mode	0	×	○
6		0 : Port Pi$_6$ input mode 1 : Port Pi$_6$ output mode	0	×	○
7		0 : Port Pi$_7$ input mode 1 : Port Pi$_7$ output mode	0	×	○

Fig. 3.5.2 Structure of port Pi direction register (i = 0 to 8)

Serial I/O3 register/transfer counter



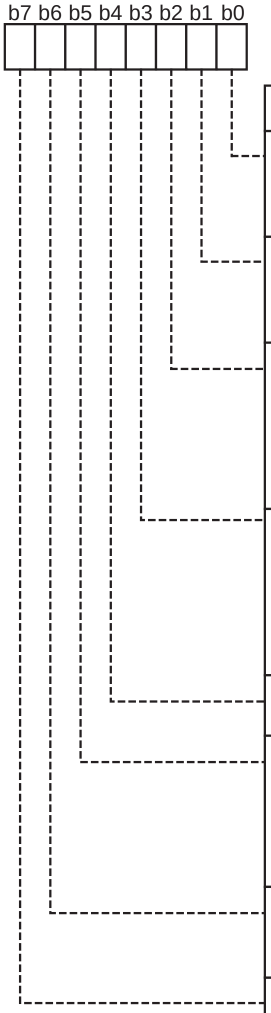
Serial I/O3 register/transfer counter
(SIO3: address 13_{16})

b	Functions	At reset	R	W
0	● When selecting the 8-bit serial I/O mode, this register becomes the transmit/receive buffer register. When selecting the internal synchronous clock, the transfer is started by writing the transmit data. ● When selecting the automatic transfer serial I/O mode, this register becomes the data transfer counter. Set (byte number – 1) to this register. ● When selecting the arbitrary bit serial I/O mode, this register becomes the data transfer counter. Set (bit number – 1) to this register. When the value is set to this register, the serial I/O communication is started at the same time.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Note: When selecting an external clock, after writing a value to the serial I/O3 register/transfer counter, wait for 5 or more cycles of internal clock φ before inputting the transfer clock to the SCLK3 pin.

Fig. 3.5.3 Structure of serial I/O3 register/transfer counter

Serial I/O3 control register 1



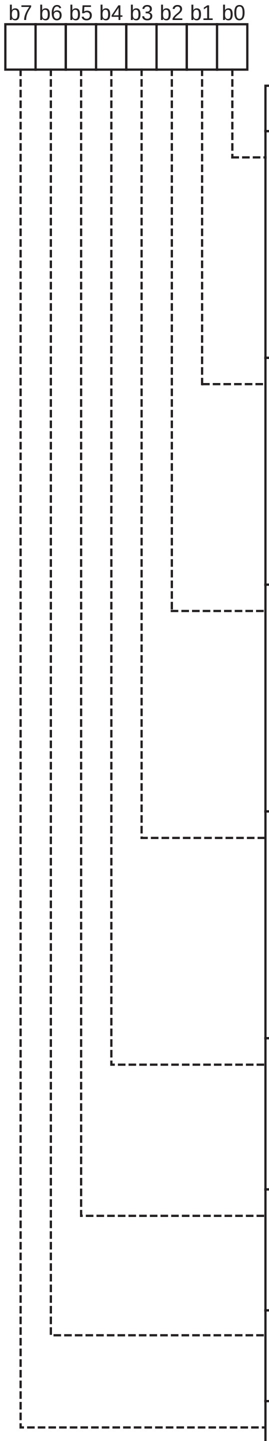
Serial I/O3 control register 1
(SIO3CON1•SC31: address 14_{16})

b	Name	Functions	At reset	R	W
0	Serial transfer selection bits	00: Serial I/O disabled ($P8_2$ to $P8_7$ pins are I/O ports.) 01: 8-bit serial I/O 10: Arbitrary bit serial I/O 11: Automatic transfer serial I/O (8 bits)	0	○	○
1			0	○	○
2	Serial I/O3 synchronous clock selection bits ($P8_7$/S_{STB3} pin control bits)	00: Internal synchronous clock ($P8_7$ pin is I/O port.) 01: External synchronous clock ($P8_7$ pin is I/O port.) 10: Internal synchronous clock ($P8_7$ pin is $\overline{S_{STB3}}$ output.) 11: Internal synchronous clock ($P8_7$ pin is S_{STB3} output.)	0	○	○
3			0	○	○
4	Serial I/O initialization bit	0: Serial I/O initialization 1: Serial I/O enabled	0	○	○
5	Transfer mode selection bit	0: Full-duplex (transmit/receive) mode ($P8_3$ pin is S_{IN3} I/O.) 1: Transmit-only mode ($P8_3$ pin is I/O port.)	0	○	○
6	Serial I/O3 transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
7	Automatic transfer RAM transmit/receive address selection bit	0: Transmit/Receive address match 200_{16} to $2FF_{16}$ **(Note 1)** 1: Transmit address 200_{16} to $27F_{16}$ Receive address 280_{16} to $2FF_{16}$ **(Note 2)**	0	○	○

Notes 1: Set "00_{16}" to "FF_{16}" to the automatic transfer data pointer.
2: Set "00_{16}" to "$7F_{16}$" to the automatic transfer data pointer.
3: Set "0" to the serial I/O initialization bit when terminating a serial transfer during transferring.
4: When writing "1" to the serial I/O initialization bit, serial I/O3 is enabled, but each register is not initialized. Set the value of each register by program.
5: A serial I/O3 interrupt request occurs when "0" is written to the serial I/O initialization bit during an operation. Disable the interrupt enable bit as necessary by program.

Fig. 3.5.4 Structure of serial I/O3 control register 1

Serial I/O3 control register 2



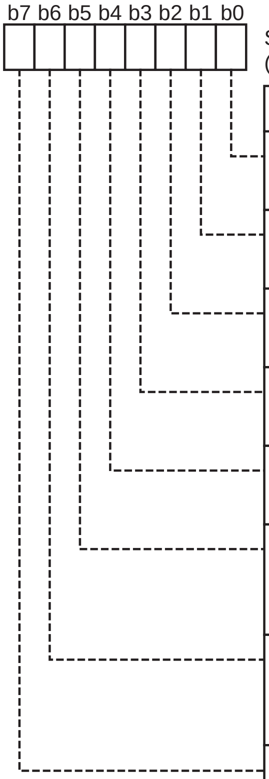
Serial I/O3 control register 2
(SIO3CON2 • SC32: address 15$_{16}$)

b	Name	Functions	At reset	R	W
0	P8$_5$/SRDY3 • P8$_6$/SBUSY3 pin control bits	0000: P8$_5$, P8$_6$ pins are I/O ports. 0001: Not used 0010: P8$_5$ pin is $\overline{\text{SRDY3}}$ output; P8$_6$ pin is I/O port. 0011: P8$_5$ pin is SRDY3 output; P8$_6$ pin is I/O port. 0100: P8$_5$ pin is I/O port; P8$_6$ pin is $\overline{\text{SBUSY3}}$ input. 0101: P8$_5$ pin is I/O port; P8$_6$ pin is SBUSY3 input. 0110: P8$_5$ pin is I/O port; P8$_6$ pin is $\overline{\text{SBUSY3}}$ output. 0111: P8$_5$ pin is I/O port; P8$_6$ pin is SBUSY3 output. 1000: P8$_5$ pin is $\overline{\text{SRDY3}}$ input; P8$_6$ pin is $\overline{\text{SBUSY3}}$ output. 1001: P8$_5$ pin is $\overline{\text{SRDY3}}$ input; P8$_6$ pin is SBUSY3 output. 1010: P8$_5$ pin is SRDY3 input; P8$_6$ pin is $\overline{\text{SBUSY3}}$ output. 1011: P8$_5$ pin is SRDY3 input; P8$_6$ pin is SBUSY3 output. 1100: P8$_5$ pin is $\overline{\text{SRDY3}}$ output; P8$_6$ pin is $\overline{\text{SBUSY3}}$ input. 1101: P8$_5$ pin is $\overline{\text{SRDY3}}$ output; P8$_6$ pin is SBUSY3 input. 1110: P8$_5$ pin is SRDY3 output; P8$_6$ pin is $\overline{\text{SBUSY3}}$ input. 1111: P8$_5$ pin is SRDY3 output; P8$_6$ pin is SBUSY3 input.	0	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4	SBUSY3 output • SSTB3 output function selection bit (Valid in automatic transfer mode)	0: Functions as signal for each 1-byte 1: Functions as signal for each transfer data set	0	○	○
5	Serial transfer status flag	0: Serial transfer completed 1: Serial transfer in-progress	0	○	×
6	SOUT3 output control bit (when serial data is not transferred)	0: Output active 1: Output high-impedance	0	○	○
7	P8$_2$/SOUT3 • P8$_4$/SCLK3 P-channel output disable bit	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○

Note: The SOUT3 output control bit can be used to select the state of the SOUT3 pin when serial data is not transferred; either output active or high-impedance. However, when selecting an external synchronous clock; the SOUT3 pin can become the high-impedance state by setting the SOUT3 output control bit to “1” when SCLK3 input is at “H” after transfer completion.

Fig. 3.5.5 Structure of serial I/O3 control register 2

Serial I/O3 control register 3



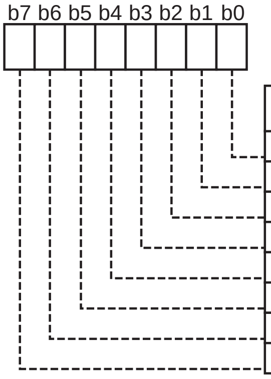
Serial I/O3 control register 3
(SIO3CON3 • SC33: address 16_{16})

b	Name	Functions	At reset	R	W
0	Automatic transfer interval set bit **(Note)**	00000: 2 cycles of transfer clock 00001: 3 cycles of transfer clock 00010: 4 cycles of transfer clock 00011: 5 cycles of transfer clock to 11110: 32 cycles of transfer clock 11111: 33 cycles of transfer clock	0	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5	Serial I/O3 internal synchronous clock selection bits	000 : $f(X_{IN})/4$ or $f(X_{CIN})/4$ 001 : $f(X_{IN})/8$ or $f(X_{CIN})/8$ 010 : $f(X_{IN})/16$ or $f(X_{CIN})/16$ 011 : $f(X_{IN})/32$ or $f(X_{CIN})/32$ 100 : $f(X_{IN})/64$ or $f(X_{CIN})/64$ 101 : $f(X_{IN})/128$ or $f(X_{CIN})/128$ 110 : $f(X_{IN})/256$ or $f(X_{CIN})/256$ 111 : $f(X_{IN})/512$ or $f(X_{CIN})/512$	0	○	○
6			0	○	○
7			0	○	○

Note: Write: kept in latch
Read: from decrement counter

Fig. 3.5.6 Structure of serial I/O3 control register 3

Serial I/O3 automatic transfer data pointer



Serial I/O3 automatic transfer data pointer
(SIO3DP: address 17_{16})

b	Name	Functions	At reset	R	W
0	Automatic transfer data pointer set bits **(Note)**	Indicates the low-order 8 bits of the address stored the start data on the serial I/O3 automatic transfer RAM.	0	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Note: Write: kept in latch
Read: from decrement counter

Fig. 3.5.7 Structure of serial I/O3 automatic transfer data pointer

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

Transmit/Receive buffer register
(TB/RB: address 18_{16})

b	Functions	At reset	R	W
0	This is the serial I/O1 exclusive buffer register which is used to write transmit data or to read receive data. • At write : The value is written to the transmit buffer register. • At read : The contents of the receive buffer register is read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Note: The contents of the transmit buffer register cannot be read out.
The value cannot be written to the receive buffer register.

Fig. 3.5.8 Structure of transmit/receive buffer register

Serial I/O1 status register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O1 status register
(SIO1STS: address 19_{16})

b	Name	Functions	At reset	R	W
0	Transmit buffer empty flag (TBE)	0: Buffer full 1: Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0: Buffer empty 1: Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0: Transmit shift in progress 1: Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0: No error 1: Overrun error	0	○	×
4	Parity error flag (PE)	0: No error 1: Parity error	0	○	×
5	Framing error flag (FE)	0: No error 1: Framing error	0	○	×
6	Summing error flag (SE)	0: (OE) U (PE) U (FE) = 0 1: (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is arranged for this bit. This is write disabled bit. When this bit is read out, the contents is "1".		1	○	×

Fig. 3.5.9 Structure of serial I/O1 status register

Serial I/O1 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O1 control register
(SIO1CON: address $1A_{16}$)

b	Name	Functions	At reset	R	W
0	BRG count source selection bit (CSS)	0: f(X_{IN}) 1: f(X_{IN})/4	0	○	○
1	Serial I/O1 synchronous clock selection bit (SCS)	•In clock synchronous mode 0: BRG output/4 1: External clock input •In UART mode 0: BRG output/16 1: External clock input/16	0	○	○
2	S_{RDY1} output enable bit (SRDY)	0: $P4_7/\overline{S_{RDY1}}$ pin operates as normal I/O port $P4_7$ 1: $P4_7/\overline{S_{RDY1}}$ pin operates as signal output pin $\overline{S_{RDY1}}$	0	○	○
3	Transmit interrupt source selection bit (TIC)	0: When transmit buffer has emptied 1: When transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0: Transmit disabled 1: Transmit enabled	0	○	○
5	Receive enable bit (RE)	0: Receive disabled 1: Receive enabled	0	○	○
6	Serial I/O1 mode selection bit (SIOM)	0: Clock asynchronous serial I/O (UART) mode 1: Clock synchronous serial I/O mode	0	○	○
7	Serial I/O1 enable bit (SIOE)	0: Serial I/O1 disabled (pins $P4_4$–$P4_7$ operate as normal I/O pins) 1: Serial I/O1 enabled (pins $P4_4$–$P4_7$ operate as serial I/O pins)	0	○	○

Notes 1: When selecting clock synchronous serial I/O
- Stop of transmission operation
  As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the serial I/O1 enable bit and the transmit enable bit to “0” (serial I/O and transmit disabled).
- Stop of receive operation
  As for the serial I/O1 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/O1 enable bit to “0” (serial I/O disabled).
- Stop of transmit/receive operation
  As for the serial I/O1 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.)

2: When selecting clock asynchronous serial I/O
- Stop of transmission operation
  As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the transmit enable bit to “0” (transmit disabled).
- Stop of receive operation
  As for the serial I/O1 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).
- Stop of transmit/receive operation
  - Only transmission operation is stopped
    As for the serial I/O1 that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the transmit enable bit to “0” (transmit disabled).
  - Only receive operation is stopped
    As for the serial I/O1 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).

3: When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the S_{CLK1} input level. Also, write data to the transmit buffer register at “H” of the S_{CLK1} input level.

4: When an external clock input is selected as the synchronous clock and the receiver perform the $\overline{S_{RDY1}}$ output, set “1” to the transmit enable bit in addition to the receive enable bit and the S_{RDY1} output enable bit.

5: When an external clock input is selected as the synchronous clock, set “1” to the transmit enable bit while the synchronous clock is “H” state.

6: Transmit interrupt request when transmit enable bit is set
The transmiit interrupt request bit is set and the interrupt request occurs even when selecting timing that either of the following flags is set to “1” as timing where the transmit interrupt occurs.
- Transmit buffer empty flag is set to “1”
- Transmit shift register completion flag is set to “1”

Accordingly, when the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as the following sequence.
- Transmit enable bit is set to “1”
- Transmit interrupt request bit is set to “0”

7: In order to stop a transmit, set the transmit enable bit to “0” (transmit disable).
Do not set only the serial I/O1 enable bit to “0”.

8: A receive operation can be stopped by either setting the receive enable bit to “0” or the serial I/O1 enable bit to “0”.

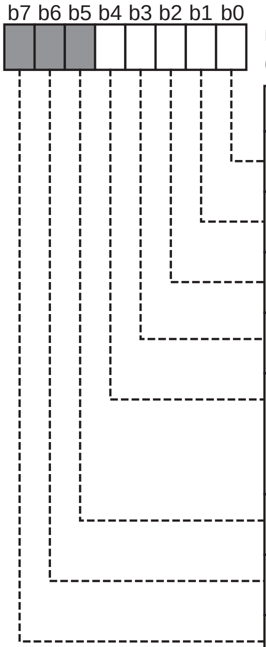
9: To stop a transmit when transferring in clock synchronous serial I/O mode, set both the transmit enable bit and the receive enable bit to “0” at the same time.

10: To set the serial I/O1 control register again, first set the transmit enable/receive enable bits to “0”. Next, reset the transmit/receive circuits, and, finally, reset the serial I/O1 control register.

11: Note when confirming the transmit shift register completion flag and controlling the data transmit after writing a transmit data to the transmit buffer. There is a delay of 0.5 to 1.5 shift clock cycles while the transmit shift register completion flag goes from “1” to “0”.

Fig. 3.5.10 Structure of serial I/O1 control register

UART control register



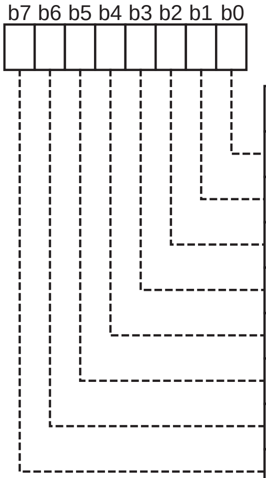
UART control register
(UARTCON: address 1B$_{16}$)

b	Name	Functions	At reset	R	W
0	Character length selection bit (CHAS)	0: 8 bits 1: 7 bits	0	○	○
1	Parity enable bit (PARE)	0: Parity checking disabled 1: Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0: Even parity 1: Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0: 1 stop bit 1: 2 stop bits	0	○	○
4	P4$_5$/TxD P-channel output disable bit (POFF)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “1”.		1	○	×
6			1	○	×
7			1	○	×

Note: The P4$_5$/TxD channel output disable bit of the UART control register is valid both when using as a normal port and when using as the TxD pin. However, do not apply a voltage of Vcc + 0.3 V or more to the P4$_5$/TxD pin even when it is used as N-channel open drain output.
In the serial I/O1, after transmit is completed, the TxD pin continues latching and outputting the last bit.

Fig. 3.5.11 Structure of UART control register

Baud rate generator

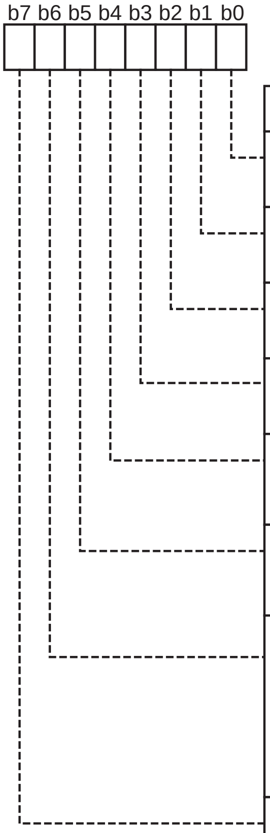


Baud rate generator
(BRG: address 1C$_{16}$)

b	Functions	At reset	R	W
0	• Bit rate of the serial transfer is determined. • This is the 8-bit counter and has the reload register. The count source is divided by n+1 by specifying a value n.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 3.5.12 Structure of baud rate generator

Serial I/O2 control register



Serial I/O2 control register
(SIO2CON: address 1D$_{16}$)

b	Name	Functions	At reset	R	W
0	Serial I/O2 internal synchronous clock selection bits	000: f(XIN)/8 or f(XCIN)/8 001: f(XIN)/16 or f(XCIN)/16 010: f(XIN)/32 or f(XCIN)/32 011: f(XIN)/64 or f(XCIN)/64 110: f(XIN)/128 or f(XCIN)/128 111: f(XIN)/256 or f(XCIN)/256	0	○	○
1			0	○	○
2			0	○	○
3	SOUT2 pin selection bit	0: I/O port 1: SOUT2 output pin	0	○	○
4	(P7$_1$/SOUT2•P7$_2$/SCLK2) P-channel output disable bit	0: CMOS 3 state 1: N-channel open-drain output	0	○	○
5	Serial I/O2 transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
6	SCLK2 pin selection bit	0: External clock (SCLK2 functions as an I/O port.) 1: Internal clock (SCLK2 functions as an output port.)	0	○	○
7	SOUT2 output control bit (when serial data is not transferred)	0: Output active 1: High-impedance	0	○	○

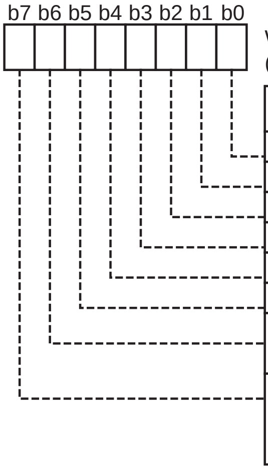
Notes 1: The serial I/O2 interrupt request bit is set to “1” by counting eight times of the transfer clock when the synchronous clock is the internal clock or the external clock. However, when using the external clock, the contents of the serial I/O2 register is being shifted continuously while the transfer clock is input to the serial I/O2’s circuit. Stop the transfer clock at 8 times. (When using the internal clock, the transfer clock automatically stops.)

2: When using the external clock, the SOUT2 pin does not become the high-impedance state after the data transfer is completed. Set “1” to the SOUT2 output control bit of the serial I/O2 control register after the data transfer is completed.
When using the internal clock, the SOUT2 pin automatically becomes the high-impedance state after the data transfer is completed.

3: When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLK2 input level. Also, write data to the serial I/O2 register at “H” of the SCLK2 input level.

Fig. 3.5.13 Structure of serial I/O2 control register

Watchdog timer control register



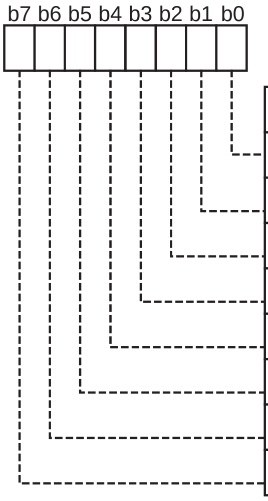
Watchdog timer control register
(WDTCON: address $1E_{16}$)

b	Name	Functions	At reset	R	W
0	Watchdog timer H (high-order 6 bits of reading exclusive)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0: STP instruction enabled 1: STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0: Watchdog timer L underflow 1: f(X_{IN})/16 or f(X_{CIN})/16	0	○	○

Note: Once a “1” is written to the STP instruction disable bit of the watchdog timer control register, it cannot be programmed to “0” again.

Fig. 3.5.14 Structure of watchdog timer control register

Serial I/O2 register

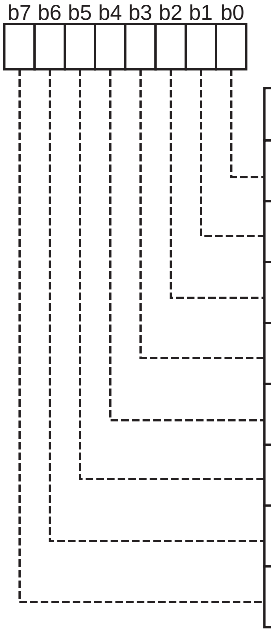


Serial I/O2 register
(SIO2: address $1F_{16}$)

b	Functions	At reset	R	W
0	This is the serial I/O2 exclusive transmit/receive buffer register. At transmit: “00_{16}” to “FF_{16}” can be set as the transmit data. The transmit data is automatically transferred to the transmit shift register by writing the transmit data. At receive: When all receive data has been input into the receive shift register, the receive data is automatically transferred to this register.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 3.5.15 Structure of serial I/O2 register

Timer X (low-order, high-order)



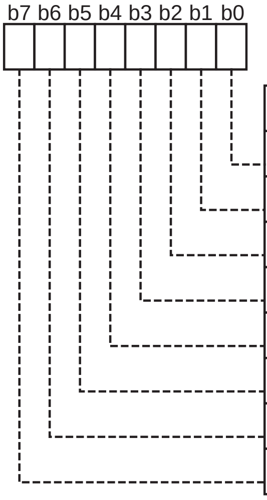
Timer X (low-order, high-order)
(TXL, TXH: addresses 20$_{16}$, 21$_{16}$)

b	Functions	At reset	R	W
0	• Set “0000$_{16}$ to FFFF$_{16}$” as timer X count value. • When the timer X write control bit of the timer X mode register is “0”, the value is written to timer X and the latch at one time. When the timer X write control bit of the timer X mode register is “1”, the value is written only to the latch. • Count-down method Timer X underflows at the first count pulse after the value of timer X reaches “0”. At this time, the value of the timer X latch is transferred to timer X automatically. • The timer X count value is read out by reading this register. • When a value n (0 to FFFF$_{16}$) is written to timer X, the division ratio is 1/(n+1).	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Notes 1: When reading and writing, perform them to both the high-order and low-order bytes.
2: Read both registers in order of TXH and TXL following.
3: Write both registers in order of TXL and TXH following.
4: Do not read both registers during a write, and do not write to both registers during a read.
5: When writing for the latch only, if writing timing for the high-order latch is the almost same as the underflow timing, unexpected value may be set in the high-order counter.

Fig. 3.5.16 Structure of timer X (low-order, high-order)

Timer Y (low-order, high-order)



Timer Y (low-order, high-order)
(TYL, TYH: addresses 22$_{16}$, 23$_{16}$)

b	Functions	At reset	R	W
0	• Set “0000$_{16}$ to FFFF$_{16}$” as timer Y count value. • Count-down method Timer Y underflows at the first count pulse after the value of timer Y reaches “0”. At this time, the value of the timer Y latch is transferred to timer Y automatically. • The timer Y count value is read out by reading this register. • When a value n (0 to FFFF$_{16}$) is written to timer Y, the division ratio is 1/(n+1).	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Notes 1: When reading and writing, perform them to both the high-order and low-order bytes.

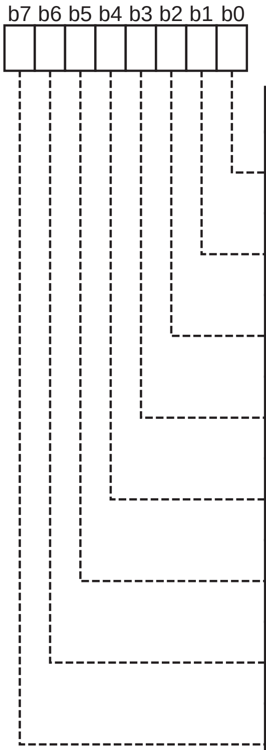
2: Read both registers in the order of TYH and TYL following.

3: Write both registers in the order of TYL and TYH following.

4: Do not read both registers during a write, and do not write both registers during a read.

Fig. 3.5.17 Structure of timer Y (low-oeder. high-oeder)

Timer i (i = 1 to 3)



Timer 1 (T1: address 24_{16})
Timer 2 (T2: address 25_{16})
Timer 3 (T3: address 26_{16})

b	Functions	At reset	R	W
0	• Set “00_{16}” to “FF_{16}” as timer i count value. • Timer 1 and timer 3 A set value is written to the timer latch and timer. • Timer 2 When the timer 2 write control bit of the timer 123 mode register is “0”, a set value is written to the latch and timer at one time. When the timer 2 write control bit of the timer 123 mode register is “1”, a set value is written to the latch only. • Count-down method The count value is read out by reading timer i. • Timer i underflows at the first count pulse after the value of timer i reaches “0”. At this time, the value of the timer i latch is transferred to timer i automatically. • When reading this register, the count value of corresponding timer is read out. • When a value n (“0” to “255”) is written to the timer latch, the division ratio is 1/(n+1).	**(Note 1)**	○	○
1			○	○
2			○	○
3			○	○
4			○	○
5			○	○
6			○	○
7			○	○

Notes 1: In timer 1 and timer 3, it becomes “FF_{16}”. In timer 2, it becomes “01_{16}”.

2: When switching the count source, a value of timer i may become an inaccurate value.
Set timers again in order of timer 1, timer 2, and timer 3.

3: When writing to the latch only is selected, the value written into timer 2 is written only in the latch for reloading and the contents of timer 2 is not changed. This written value is transferred to timer 2 at the first underflow after writing. Normally, a value is written in both the latch and timer at one time. That is, when a value is written to the timer, it is set in both the latch and the timer.

Fig. 3.5.18 Structure of timer i (i = 1 to 3)

Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer X mode register
(TXM: address 27_{16})

b	Name	Functions	At reset	R	W
0	Timer X write control bit	0 : Write value in latch and counter 1 : Write value in latch only	0	○	○
1	Real time port control bit	0 : Real time port function invalid 1 : Real time port function valid	0	○	○
2	P5$_6$ data for real time port	0 : "0" output 1 : "1" output	0	○	○
3	P5$_7$ data for real time port	0 : "0" output 1 : "1" output	0	○	○
4	Timer X operating mode bits	b5 b4 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
5			0	○	○
6	CNTR$_0$ active edge switch bit	0 : •Count at rising edge in event counter mode •Start from "H" output in pulse output mode •Measure "H" pulse width in pulse width measurement mode •Falling edge active for CNTR$_0$ interrupt 1 : •Count at falling edge in event counter mode •Start from "L" output in pulse output mode •Measure "L" pulse width in pulse width measurement mode •Rising edge active for CNTR$_0$ interrupt	0	○	○
7	Timer X stop control bit	0 : Count start 1 : Count stop	0	○	○

Notes 1: Setting the CNTR$_0$ active edge switch bit affects the active edge of an interrupt. Consequently, a CNTR$_0$ interrupt request may occur by setting the CNTR0 active edge switch bit. As a countermeasure against the above, switch the active edge after disabling the CNTR$_0$ interrupt, and set "0" to the CNTR$_0$ interrupt request bit.

2: Pulse output mode
- Set "1" to bit 4 (CNTR$_0$ pin) of the port P5 direction register (output mode).
- When bit 4 (CNTR$_0$ pin) of the port P5 register is read, the value of the port register are not read out, but the output value of the pin is read out.

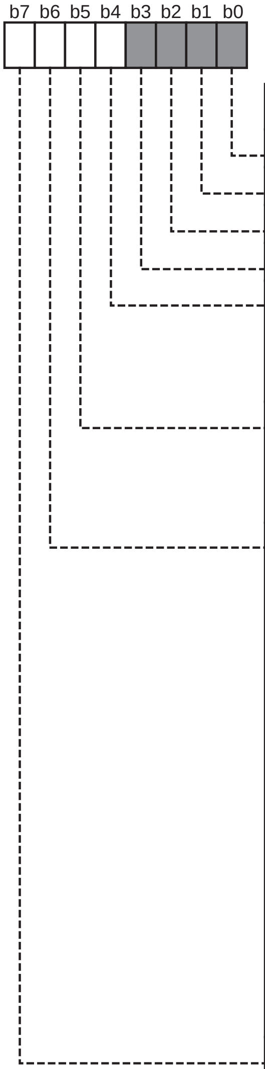
3: Event counter mode
- Set "0" to bit 4 (CNTR$_0$ pin) of the port P5 direction register (input mode).

4: Pulse width measurement mode
- Set "0" to bit 4 (CNTR$_0$ pin) of the port P5 direction register (input mode).
- When reading bit 4 of port P5, the value is "1" at "H" level input or "0" at "L" level input regardless of the value of the CNTR$_0$ active edge switch bit of the timer X mode register.

5: Real time port function
- Port P5$_6$ (RTP$_0$ pin) and port P5$_7$ (RTP$_1$ pin) function as a normal I/O port after reset released. When using ports P5$_6$ and P5$_7$ as real time port function pins, set "1" to the corresponding port direction register to be the output mode.
- Do not switch the pins which is used as the real time port to the input mode during operation.

Fig. 3.5.19 Structure of timer X mode register

Timer Y mode register



Timer Y mode register
(TYM: address 28_{16})

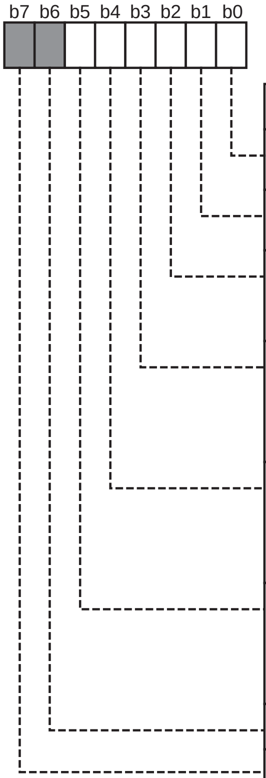
b	Name	Functions	At reset	R	W
0	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
1			0	○	×
2			0	○	×
3			0	○	×
4	Timer Y operating mode bits	b5 b4 0 0 : Timer mode 0 1 : Period measurement mode 1 0 : Event counter mode 1 1 : Pulse width HL continuously measurement mode	0	○	○
5			0	○	○
6	CNTR$_1$ active edge switch bit	0 : • Count at rising edge in event counter mode • Measure the falling edge to falling edge period in period measurement mode • Falling edge active for CNTR$_1$ interrupt 1 : • Count at falling edge in event counter mode • Measure the rising edge to rising edge period in period measurement mode • Rising edge active for CNTR$_1$ interrupt	0	○	○
7	Timer Y stop control bit	0 : Count start 1 : Count stop	0	○	○

Notes 1: Event counter mode and period measurement mode
•Set “0” to bit 5 (CNTR$_1$ pin) of the port P5 direction register to be the input mode.

2: Pulse width HL continuously measurement mode
•Set “0” to bit 5 (CNTR$_1$ pin) of the port P5 direction register to be the input mode.
•The CNTR$_1$ interrupt request occurs at both edges of a input pulse regardless of the contents of the CNTR$_1$ active edge switch bit of the timer Y mode register.

Fig. 3.5.20 Structure of timer Y mode register

Timer 123 mode register



Timer 123 mode register
(T123M: address 29_{16})

b	Name	Functions	At reset	R	W
0	TOUT output active edge switch bit	0 : Start at “H” output 1 : Start at “L” output	0	○	○
1	TOUT output control bit	0 : TOUT output disabled 1 : TOUT output enabled	0	○	○
2	Timer 2 write control bit	0 : Write data in latch and counter at one time 1 : Write data in latch only	0	○	○
3	Timer 2 count source selection bit	0 : Timer 1 output 1 : f(XIN)/16 (or f(XCIN)/16 in low-speed mode) **(Note)**	0	○	○
4	Timer 3 count source selection bit	0 : Timer 1 output 1 : f(XIN)/16 (or f(XCIN)/16 in low-speed mode) **(Note)**	0	○	○
5	Timer 1 count source selection bit	1 : f(XIN)/16 (or f(XCIN)/16 in low-speed mode) **(Note)** 0 : f(XCIN)	0	○	○
6	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
7			0	○	×

Note 1 : In the low-speed mode, the internal clock ϕ is XIN pin input signal divided by 2.

2 : In case of the following, a short pulse occurs on counted input signals, so that the timer count value may change greatly. Accordingly, after setting their count sources, set values in order of timer 1, timer 2, and timer 3 following:

•When the count sources of timers 1 to 3 are switched

•When the timer 1 output signal is selected as a count source of timer 2 or timer 3 and data is written to timer 1

3 : In the TOUT output enabled state of timer 2, a signal whose polarity is reversed each time timer 2 underflows is output from the TOUT pin. In this case, set port $P5_0$ (sharing with the TOUT pin) to the output mode.

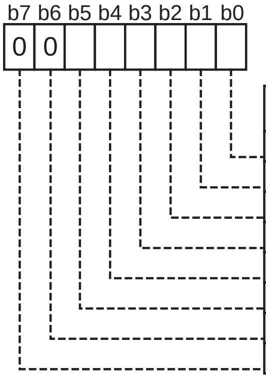
4 : In stop mode

•Set the interrupt enabled bits of timer 1 and timer 2 to the disabled state (“0”) before executing the STP instruction.

•Oscillation restarts at reset or when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU until timer 2 underflows. It is because that retains time to stabilize oscillation using a ceramic resonator, etc. Set values for stabilizing oscillation to the timer 1 latch and timer 2 latch before executing the STP instruction.

Fig. 3.5.21 Structure of timer 123 mode register

Communication mode register



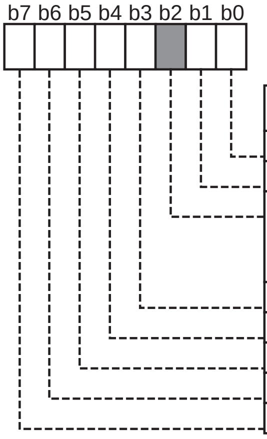
Communication mode register
(BUSM: address $2A_{16}$)

b	Name	Functions	At reset	R	W
0	Arbitrary bits: defined according to each communication protocol specification confirmation.		0	(Note)	
1			Undefined		
2			Undefined		
3			Undefined		
4			0	○	×
5			0	○	×
6	Always write "00" to these bits.		0	○	○
7			0	○	○

Note: Defined according to each communication protcol specification confirmation.

Fig. 3.5.22 Structure of communication mode register

Transmit control register



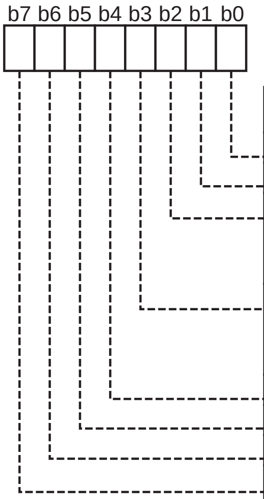
Transmit control register
(TXDCON: address $2B_{16}$)

b	Name	Functions	At reset	R	W
0	Arbitrary bits: defined according to each communication protocol specification confirmation.		0	(Note)	
1			0		
2	Nothing is arranged for this bit. This bit is write disabled bit. When this bit is read out, the contents is "0".		0	0	×
3	Arbitrary bits: defined according to each communication protocol specification confirmation.		0	(Note)	
4			1		
5			0		
6			0		
7			0		

Note: Defined according to each communication protocol specification confirmation.

Fig. 3.5.23 Structure of transmit control register

Transmit status register



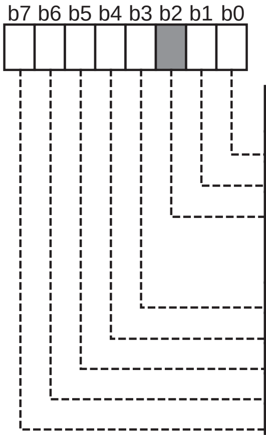
Transmit status register
(TXDSTS: address $2C_{16}$)

b	Name	Functions	At reset	R	W
0	Arbitrary bits: defined according to each communication protocol specification confirmation.		0	(Note)	
1			0		
2	Transmit bus interrupt source 1 request bit	0: No interrupt request issued 1: Interrupt request issued	0	○	✻
3	Transmit bus interrupt source 2 request bit	0: No interrupt request issued 1: Interrupt request issued	0	○	✻
4	Arbitrary bits: defined according to each communication protocol specification confirmation.		0	(Note)	
5			0		
6			0		
7	Transmit bus interrupt source 3 request bit	0: No interrupt request issued 1: Interrupt request issued	0	○	✻

✻: This bit can be set to “0” by software, but cannot be set to “1”.
Note: Defined according to each communication protocol specification confirmation.

Fig. 3.5.24 Structure of transmit status register

Receive control register



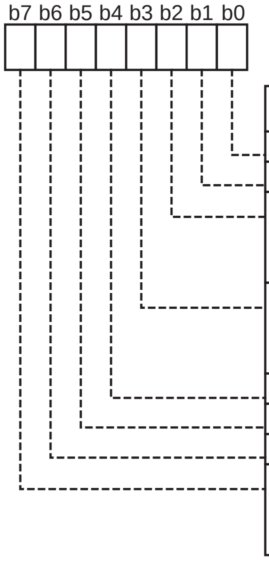
Receive control register
(RXDCON: address $2D_{16}$)

b	Name	Functions	At reset	R	W
0	Arbitrary bits: defined according to each communication protocol specification confirmation.		0	(Note)	
1			0		
2	Nothing is arranged for this bit. This bit is write disabled bit. When this bit is read out, the contents is “0”.		0	○	×
3	Arbitrary bits: defined according to each communication protocol specification confirmation.		0	(Note)	
4			1		
5			0		
6			0		
7			0		

Note: Defined according to each communication protocol specification confirmation.

Fig. 3.5.25 Structure of receive control register

Receive status register



Receive status register
(RXDSTS: address $2E_{16}$)

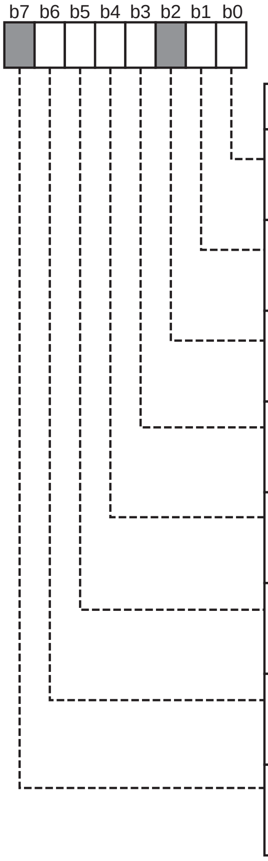
b	Name	Functions	At reset	R	W
0	Arbitrary bits: defined according to each communication protocol specification confirmation.		1	(Note)	
1			0		
2	Receive bus interrupt source 1 request bit	0: No interrupt request issued 1: Interrupt request issued	0	○	✻
3	Receive bus interrupt source 2 request bit	0: No interrupt request issued 1: Interrupt request issued	0	○	✻
4	Arbitrary bits: defined according to each communication protocol specification confirmation.		0	(Note)	
5			0		
6			0		
7	Receive bus interrupt source 3 request bit	0: No interrupt request issued 1: Interrupt request issued	0	○	✻

✻: This bit can be set to “0” by software, but cannot be set to “1”.

Note: Defined according to each communication protocol specification confirmation.

Fig. 3.5.26 Structure of receive status register

Bus interrupt source discrimination control register



Bus interrupt source discrimination control register
(BICOND: address $2F_{16}$)

b	Name	Functions	At reset	R	W
0	Transmit bus interrupt source 1 enable bit	0: Interrupt disabled 1: Interrupt enabled	0	○	○
1	Transmit bus interrupt source 2 enable bit	0: Interrupt disabled 1: Interrupt enabled	0	○	○
2	Nothing is arranged for this bit. This bit is write disabled bit. When this bit is read out, the contents is “0”.		0	○	×
3	Transmit bus interrupt source 3 enable bit	0: Interrupt disabled 1: Interrupt enabled	0	○	○
4	Receive bus interrupt source 1 enable bit	0: Interrupt disabled 1: Interrupt enabled	0	○	○
5	Receive bus interrupt source 2 enable bit	0: Interrupt disabled 1: Interrupt enabled	0	○	○
6	Receive bus interrupt source 2 enable bit	0: Interrupt disabled 1: Interrupt enabled	0	○	○
7	Nothing is arranged for this bit. This bit is write disable bit. When this bit is read out, the contents is “0”.		0	○	×

Fig. 3.5.27 Structure of bus interrupt source discrimination control register

Control field selection register

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0				

Control field selection register
(CFSEL: address 30_{16})

b	Name	Functions	At reset	R	W
0	b3 b2 b1 b0 0000 : 0001 : 0010 : 0011 : 0100 : 0101 : 0110 : 0111 : 1000 : 1001 : 1010 : 1011 : 1100 : 1101 : 1110 : Disabled 1111 : Disabled	Arbitrary bits: defined according to each communication protocol specification confirmation.	0	(Note)	
1			0		
2			0		
3			0		
4	Fix “0000” to these bits.		0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Note: Defined according to each communication protcol specification confirmation.

Fig. 3.5.28 Structure of control field selection register

Control field register

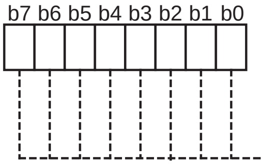
b7	b6	b5	b4	b3	b2	b1	b0

Control field register
(CF: address 31_{16})

b	Name	Functions	At reset	R	W
0	•This is a register for confirmation and change of control field data. •The value of the control field selection register is used for the pointer which selects the control field data.		0	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Fig. 3.5.29 Structure of control field register

Transmit/Receive FIFO

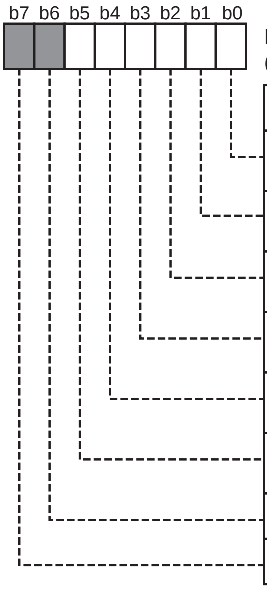


Transmit/Receive FIFO
(TRFIFO: address 32_{16})

b	Name	Functions	At reset	R	W
0 to 16	The transmit FIFO and the receive FIFO is arranged to the same address. The transmit FIFO consists of 8 bits and the receive FIFO consists of 16 bits.		Undefined	○	○

Fig. 3.5.30 Structure of transmit/receive FIFO

PULL UP register



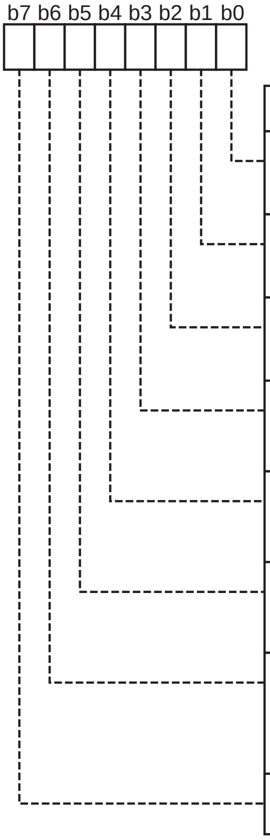
PULL UP register
(PULLU: address 33_{16})

b	Name	Functions	At reset	R	W
0	$P2_6$, $P2_7$ pull-up **(Note 1)**	0: No pull-up 1: Pull-up	0	○	○
1	$P2_5$ pull-up **(Note 2)**	0: No pull-up 1: Pull-up	0	○	○
2	$P2_2$ to $P2_4$ pull-up **(Note 3)**	0: No pull-up 1: Pull-up	0	○	○
3	$P2_0$, $P2_1$ pull-up **(Note 1)**	0: No pull-up 1: Pull-up	0	○	○
4	TxD, SCLK1 pull-up **(Note 1)**	0: No pull-up 1: Pull-up	0	○	○
5	SOUT2, SCLK2 pull-up **(Note 1)**	0: No pull-up 1: Pull-up	0	○	○
6	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
7			0	○	×

Notes 1: This controls pull-up in a unit of 2 bits.
2: This controls pull-up in a unit of 1 bit.
3: This controls pull-up in a unit of 3 bits.

Fig. 3.5.31 Structure of PULL UP register

A-D control register



A-D control register
(ADCON: address 34$_{16}$)

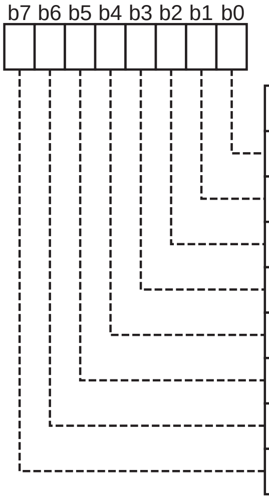
b	Name	Functions	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 000 : P6$_0$/AN$_0$ 001 : P6$_1$/AN$_1$ 010 : P6$_2$/AN$_2$ 011 : P6$_3$/AN$_3$ 100 : P6$_4$/AN$_4$ 101 : P6$_5$/AN$_5$ 110 : P6$_6$/AN$_6$ 111 : P6$_7$/AN$_7$	0	○	○
1			0	○	○
2			0	○	○
3	AD conversion completion bit **(Note)**	0: Conversion in progress 1: Conversion completed	1	○	○
4	V$_{REF}$ input switch bit	0: OFF 1: ON	0	○	○
5	AD external trigger valid bit	0: AD external trigger invalid 1: AD external trigger valid	0	○	○
6	Interrupt source selection bit	0: Interrupt request at A-D conversion completed 1: Interrupt request at ADT input falling	0	○	○
7	DA output enable bit	0: DA output disabled 1: DA output enabled	0	○	○

Notes 1: When setting “0” to bit 3, A-D converter is started. Writing “0” to bit 3 is valid, but if writing “1” to bit 3, bit 3 is not set to “1”. Accordingly, set “1” to bit 3 when a value is written to the A-D control register with bit 3 no affected.

2: When inputting the falling signal to the ADT pin during A-D conversion, A-D conversion which is in progress is stopped, and A-D conversion is started again.

Fig. 3.5.32 Structure of A-D control register

A-D/D-A conversion register



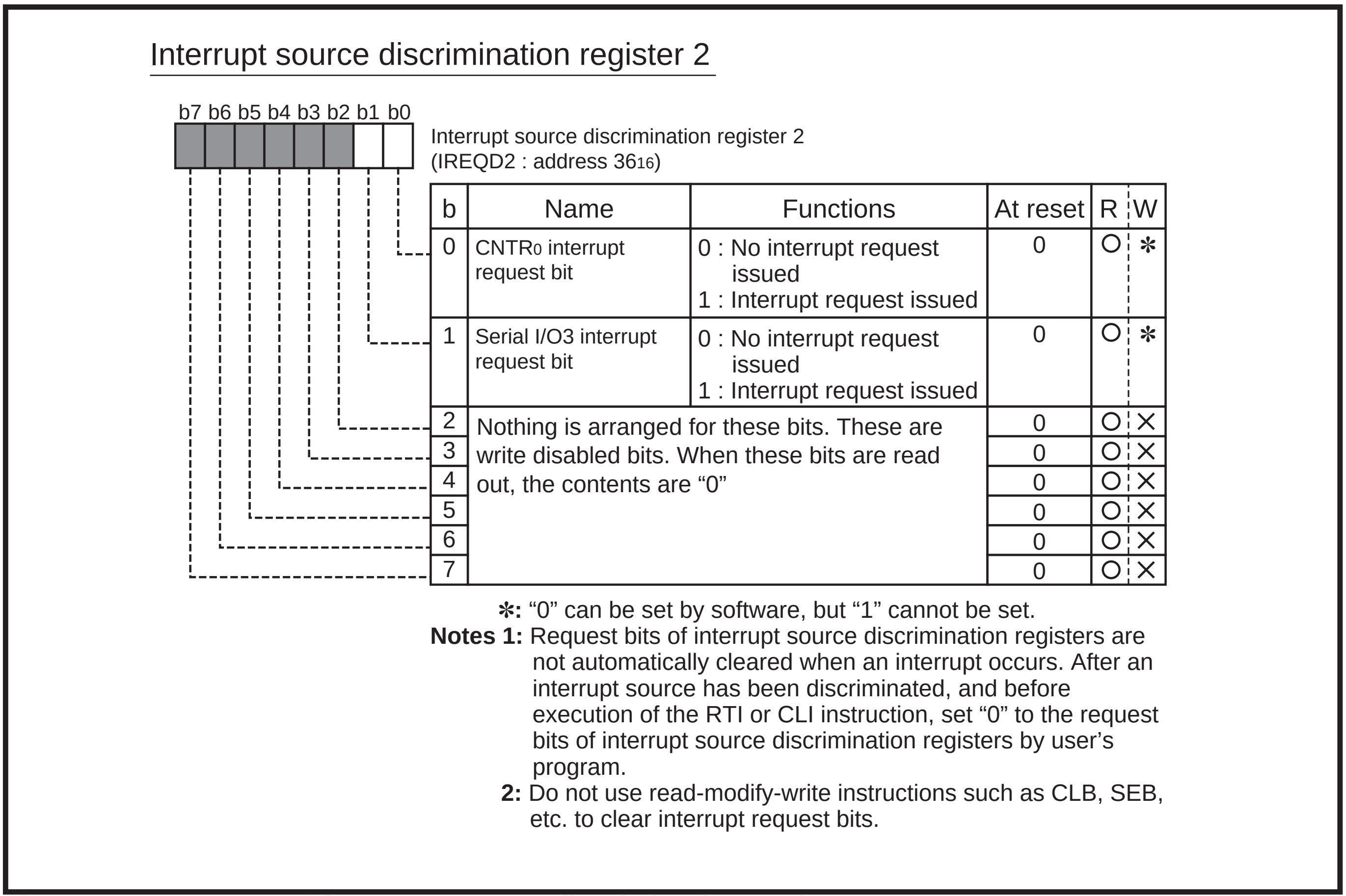
A-D/D-A conversion register
(AD: address 35_{16})

b	Functions	At reset	R	W
0	• The A-D conversion register and the D-A conversion register are assigned to the same address. • When using the A-D converter, the result of an A-D conversion is stored in this register. This register can be read only when using the A-D converter. • When using the D-A converter, this is started by setting a value in this register. This register can be written only when using the D-A converter.	**(Note)**	○	○
1			○	○
2			○	○
3			○	○
4			○	○
5			○	○
6			○	○
7			○	○

Notes 1 : When using A-D converter: Undefined
When using D-A converter: “0”

2 : The A-D/D-A conversion register functions as an A-D conversion register during a read and a D-A conversion register during a write. Accordingly, the D-A conversion register set value cannot be read out.

Fig. 3.5.33 Structure of A-D/D-A conversion register



Interrupt source discrimination register 2

Interrupt source discrimination register 2
(IREQD2 : address 36_{16})

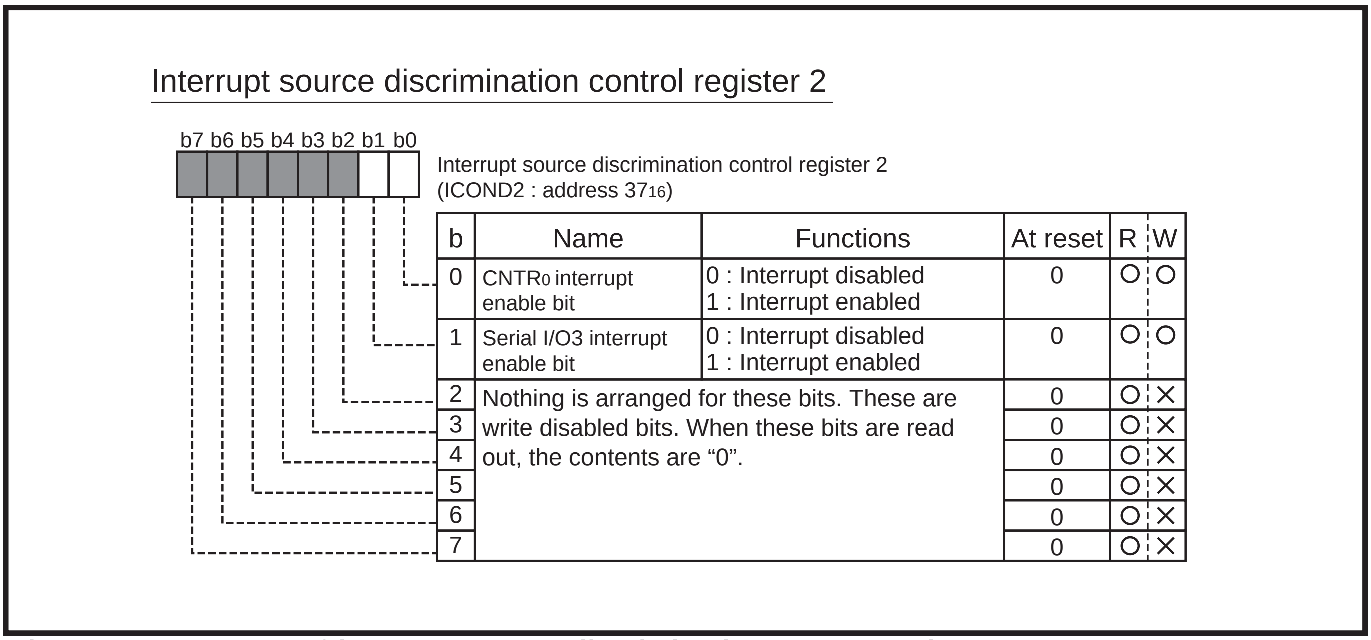
b	Name	Functions	At reset	R	W
0	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Serial I/O3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

✻: “0” can be set by software, but “1” cannot be set.

Notes 1: Request bits of interrupt source discrimination registers are not automatically cleared when an interrupt occurs. After an interrupt source has been discriminated, and before execution of the RTI or CLI instruction, set “0” to the request bits of interrupt source discrimination registers by user’s program.

2: Do not use read-modify-write instructions such as CLB, SEB, etc. to clear interrupt request bits.

Fig. 3.5.34 Structure of interrupt source discrimination register 2



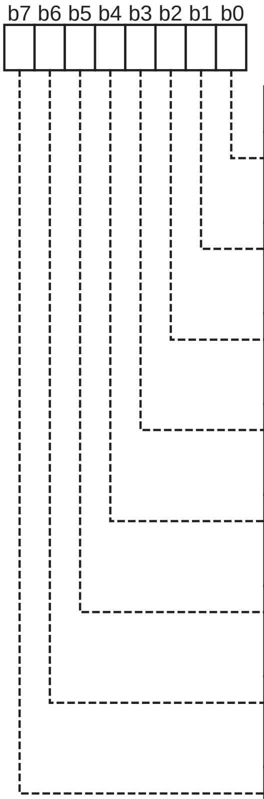
Interrupt source discrimination control register 2

Interrupt source discrimination control register 2
(ICOND2 : address 37_{16})

b	Name	Functions	At reset	R	W
0	CNTR0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Serial I/O3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.35 Structure of interrupt source discrimination control register 2

Interrupt source discrimination register 1



Interrupt source discrimination register 1
(IREQD1 : address 38$_{16}$)

b	Name	Functions	At reset	R	W
0	INT$_3$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT$_4$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	INT$_5$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Serial I/O1 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key input interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	ADT/A-D converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

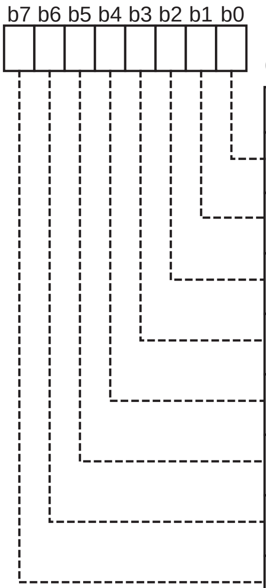
✻ : “0” can be set by software, but “1” cannot be set.

Notes 1: Request bits of interrupt source discrimination registers are not automatically cleared when an interrupt occurs. After an interrupt source has been discriminated, and before execution of the RTI or CLI instruction, set “0” to the request bits of interrupt source discrimination registers by user’s program.

2: Do not use read-modify-write instructions such as CLB, SEB, etc. to clear interrupt request bits.

Fig. 3.5.36 Structure of interrupt source discrimination register 1

Interrupt source discrimination control register 1

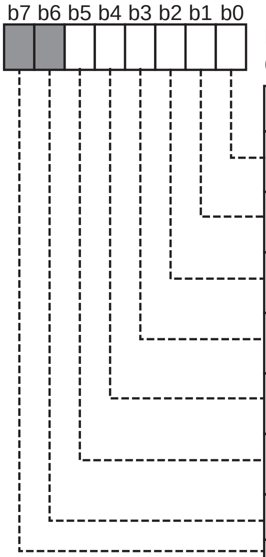


Interrupt source discrimination control register 1
(ICOND1 : address 39_{16})

b	Name	Functions	At reset	R	W
0	INT$_3$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT$_4$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT$_5$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Serial I/O1 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key input interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	ADT/A-D converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 3.5.37 Structure of interrupt source discrimination control register 1

Interrupt edge selection register

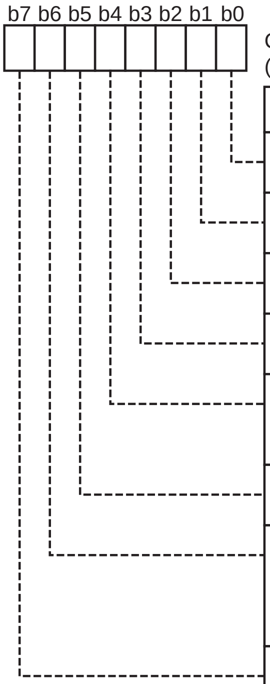


Interrupt edge selection register
(INTEDGE : address $3A_{16}$)

b	Name	Functions	At reset	R	W
0	INT$_0$ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT$_1$ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	INT$_2$ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	INT$_3$ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4	INT$_4$ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
5	INT$_5$ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	×
6	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
7					

Fig. 3.5.38 Structure of interrupt edge selection register

CPU mode register



CPU mode register
(CPUM: address $3B_{16}$)

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : 10 : } Not available 11 :	0	○	○
1			0	○	○
2	Stack page selection bit	0 : Page 0 1 : Page 1	0	○	○
3	X_{COUT} drivability selection bit	0: Low drive 1: High drive	1	○	○
4	Port Xc switch bit	0: I/O port function 1: X_{CIN}-X_{COUT} oscillating function	0	○	○
5	Main clock (X_{IN}-X_{OUT}) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bits	b7 b6 00 : ϕ = f(X_{IN})/2 (high-speed mode) 01 : ϕ = f(X_{IN})/8 (middle-speed mode) 10 : ϕ = f(X_{CIN})/2 (low-speed mode) 11 : ϕ = f(X_{IN}) (double-speed mode)	1	○	○
7			0	○	○

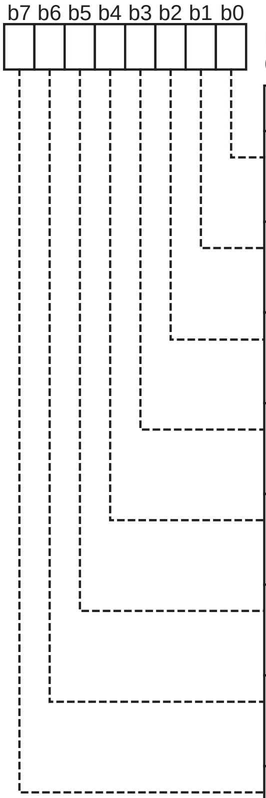
Notes 1: Use the LDM, STA, etc. instructions to modify the division ratio of the internal clock ϕ. (Do not use read-modify-write instructions such as CLB, SEB, etc.)

2: Do not modify the division ratio of the internal system clock during the cycle which is necessary for switching mode. For details concerning the number of cycles necessary to change modes, refer to the clock section in the explanation of about function blocks of **"CHAPTER 1. HARDWARE"**.

3: Do not switch the mode from low-speed to double-speed directly. Insert "clock switch timing wait" for switching the mode to middle/high-speed mode, and then switch the mode to double-speed.

Fig. 3.5.39 Structure of CPU mode register

Interrupt request register 1



Interrupt request register 1
(IREQ1 : address $3C_{16}$)

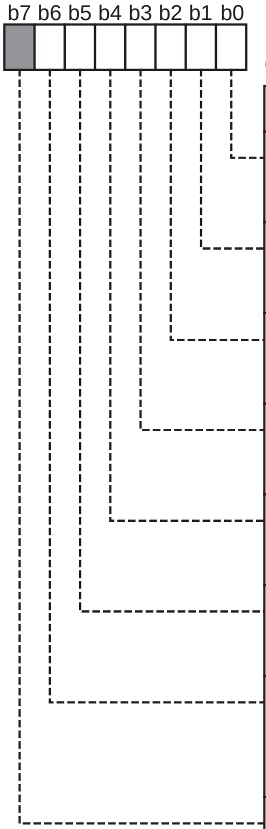
b	Name	Functions	At reset	R	W
0	INT$_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT$_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Receive bus interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Transmit bus interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: “0” can be set by software, but “1” cannot be set.

Note: 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” by using a data transfer instruction, execute one or more instructions before executing the BBC or BBS instruction.

Fig. 3.5.40 Structure of interrupt request register 1

Interrupt request register 2



Interrupt request register 2
(IREQ2 : address 3D$_{16}$)

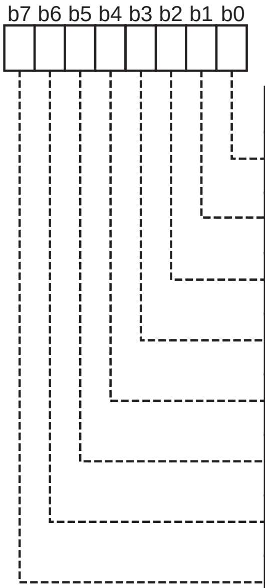
b	Name	Functions	At reset	R	W
0	INT$_2$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR$_0$, serial I/O3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	CNTR$_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT$_3$, INT$_4$, INT$_5$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	ADT/A-D converter, serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Key input, serial I/O1 receive, serial I/O1 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is write disabled bit. When this bit is read out, the contents is “0”.		0	○	×

✻: “0” can be set by software, but “1” cannot be set.

Note: 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” by using a data transfer instruction, execute one or more instructions before executing the BBC or BBS instruction.

Fig. 3.5.41 Structure of interrupt request register 2

Interrupt control register 1

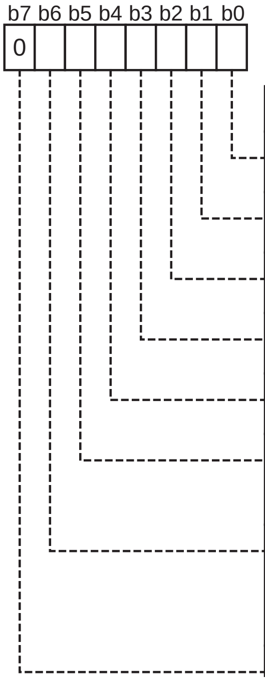


Interrupt control register 1
(ICON1 : address 3E$_{16}$)

b	Name	Functions	At reset	R	W
0	INT$_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT$_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Receive bus interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Transmit bus interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 3.5.42 Structure of interrupt control register 1

Interrupt control register 2



Interrupt control register 2
(ICON2 : address 3F$_{16}$)

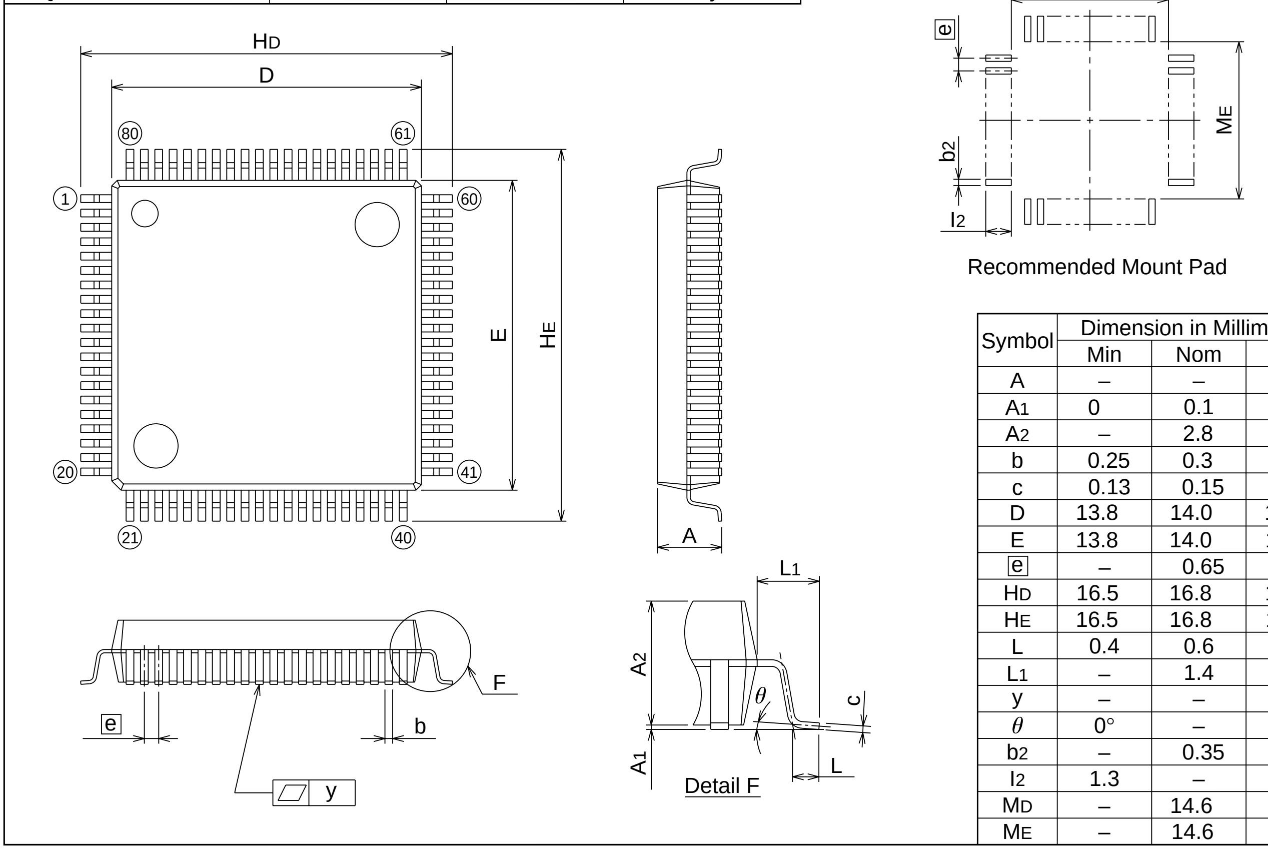
b	Name	Functions	At reset	R	W
0	INT$_2$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
1	CNTR$_0$, serial I/O3 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
2	CNTR$_1$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer 1 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT$_3$, INT$_4$, INT$_5$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
5	ADT/A-D converter, serial I/O2 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
6	Key input, serial I/O1 receive, serial I/O1 transmit interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix “0” to this bit.		0	○	○

Fig. 3.5.43 Structure of interrupt control register 2

3.6 Package outline

80P6S-A **Plastic 80pin 14✕14mm body QFP**

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP80-P-1414-0.65		1.11	Alloy 42



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A_1	0	0.1	0.2
A_2	–	2.8	–
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	13.8	14.0	14.2
$\boxed{e}$	–	0.65	–
H_D	16.5	16.8	17.1
H_E	16.5	16.8	17.1
L	0.4	0.6	0.8
L_1	–	1.4	–
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.35	–
l_2	1.3	–	–
M_D	–	14.6	–
M_E	–	14.6	–

3.7 List of instruction code

	D_3 – D_0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
D_7 – D_4	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	MUL ZP, X	BBS 3, A	TST ZP	ADC ZP	ROR ZP	BBS 3, ZP	PLA	ADC IMM	ROR A	SEB 3, A	JMP IND	ADC ABS	ROR ABS	SEB 3, ZP
0111	7	BVS	ADC IND, Y	—	BBC 3, A	—	ADC ZP, X	ROR ZP, X	BBC 3, ZP	SEI	ADC ABS, Y	—	CLB 3, A	—	ADC ABS, X	ROR ABS, X	CLB 3, ZP
1000	8	BRA	STA IND, X	RRF ZP	BBS 4, A	STY ZP	STA ZP	STX ZP	BBS 4, ZP	DEY	—	TXA	SEB 4, A	STY ABS	STA ABS	STX ABS	SEB 4, ZP
1001	9	BCC	STA IND, Y	—	BBC 4, A	STY ZP, X	STA ZP, X	STX ZP, Y	BBC 4, ZP	TYA	STA ABS, Y	TXS	CLB 4, A	—	STA ABS, X	—	CLB 4, ZP
1010	A	LDY IMM	LDA IND, X	LDX IMM	BBS 5, A	LDY ZP	LDA ZP	LDX ZP	BBS 5, ZP	TAY	LDA IMM	TAX	SEB 5, A	LDY ABS	LDA ABS	LDX ABS	SEB 5, ZP
1011	B	BCS	LDA IND, Y	JMP ZP, IND	BBC 5, A	LDY ZP, X	LDA ZP, X	LDX ZP, Y	BBC 5, ZP	CLV	LDA ABS, Y	TSX	CLB 5, A	LDY ABS, X	LDA ABS, X	LDX ABS, Y	CLB 5, ZP
1100	C	CPY IMM	CMP IND, X	WIT	BBS 6, A	CPY ZP	CMP ZP	DEC ZP	BBS 6, ZP	INY	CMP IMM	DEX	SEB 6, A	CPY ABS	CMP ABS	DEC ABS	SEB 6, ZP
1101	D	BNE	CMP IND, Y	—	BBC 6, A	—	CMP ZP, X	DEC ZP, X	BBC 6, ZP	CLD	CMP ABS, Y	—	CLB 6, A	—	CMP ABS, X	DEC ABS, X	CLB 6, ZP
1110	E	CPX IMM	SBC IND, X	DIV ZP, X	BBS 7, A	CPX ZP	SBC ZP	INC ZP	BBS 7, ZP	INX	SBC IMM	NOP	SEB 7, A	CPX ABS	SBC ABS	INC ABS	SEB 7, ZP
1111	F	BEQ	SBC IND, Y	—	BBC 7, A	—	SBC ZP, X	INC ZP, X	BBC 7, ZP	SED	SBC ABS, Y	—	CLB 7, A	—	SBC ABS, X	INC ABS, X	CLB 7, ZP

▩ : 3-byte instruction

▒ : 2-byte instruction

□ : 1-byte instruction

3.8 Machine instructions

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
ADC (Note 1) (Note 5)	When T = 0 A ← A + M + C When T = 1 M(X) ← M(X) + M + C	When T = 0, this instruction adds the contents M, C, and A; and stores the results in A and C. When T = 1, this instruction adds the contents of M(X), M and C; and stores the results in M(X) and C. When T=1, the contents of A remain unchanged, but the contents of status flags are changed. M(X) represents the contents of memory where is indicated by X.				69	2	2							65	3	2			
AND (Note 1)	When T = 0 A ← A ∧ M When T = 1 M(X) ← M(X) ∧ M	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise AND operation and stores the result back in A. When T = 1, this instruction transfers the contents M(X) and M to the ALU which performs a bit-wise AND operation and stores the results back in M(X). When T = 1, the contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				29	2	2							25	3	2			
ASL	7 0 C ← □ ← 0	This instruction shifts the content of A or M by one bit to the left, with bit 0 always being set to 0 and bit 7 of A or M always being contained in C.							0A	2	1				06	5	2			
BBC (Note 4)	Ai or Mi = 0?	This instruction tests the designated bit i of M or A and takes a branch if the bit is 0. The branch address is specified by a relative address. If the bit is 1, next instruction is executed.										13 + 20i	4	2				17 + 20i	5	3
BBS (Note 4)	Ai or Mi = 1?	This instruction tests the designated bit i of the M or A and takes a branch if the bit is 1. The branch address is specified by a relative address. If the bit is 0, next instruction is executed.										03 + 20i	4	2				07 + 20i	5	3
BCC (Note 4)	C = 0?	This instruction takes a branch to the appointed address if C is 0. The branch address is specified by a relative address. If C is 1, the next instruction is executed.																		
BCS (Note 4)	C = 1?	This instruction takes a branch to the appointed address if C is 1. The branch address is specified by a relative address. If C is 0, the next instruction is executed.																		
BEQ (Note 4)	Z = 1?	This instruction takes a branch to the appointed address when Z is 1. The branch address is specified by a relative address. If Z is 0, the next instruction is executed.																		
BIT	A ∧ M	This instruction takes a bit-wise logical AND of A and M contents; however, the contents of A and M are not modified. The contents of N, V, Z are changed, but the contents of A, M remain unchanged.													24	3	2			
BMI (Note 4)	N = 1?	This instruction takes a branch to the appointed address when N is 1. The branch address is specified by a relative address. If N is 0, the next instruction is executed.																		
BNE (Note 4)	Z = 0?	This instruction takes a branch to the appointed address if Z is 0. The branch address is specified by a relative address. If Z is 1, the next instruction is executed.																		

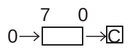
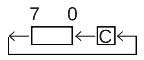
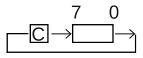
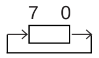
Symbol	Addressing mode																																	Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
ADC	75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2							N	V	•	•	•	•	Z	C
AND	35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2							N	•	•	•	•	•	Z	•
ASL	16	6	2				0E	6	3	1E	7	3																						N	•	•	•	•	•	Z	C
BBC																																		•	•	•	•	•	•	•	•
BBS																																		•	•	•	•	•	•	•	•
BCC																												90	2	2				•	•	•	•	•	•	•	•
BCS																												B0	2	2				•	•	•	•	•	•	•	•
BEQ																												F0	2	2				•	•	•	•	•	•	•	•
BIT							2C	4	3																									M_7	M_6	•	•	•	•	Z	•
BMI																												30	2	2				•	•	•	•	•	•	•	•
BNE																												D0	2	2				•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
BPL (Note 4)	N = 0?	This instruction takes a branch to the appointed address if N is 0. The branch address is specified by a relative address. If N is 1, the next instruction is executed.																		
BRA	PC ← PC ± offset	This instruction branches to the appointed address. The branch address is specified by a relative address.																		
BRK	B ← 1 (PC) ← (PC) + 2 M(S) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 M(S) ← PS S ← S – 1 I← 1 PC_L ← AD_L PC_H ← AD_H	When the BRK instruction is executed, the CPU pushes the current PC contents onto the stack. The BADRS designated in the interrupt vector table is stored into the PC.	00	7	1															
BVC (Note 4)	V = 0?	This instruction takes a branch to the appointed address if V is 0. The branch address is specified by a relative address. If V is 1, the next instruction is executed.																		
BVS (Note 4)	V = 1?	This instruction takes a branch to the appointed address when V is 1. The branch address is specified by a relative address. When V is 0, the next instruction is executed.																		
CLB	Ai or Mi ← 0	This instruction clears the designated bit i of A or M.										1B + 20i	2	1				1F + 20i	5	2
CLC	C ← 0	This instruction clears C.	18	2	1															
CLD	D ← 0	This instruction clears D.	D8	2	1															
CLI	I ← 0	This instruction clears I.	58	2	1															
CLT	T ← 0	This instruction clears T.	12	2	1															
CLV	V ← 0	This instruction clears V.	B8	2	1															
CMP (Note 3)	When T = 0 A – M When T = 1 M(X) – M	When T = 0, this instruction subtracts the contents of M from the contents of A. The result is not stored and the contents of A or M are not modified. When T = 1, the CMP subtracts the contents of M from the contents of M(X). The result is not stored and the contents of X, M, and A are not modified. M(X) represents the contents of memory where is indicated by X.				C9	2	2							C5	3	2			
COM	M ← $\overline{M}$	This instruction takes the one's complement of the contents of M and stores the result in M.													44	5	2			
CPX	X – M	This instruction subtracts the contents of M from the contents of X. The result is not stored and the contents of X and M are not modified.				E0	2	2							E4	3	2			
CPY	Y – M	This instruction subtracts the contents of M from the contents of Y. The result is not stored and the contents of Y and M are not modified.				C0	2	2							C4	3	2			
DEC	A ← A – 1 or M ← M – 1	This instruction subtracts 1 from the contents of A or M.							1A	2	1				C6	5	2			

Symbol	Addressing mode																																Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
BPL																												10	2	2				•	•	•	•	•	•	•	•
BRA																												80	4	2				•	•	•	•	•	•	•	•
BRK																																		•	•	•	1	•	1	•	•
BVC																												50	2	2				•	•	•	•	•	•	•	•
BVS																												70	2	2				•	•	•	•	•	•	•	•
CLB																																		•	•	•	•	•	•	•	•
CLC																																		•	•	•	•	•	•	•	0
CLD																																		•	•	•	•	0	•	•	•
CLI																																		•	•	•	•	•	0	•	•
CLT																																		•	•	0	•	•	•	•	•
CLV																																		•	0	•	•	•	•	•	•
CMP	D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
COM																																		N	•	•	•	•	•	Z	•
CPX							EC	4	3																									N	•	•	•	•	•	Z	C
CPY							CC	4	3																									N	•	•	•	•	•	Z	C
DEC	D6	6	2				CE	6	3	DE	7	3																						N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
DEX	X ← X – 1	This instruction subtracts one from the current contents of X.	CA	2	1															
DEY	Y ← Y – 1	This instruction subtracts one from the current contents of Y.	88	2	1															
DIV	A ← (M(zz + X + 1), M(zz + X)) / A M(S) ← one's complement of Remainder S ← S – 1	This instruction divides the 16-bit data in M(zz+(X)) (low-order byte) and M(zz+(X)+1) (high-order byte) by the contents of A. The quotient is stored in A and the one's complement of the remainder is pushed onto the stack.																		
EOR (Note 1)	When T = 0 A ← A ∀ M When T = 1 M(X) ← M(X) ∀ M	When T = 0, this instruction transfers the contents of the M and A to the ALU which performs a bit-wise Exclusive OR, and stores the result in A. When T = 1, the contents of M(X) and M are transferred to the ALU, which performs a bit-wise Exclusive OR and stores the results in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				49	2	2							45	3	2			
INC	A ← A + 1 or M ← M + 1	This instruction adds one to the contents of A or M.							3A	2	1				E6	5	2			
INX	X ← X + 1	This instruction adds one to the contents of X.	E8	2	1															
INY	Y ← Y + 1	This instruction adds one to the contents of Y.	C8	2	1															
JMP	If addressing mode is ABS $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is IND $PC_L \leftarrow M(AD_H, AD_L)$ $PC_H \leftarrow M(AD_H, AD_L + 1)$ If addressing mode is ZP, IND $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	This instruction jumps to the address designated by the following three addressing modes: Absolute Indirect Absolute Zero Page Indirect Absolute																		
JSR	$M(S) \leftarrow PC_H$ S ← S – 1 $M(S) \leftarrow PC_L$ S ← S – 1 After executing the above, if addressing mode is ABS, $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ if addressing mode is SP, $PC_L \leftarrow AD_L$ $PC_H \leftarrow FF$ If addressing mode is ZP, IND, $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	This instruction stores the contents of the PC in the stack, then jumps to the address designated by the following addressing modes: Absolute Special Page Zero Page Indirect Absolute																		
LDA (Note 2)	When T = 0 A ← M When T = 1 M(X) ← M	When T = 0, this instruction transfers the contents of M to A. When T = 1, this instruction transfers the contents of M to (M(X)). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				A9	2	2							A5	3	2			
LDM	M ← nn	This instruction loads the immediate value in M.													3C	4	3			
LDX	X ← M	This instruction loads the contents of M in X.				A2	2	2							A6	3	2			
LDY	Y ← M	This instruction loads the contents of M in Y.				A0	2	2							A4	3	2			

Symbol	Addressing mode																																	Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
DEX																																		N	•	•	•	•	•	Z	•
DEY																																		N	•	•	•	•	•	Z	•
DIV	E2	16	2																															•	•	•	•	•	•	•	•
EOR	55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
INC	F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
INX																																		N	•	•	•	•	•	Z	•
INY																																		N	•	•	•	•	•	Z	•
JMP							4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
JSR							20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
LDA	B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
LDM																																		•	•	•	•	•	•	•	•
LDX				B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
LDY	B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
LSR		This instruction shifts either A or M one bit to the right such that bit 7 of the result always is set to 0, and the bit 0 is stored in C.							4A	2	1				46	5	2			
MUL	M(S) • A ← A ✻ M(zz + X) S ← S – 1	This instruction multiply Accumulator with the memory specified by the Zero Page X address mode and stores the high-order byte of the result on the Stack and the low-order byte in A.																		
NOP	PC ← PC + 1	This instruction adds one to the PC but does no otheroperation.	EA	2	1															
ORA (Note 1)	When T = 0 A ← A V M When T = 1 M(X) ← M(X) V M	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise “OR”, and stores the result in A. When T = 1, this instruction transfers the contents of M(X) and the M to the ALU which performs a bit-wise OR, and stores the result in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				09	2	2							05	3	2			
PHA	S ← S – 1	This instruction pushes the contents of A to the memory location designated by S, and decrements the contents of S by one.	48	3	1															
PHP	M(S) ← PS S ← S – 1	This instruction pushes the contents of PS to the memory location designated by S and decrements the contents of S by one.	08	3	1															
PLA	S ← S + 1 A ← M(S)	This instruction increments S by one and stores the contents of the memory designated by S in A.	68	4	1															
PLP	S ← S + 1 PS ← M(S)	This instruction increments S by one and stores the contents of the memory location designated by S in PS.	28	4	1															
ROL		This instruction shifts either A or M one bit left through C. C is stored in bit 0 and bit 7 is stored in C.							2A	2	1				26	5	2			
ROR		This instruction shifts either A or M one bit right through C. C is stored in bit 7 and bit 0 is stored in C.							6A	2	1				66	5	2			
RRF		This instruction rotates 4 bits of the M content to the right.													82	8	2			
RTI	S ← S + 1 PS ← M(S) S ← S + 1 PC_L ← M(S) S ← S + 1 PC_H ← M(S)	This instruction increments S by one, and stores the contents of the memory location designated by S in PS. S is again incremented by one and stores the contents of the memory location designated by S in PC_L. S is again incremented by one and stores the contents of memory location designated by S in PC_H.	40	6	1															
RTS	S ← S + 1 PC_L ← M(S) S ← S + 1 PC_H ← M(S) (PC) ← (PC) + 1	This instruction increments S by one and stores the contents of the memory location designated by S in PC_L. S is again incremented by one and the contents of the memory location is stored in PC_H. PC is incremented by 1.	60	6	1															

Symbol	Addressing mode																																Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
LSR	56	6	2				4E	6	3	5E	7	3																						0	•	•	•	•	•	Z	C
MUL	62	15	2																															•	•	•	•	•	•	•	•
NOP																																		•	•	•	•	•	•	•	•
ORA	15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•
PHA																																		•	•	•	•	•	•	•	•
PHP																																		•	•	•	•	•	•	•	•
PLA																																		N	•	•	•	•	•	Z	•
PLP																																		(Value saved in stack)							
ROL	36	6	2				2E	6	3	3E	7	3																						N	•	•	•	•	•	Z	C
ROR	76	6	2				6E	6	3	7E	7	3																						N	•	•	•	•	•	Z	C
RRF																																		•	•	•	•	•	•	•	•
RTI																																		(Value saved in stack)							
RTS																																		•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
SBC (Note 1) (Note 5)	When T = 0 $A \leftarrow A - M - \overline{C}$ When T = 1 $M(X) \leftarrow M(X) - M - \overline{C}$	When T = 0, this instruction subtracts the value of M and the complement of C from A, and stores the results in A and C. When T = 1, the instruction subtracts the contents of M and the complement of C from the contents of M(X), and stores the results in M(X) and C. A remain unchanged, but status flag are changed. M(X) represents the contents of memory where is indicated by X.				E9	2	2							E5	3	2			
SEB	Ai or Mi ← 1	This instruction sets the designated bit i of A or M.										0B + 20i	2	1				0F + 20i	5	2
SEC	C ← 1	This instruction sets C.	38	2	1															
SED	D ← 1	This instruction set D.	F8	2	1															
SEI	I ← 1	This instruction set I.	78	2	1															
SET	T ← 1	This instruction set T.	32	2	1															
STA	M ← A	This instruction stores the contents of A in M. The contents of A does not change.													85	4	2			
STP		This instruction resets the oscillation control F/F and the oscillation stops. Reset or interrupt input is needed to wake up from this mode.	42	2	1															
STX	M ← X	This instruction stores the contents of X in M. The contents of X does not change.													86	4	2			
STY	M ← Y	This instruction stores the contents of Y in M. The contents of Y does not change.													84	4	2			
TAX	X ← A	This instruction stores the contents of A in X. The contents of A does not change.	AA	2	1															
TAY	Y ← A	This instruction stores the contents of A in Y. The contents of A does not change.	A8	2	1															
TST	M = 0?	This instruction tests whether the contents of M are "0" or not and modifies the N and Z.													64	3	2			
TSX	X ← S	This instruction transfers the contents of S in X.	BA	2	1															
TXA	A ← X	This instruction stores the contents of X in A.	8A	2	1															
TXS	S ← X	This instruction stores the contents of X in S.	9A	2	1															
TYA	A ← Y	This instruction stores the contents of Y in A.	98	2	1															
WIT		The WIT instruction stops the internal clock but not the oscillation of the oscillation circuit is not stopped. CPU starts its function after the Timer X over flows (comes to the terminal count). All registers or internal memory contents except Timer X will not change during this mode. (Of course needs VDD).	C2	2	1															

Notes 1 : The number of cycles "n" is increased by 3 when T is 1.
2 : The number of cycles "n" is increased by 2 when T is 1.
3 : The number of cycles "n" is increased by 1 when T is 1.
4 : The number of cycles "n" is increased by 2 when branching has occurred.
5 : N, V, and Z flags are invalid in decimal operation mode.

Symbol	Addressing mode																																Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
SBC	F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
SEB																																		•	•	•	•	•	•	•	•
SEC																																		•	•	•	•	•	•	•	1
SED																																		•	•	•	•	1	•	•	•
SEI																																		•	•	•	•	•	1	•	•
SET																																		•	•	1	•	•	•	•	•
STA	95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
STP																																		•	•	•	•	•	•	•	•
STX				96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
STY	94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
TAX																																		N	•	•	•	•	•	Z	•
TAY																																		N	•	•	•	•	•	Z	•
TST																																		N	•	•	•	•	•	Z	•
TSX																																		N	•	•	•	•	•	Z	•
TXA																																		N	•	•	•	•	•	Z	•
TXS																																		•	•	•	•	•	•	•	•
TYA																																		N	•	•	•	•	•	Z	•
WIT																																		•	•	•	•	•	•	•	•

3.8 Machine instructions

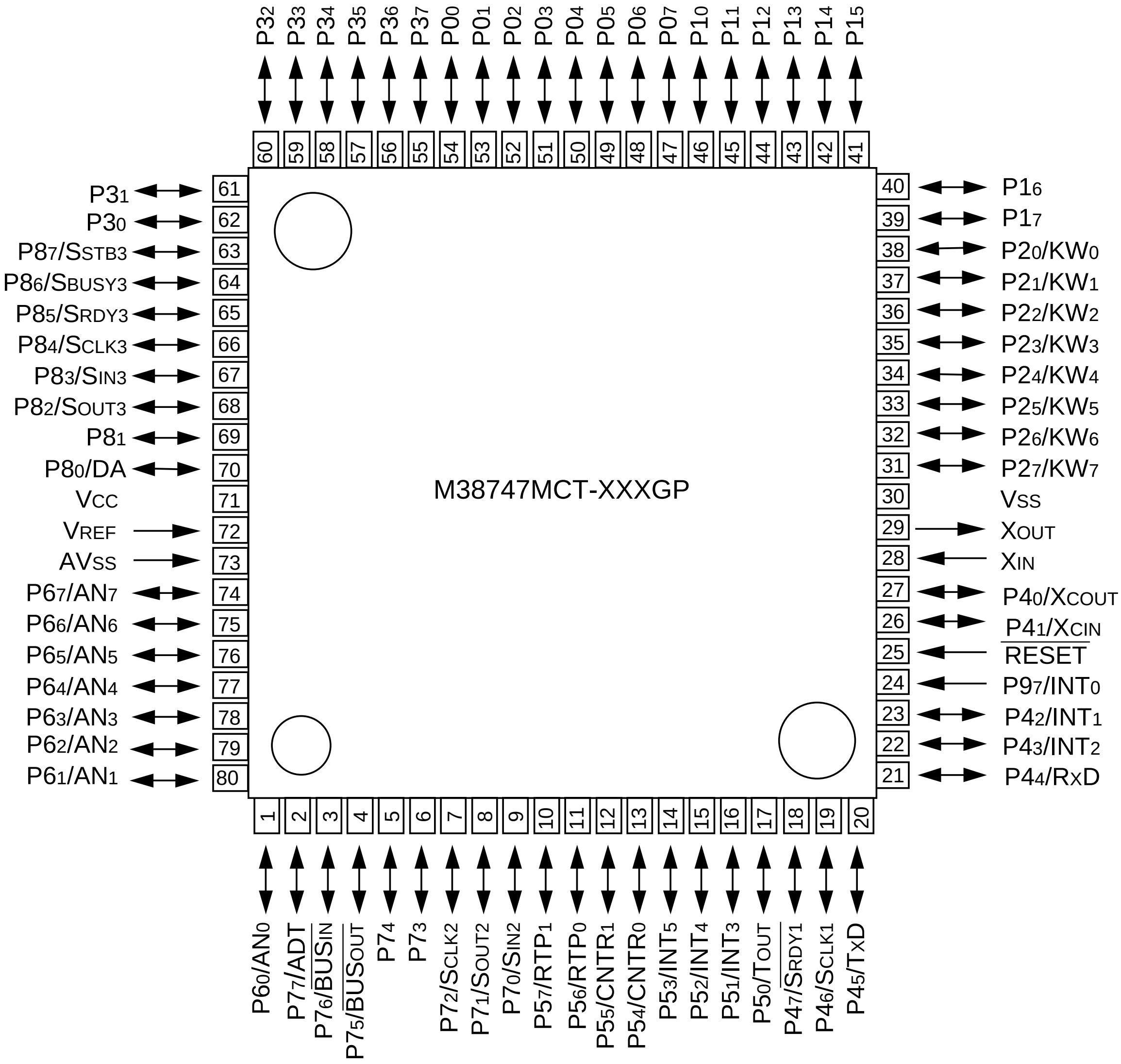
Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	–	Subtraction
A	Accumulator or Accumulator addressing mode	✻	Multiplication
BIT, A	Accumulator bit addressing mode	/	Division
BIT, A, R	Accumulator bit relative addressing mode	Λ	Logical OR
ZP	Zero page addressing mode	V	Logical AND
BIT, ZP	Zero page bit addressing mode	∀	Logical exclusive OR
BIT, ZP, R	Zero page bit relative addressing mode	—	Negation
ZP, X	Zero page X addressing mode	←	Shows direction of data flow
ZP, Y	Zero page Y addressing mode	X	Index register X
ABS	Absolute addressing mode	Y	Index register Y
ABS, X	Absolute X addressing mode	S	Stack pointer
ABS, Y	Absolute Y addressing mode	PC	Program counter
IND	Indirect absolute addressing mode	PS	Processor status register
		PC$_H$	8 high-order bits of program counter
ZP, IND	Zero page indirect absolute addressing mode	PC$_L$	8 low-order bits of program counter
		AD$_H$	8 high-order bits of address
IND, X	Indirect X addressing mode	AD$_L$	8 low-order bits of address
IND, Y	Indirect Y addressing mode	FF	FF in Hexadecimal notation
REL	Relative addressing mode	nn	Immediate value
SP	Special page addressing mode	zz	Zero page address
C	Carry flag	M	Memory specified by address designation of any addressing mode
Z	Zero flag		
I	Interrupt disable flag	M(X)	Memory of address indicated by contents of index register X
D	Decimal mode flag		
B	Break flag	M(S)	Memory of address indicated by contents of stack pointer
T	X-modified arithmetic mode flag		
V	Overflow flag	M(AD$_H$, AD$_L$)	Contents of memory at address indicated by AD$_H$ and AD$_L$, in AD$_H$ is 8 high-order bits and AD$_L$ is 8 low-order bits.
N	Negative flag		
		M(00, AD$_L$)	Contents of address indicated by zero page AD$_L$
		Ai	Bit i (i = 0 to 7) of accumulator
		Mi	Bit i (i = 0 to 7) of memory
		OP	Opcode
		n	Number of cycles
		#	Number of bytes

3.9 SFR memory map

0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	Port P5 (P5)
$000B_{16}$	Port P5 direction register (P5D)
$000C_{16}$	Port P6 (P6)
$000D_{16}$	Port P6 direction register (P6D)
$000E_{16}$	Port P7 (P7)
$000F_{16}$	Port P7 direction register (P7D)
0010_{16}	Port P8 (P8)
0011_{16}	Port P8 direction register (P8D)
0012_{16}	Port P9 (P9)
0013_{16}	Serial I/O3 register/Transfer counter (SIO3)
0014_{16}	Serial I/O3 control register 1 (SIO3CON1)
0015_{16}	Serial I/O3 control register 2 (SIO3CON2)
0016_{16}	Serial I/O3 control register 3 (SIO3CON3)
0017_{16}	Serial I/O3 automatic transfer data pointer (SIO3DP)
0018_{16}	Transmit/Receive buffer register (TB/RB)
0019_{16}	Serial I/O1 status register (SIO1STS)
$001A_{16}$	Serial I/O1 control register (SIO1CON)
$001B_{16}$	UART control register (UARTCON)
$001C_{16}$	Baud rate generator (BRG)
$001D_{16}$	Serial I/O2 control register (SIO2CON)
$001E_{16}$	Watchdog timer control register (WDTCON)
$001F_{16}$	Serial I/O2 register (SIO2)

0020_{16}	Timer X (low-order) (TXL)
0021_{16}	Timer X (high-order) (TXH)
0022_{16}	Timer Y (low-order) (TYL)
0023_{16}	Timer Y (high-order) (TYH)
0024_{16}	Timer 1 (T1)
0025_{16}	Timer 2 (T2)
0026_{16}	Timer 3 (T3)
0027_{16}	Timer X mode register (TXM)
0028_{16}	Timer Y mode register (TYM)
0029_{16}	Timer 123 mode register (T123M)
$002A_{16}$	Communication mode register (BUSM)
$002B_{16}$	Transmit control register (TXDCON)
$002C_{16}$	Transmit status register (TXDSTS)
$002D_{16}$	Receive control register (RXDCON)
$002E_{16}$	Receive status register (RXDSTS)
$002F_{16}$	Bus interrupt source discrimination control register (BICOND)
0030_{16}	Control field selection register (CFSEL)
0031_{16}	Control field register (CF)
0032_{16}	Transmit/Receive FIFO (TRFIFO)
0033_{16}	PULL UP register (PULLU)
0034_{16}	A-D control register (ADCON)
0035_{16}	A-D/D-A conversion register (AD)
0036_{16}	Interrupt source discrimination register 2 (IREQD2)
0037_{16}	Interrupt source discrimination control register 2 (ICOND2)
0038_{16}	Interrupt source discrimination register 1 (IREQD1)
0039_{16}	Interrupt source discrimination control register 1 (ICOND1)
$003A_{16}$	Interrupt edge selection register (INTEDGE)
$003B_{16}$	CPU mode register (CPUM)
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003D_{16}$	Interrupt request register 2 (IREQ2)
$003E_{16}$	Interrupt control register 1 (ICON1)
$003F_{16}$	Interrupt control register 2 (ICON2)

3.10 Pin configuration



MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
3874 Group

Sep. First Edition 1998

Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.
©1998 MITSUBISHI ELECTRIC CORPORATION

User's Manual
3874 Group

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: MITSUBISHI DENKI BLDG., MARUNOUCHI, TOKYO 100. TELEX: J24532 CABLE: MELCO TOKYO

© 1998 MITSUBISHI ELECTRIC CORPORATION.

New publication, effective Sep. 1998.
Specifications subject to change without notice.

REVISION DESCRIPTION LIST	3874 Group User’s Manual

Rev. No.	Revision Description	Rev. date
1.0	First Edition	980910