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

38C3 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.

REVISION DESCRIPTION LIST	38C3 Group User's Manual

Rev. No.	Revision Description	Rev. date
1.0	First Edition	990412

Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 38C3 Group.
After reading this manual, the user should have a through knowledge of the functions and features of the 38C3 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.

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**

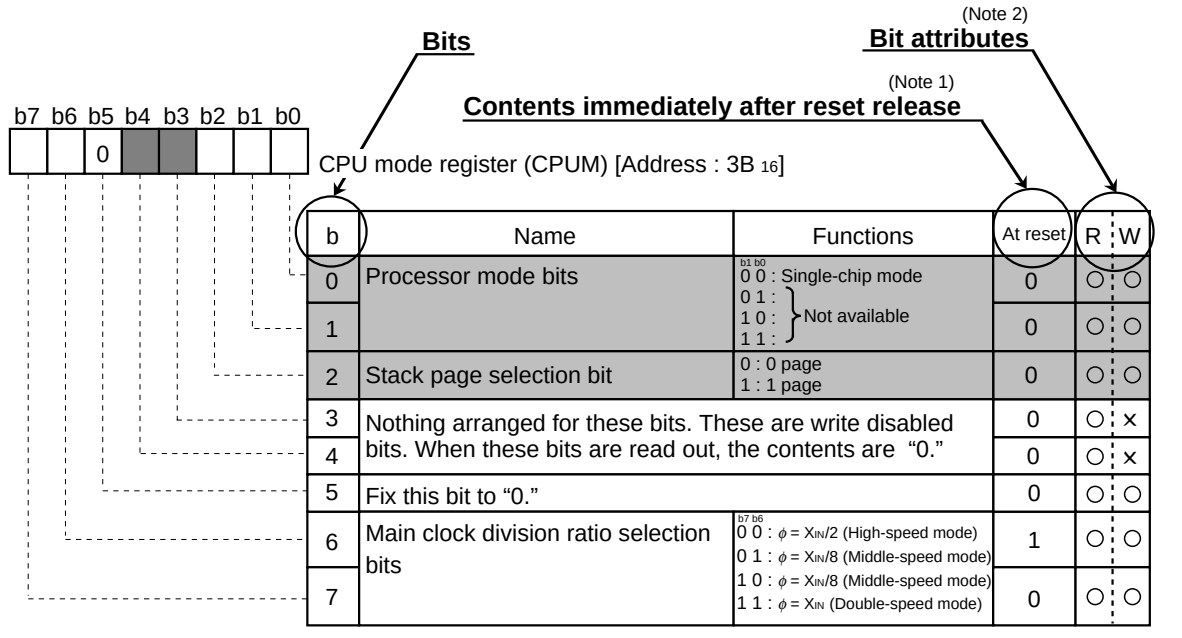
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of relevant registers.

● **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:



: 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
○••••••Read enabled
×••••••Read disabled

W••••••Write
○ ••••••Write enabled
× ••••••Write disabled

Table of contents

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

Table of contents

List of figures

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

List of figures

CHAPTER 3 APPENDIX

List of figures

List of tables

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

CHAPTER 1
HARDWARE

HARDWARE

DESCRIPTION/FEATURES/APPLICATION/PIN CONFIGURATION

DESCRIPTION

The 38C3 group is the 8-bit microcomputer based on the 740 family core technology.

The 38C3 group has a LCD drive control circuit, a 10-channel A-D converter, and a Serial I/O as additional functions.

The various microcomputers in the 38C3 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 38C3 group, refer to the section on group expansion.

FEATURES

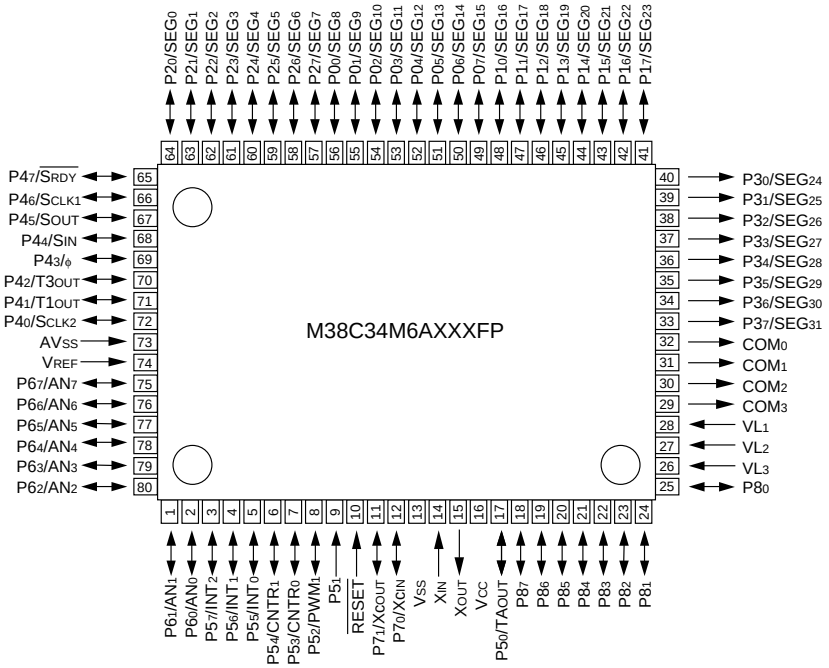
- Basic machine-language instructions 71
- The minimum instruction execution time ..... 0.5 µs
  (at 8MHz oscillation frequency)
- Memory size
  ROM ..... 4 K to 48 K bytes
  RAM ..... 192 to 1024 bytes
- Programmable input/output ports 57
- Software pull-up/pull-down resistors
  ..... (Ports P0–P8 except Port P5$_1$)
- Interrupts ..... 16 sources, 16 vectors
  (includes key input interrupt)
- Timers ..... 8-bit × 6, 16-bit × 1
- A-D converter ..... 10-bit × 8 channels
- Serial I/O ..... 8-bit × 1 (Clock-synchronized)
- LCD drive control circuit
  Bias 1/1, 1/2, 1/3
  Duty 1/1, 1/2, 1/3, 1/4
  Common output 4
  Segment output 32
- 2 Clock generating circuit
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage
  In high-speed mode ..... 4.0 to 5.5 V
  In middle-speed mode ..... 2.5 to 5.5 V
  (M version is 2.2* to 5.5 V)
  In low-speed mode ..... 2.5 to 5.5 V
  (M version is 2.2* to 5.5 V)
- Power dissipation
  In high-speed mode ..... 32 mW
  (at 8 MHz oscillation frequency)
  In low-speed mode ..... 45 µW
  (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Operating temperature range ..... – 20 to 85°C

* Mask ROM version only

APPLICATION

Camera, household appliances, consumer electronics, etc.

PIN CONFIGURATION (TOP VIEW)



Package type : 80P6N-A
80-pin plastic-molded QFP

Fig. 1 M38C34M6AXXXFP pin configuration

FUNCTIONAL BLOCK DIAGRAM

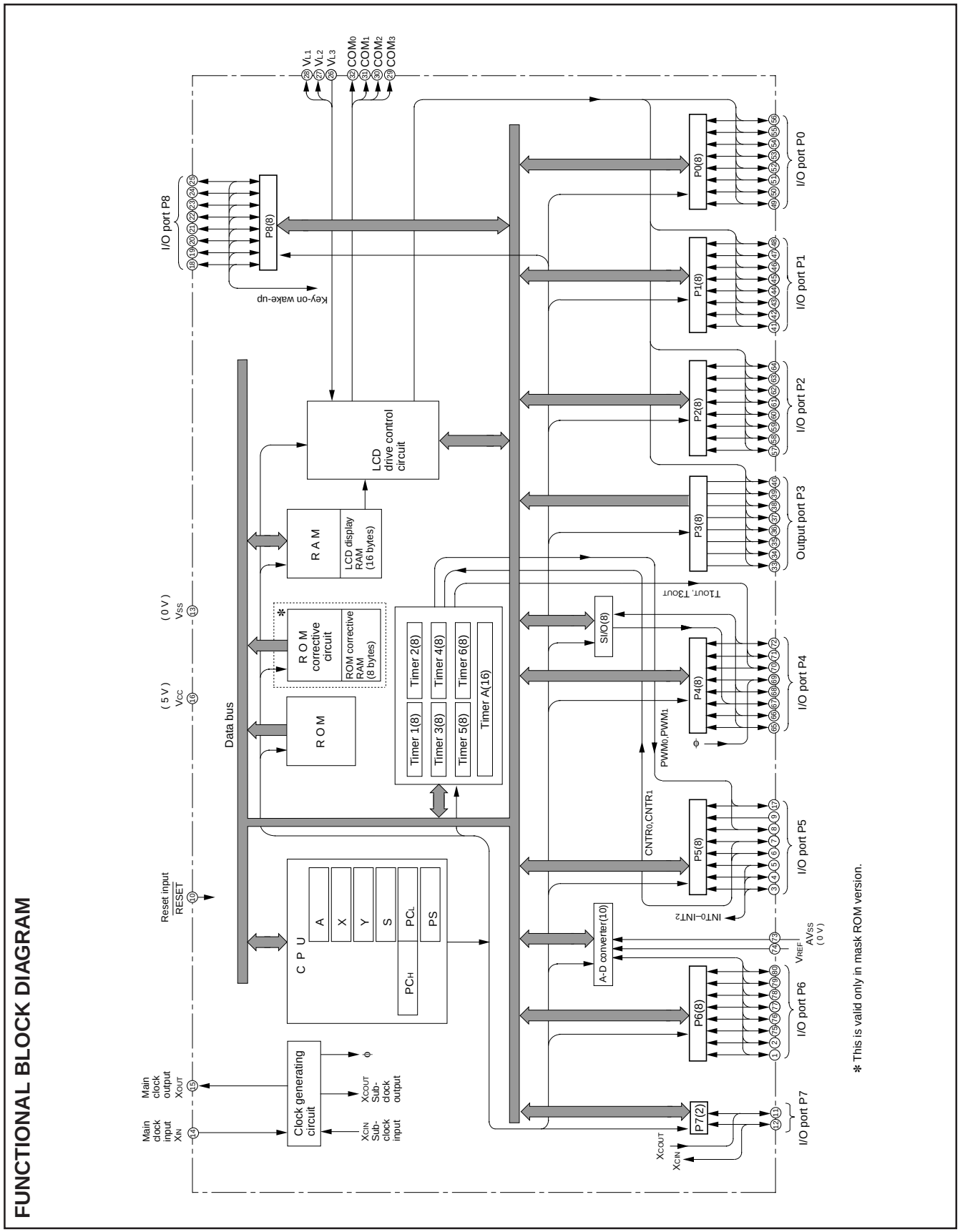


Fig. 2 Functional block diagram

PIN DESCRIPTION

PIN DESCRIPTION

Table 1 Pin description (1)

Pin	Name	Function	Function except a port function
V_{CC}, V_{SS}	Power source	• Apply voltage of 2.5* V to 5.5 V to V_{CC}, and 0 V to V_{SS}.	
V_{REF}	Analog reference voltage	• Reference voltage input pin for A-D converter.	
AV_{SS}	Analog power source	• GND input pin for A-D converter. • Connect to V_{SS}.	
$\overline{RESET}$	Reset input	• Reset input pin for active "L."	
X_{IN}	Clock input	• Input and output pins for the main clock generating circuit. • Feedback resistor is built in between X_{IN} pin and X_{OUT} pin. • Connect a ceramic resonator or a quartz-crystal oscillator between the X_{IN} and X_{OUT} pins to set the oscillation frequency. • If an external clock is used, connect the clock source to the X_{IN} pin and leave the X_{OUT} pin open.	
X_{OUT}	Clock output		
V_{L1} – V_{L3}	LCD power source	• Input $0 \le V_{L1} \le V_{L2} \le V_{L3} \le V_{CC}$ voltage. • Input 0 – V_{L3} voltage to LCD.	
COM_0 – COM_3	Common output	• LCD common output pins. • COM_1, COM_2, and COM_3 are not used at 1/1 duty ratio. • COM_2 and COM_3 are not used at 1/2 duty ratio. • COM_3 is not used at 1/3 duty ratio.	
$P0_0/SEG_8$ – $P0_7/SEG_{15}$	I/O port P0	• 8-bit I/O port. • CMOS compatible input level. • CMOS 3-state output structure. • I/O direction register allows each port to be individually programmed as either input or output. • Pull-down control is enabled.	• LCD segment pins
$P1_0/SEG_{16}$ – $P1_7/SEG_{23}$	I/O port P1		
$P2_0/SEG_0$ – $P2_7/SEG_7$	I/O port P2		
$P3_0/SEG_{24}$ – $P3_7/SEG_{31}$	Output port P3	• 8-bit output port. • CMOS state output. • Pull-down control is enabled.	
$P4_0/S_{CLK2}$	I/O port P4	• 8-bit I/O port. • CMOS compatible input level. • CMOS 3-state output structure. • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• Serial I/O function pin
$P4_1/T_{1OUT}$			• Timer output pin
$P4_2/T_{3OUT}$			• Timer output pin
$P4_3/\phi$			• ϕ output pin
$P4_4/S_{IN}$, $P4_5/S_{OUT}$, $P4_6/\overline{S_{CLK1}}$, $P4_7/\overline{S_{RDY}}$			• Serial I/O function pins

* Mask ROM version of M version is 2.2 V to 5.5 V.

Table 2 Pin description (2)

Pin	Name	Function	Function except a port function
$P5_1$	Input port P5	• 1-bit input pin. • CMOS compatible input level.	
$P5_0$/TA_{OUT}	I/O port P5	• 7-bit I/O port. • CMOS compatible input level. • CMOS 3-state output structure. • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• Timer A output pin
$P5_2$/PWM_1			• PWM_1 output (timer output) pin
$P5_3$/$CNTR_0$, $P5_4$/$CNTR_1$			• External count I/O pins
$P5_5$/INT_0, $P5_6$/INT_1, $P5_7$/INT_2			• External interrupt input pins
$P6_0$/AN_0 – $P6_7$/AN_7	I/O port P6	• 8-bit I/O port. • CMOS compatible input level. • CMOS 3-state output structure. • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• A-D conversion input pins
$P7_0$/X_{CIN}, $P7_1$/X_{COUT}	I/O port P7	• 2-bit I/O port. • CMOS compatible input level. • CMOS 3-state output structure. • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• Sub-clock generating circuit I/O pins
$P8_0$ – $P8_7$	I/O port P8	• 8-bit I/O port. • TTL input level. • CMOS 3-state output structure. • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• Key input (Key-on wake-up) interrupt input pins

PART NUMBERING

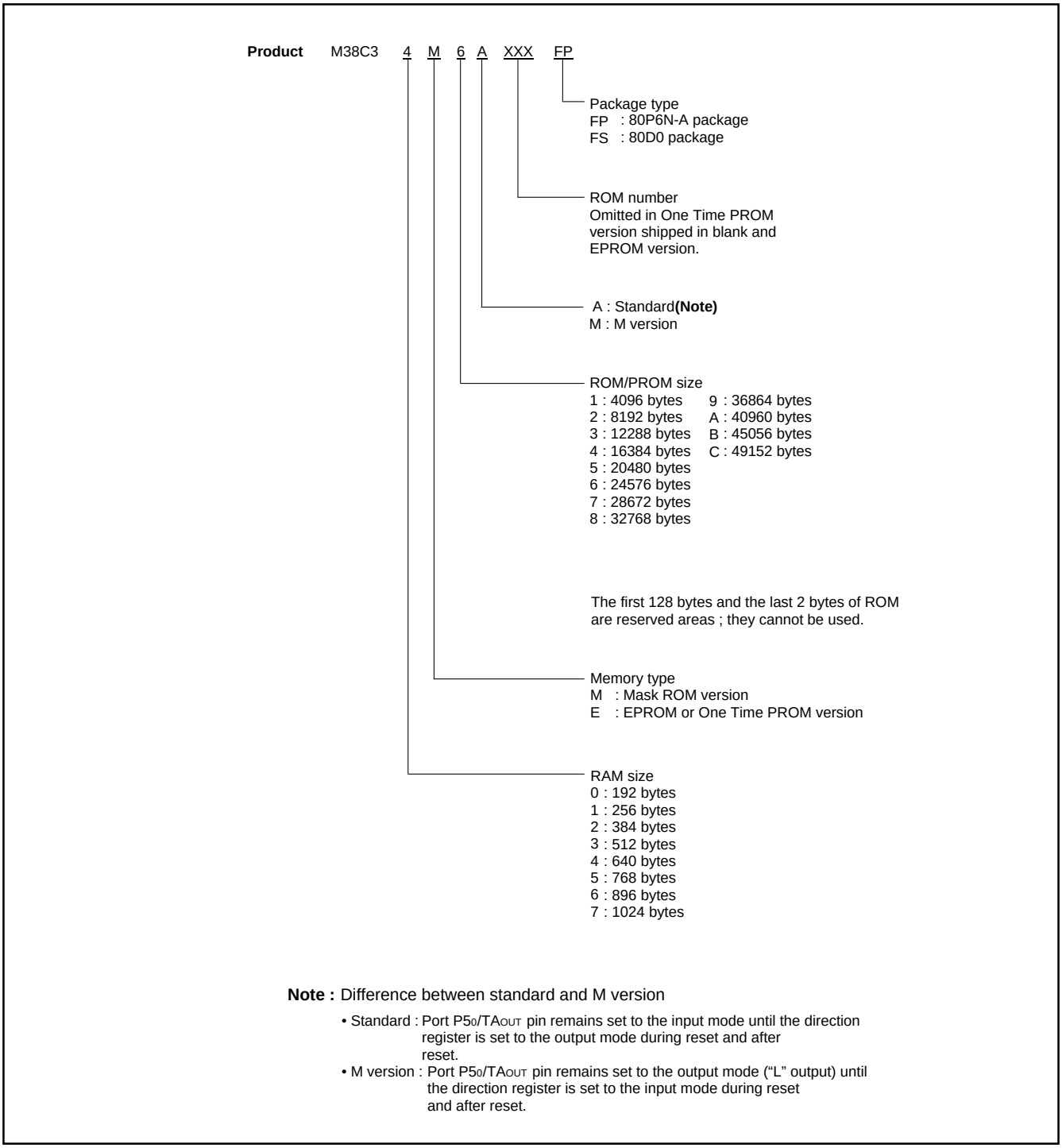


Fig. 3 Part numbering

GROUP EXPANSION

Mitsubishi plans to expand the 38C3 group as follows.

Memory Type

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

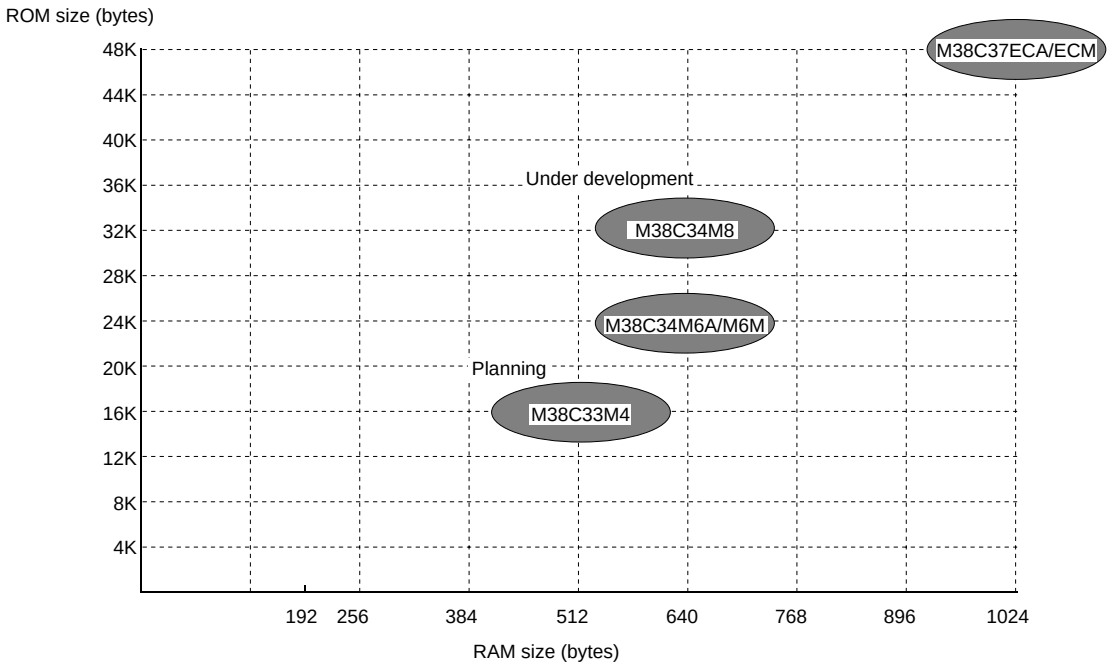
Memory Size

ROM/PROM size ..... 16 K to 48 K bytes
RAM size ..... 512 to 1024 bytes

Packages

80P6N-A ..... 0.8 mm-pitch plastic molded QFP
80D0 ..... 0.8 mm-pitch ceramic LCC (EPROM version)

Memory Expansion Plan



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

Fig. 4 Memory expansion plan

Currently supported products are listed below.

Table 3 Support products

As of December 1998

Product name	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38C34M6AXXXFP	24576 (24446)	640	80P6N-A	Mask ROM version
M38C37ECAXXXFP	49152 (49022)	1024		One Time PROM version
M38C37ECAFP				One Time PROM version (blank)
M38C37ECAFS			80D0	EPROM version
M38C34M6MXXXFP	24576 (24446)	640	80P6N-A	Mask ROM version
M38C37ECMXXXFP	49152 (49022)	1024		One Time PROM version
M38C37ECMFP				One Time PROM version (blank)
M38C37ECMFS			80D0	EPROM version

FUNCTIONAL DESCRIPTION
Central Processing Unit (CPU)

The 38C3 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 6.

Store registers other than those described in Figure 6 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.

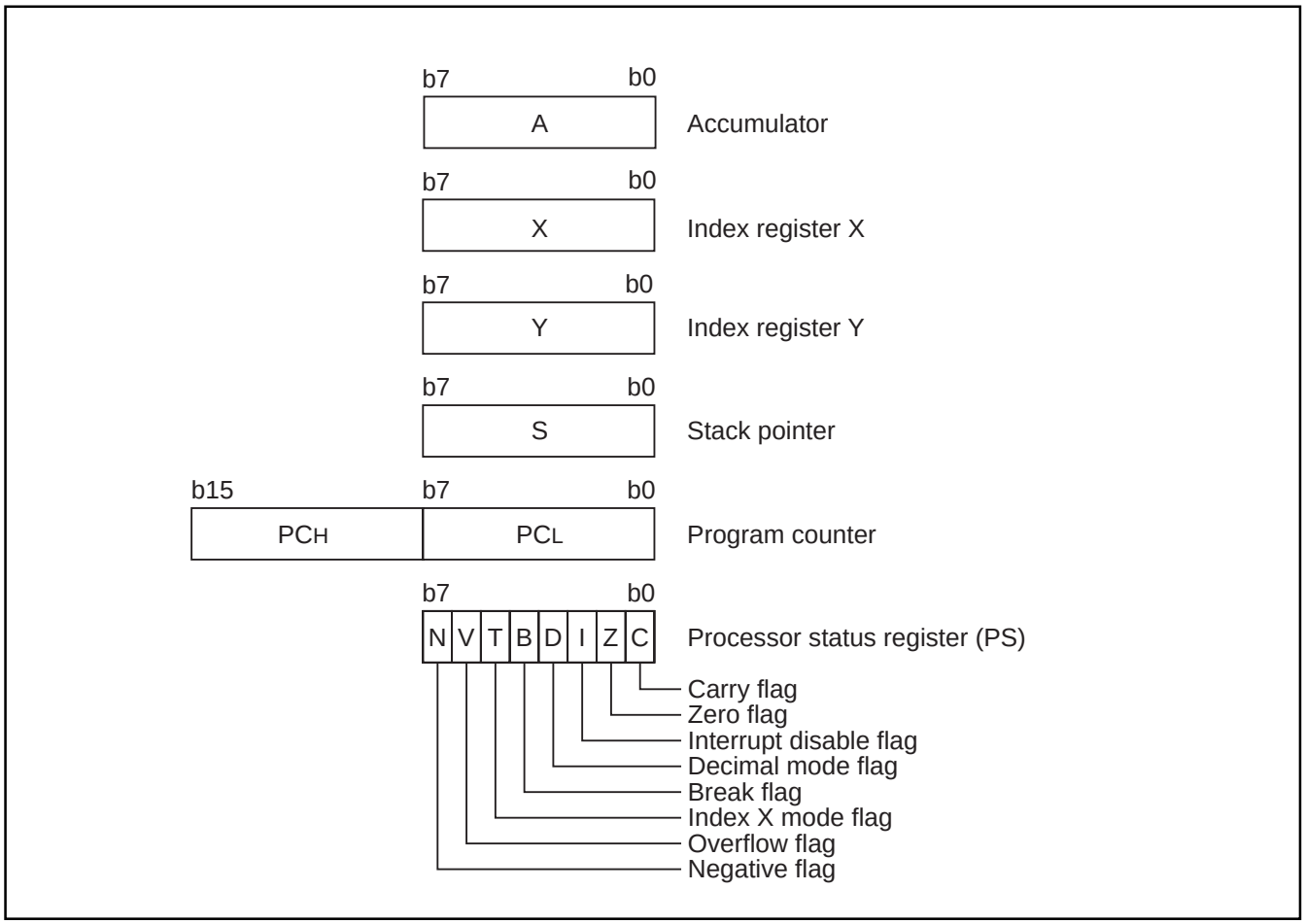


Fig. 5 740 Family CPU register structure

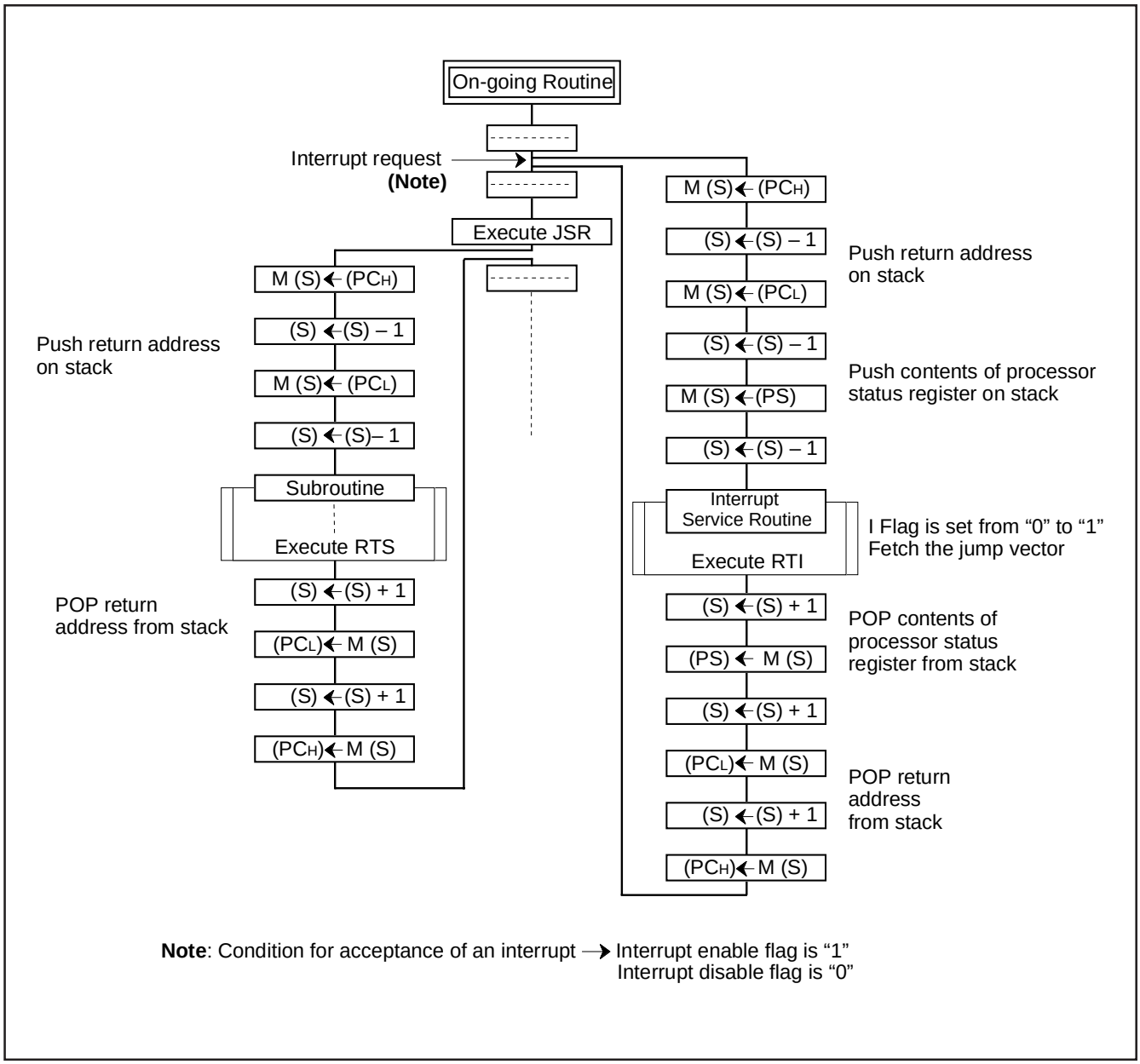


Fig. 6 Register push and pop at interrupt generation and subroutine call

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

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

FUNCTIONAL DESCRIPTION

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

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

[CPU Mode Register (CPUM)] 003B16

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

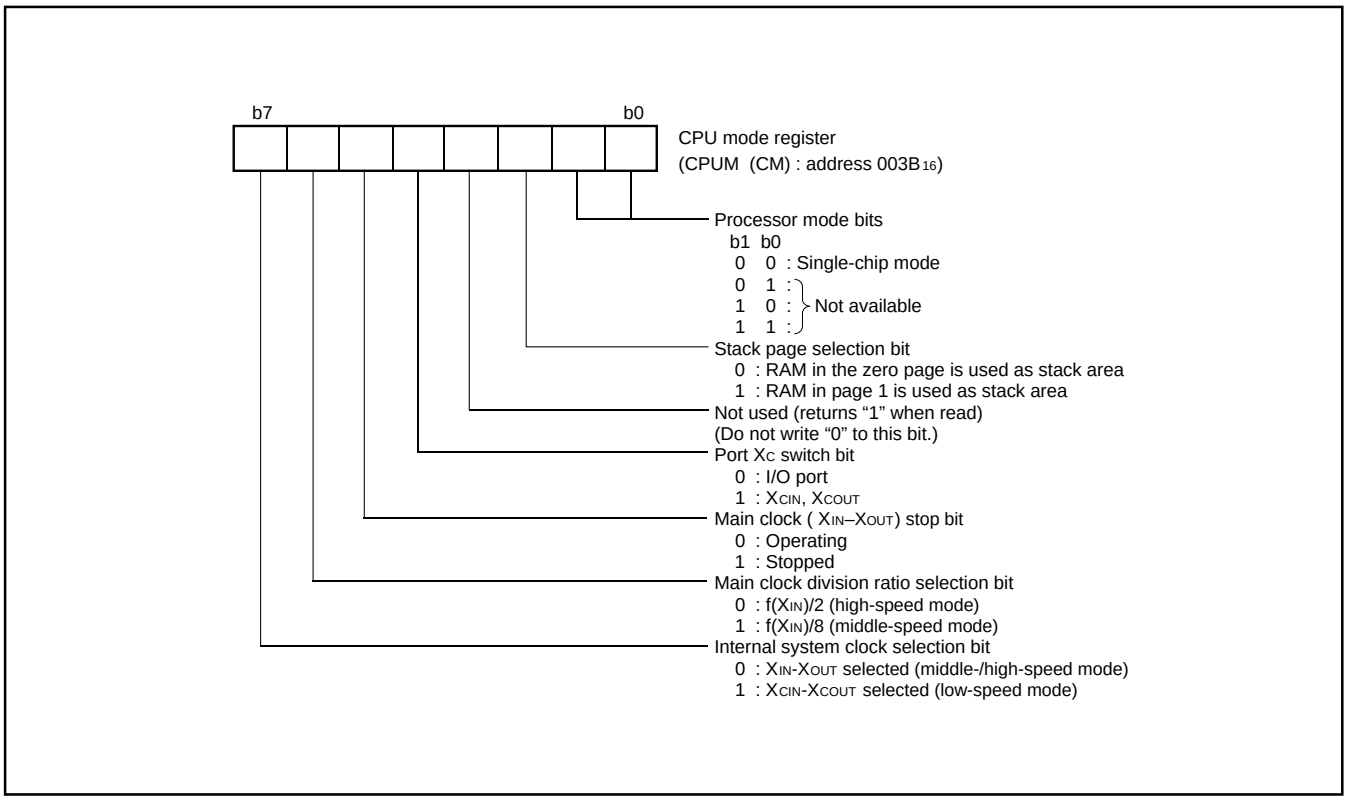


Fig. 7 Structure of CPU mode register

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

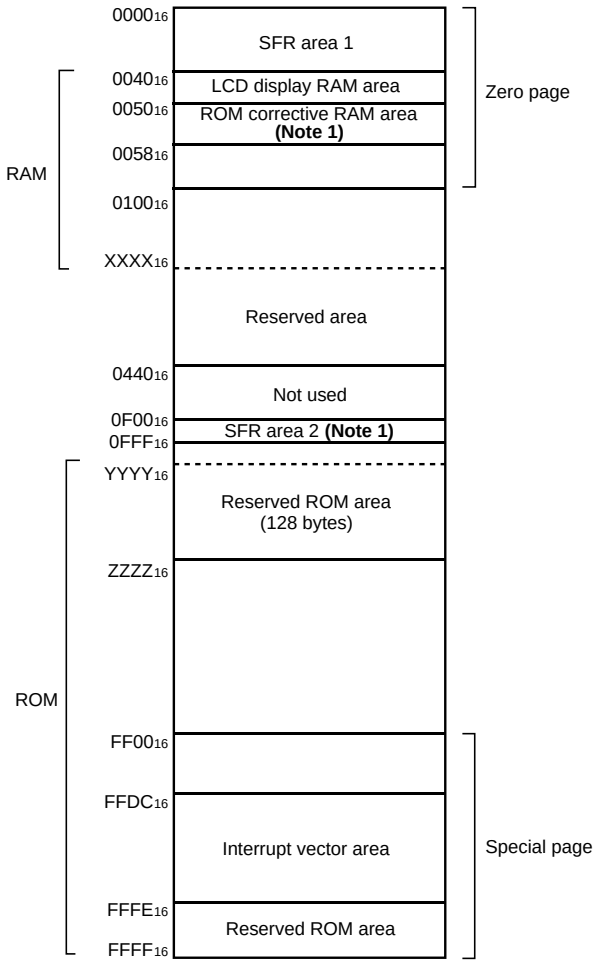
Access to this area with only 2 bytes is possible in the special page addressing mode.

RAM area

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

ROM area

ROM size (bytes)	Address $YYYY_{16}$	Address $ZZZZ_{16}$
4096	$F000_{16}$	$F080_{16}$
8192	$E000_{16}$	$E080_{16}$
12288	$D000_{16}$	$D080_{16}$
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}



Note 1 : This is valid only in mask ROM version.

Fig. 8 Memory map diagram

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}	
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}	
0013_{16}	
0014_{16}	
0015_{16}	
0016_{16}	PULL register A (PULLA)
0017_{16}	PULL register B (PULLB)
0018_{16}	Port P8 output selection register (P8SEL)
0019_{16}	Serial I/O control register 1 (SIOCON1)
$001A_{16}$	Serial I/O control register 2 (SIOCON2)
$001B_{16}$	Serial I/O register (SIO)
$001C_{16}$	
$001D_{16}$	
$001E_{16}$	
$001F_{16}$	

0020_{16}	Timer 1 (T1)
0021_{16}	Timer 2 (T2)
0022_{16}	Timer 3 (T3)
0023_{16}	Timer 4 (T4)
0024_{16}	Timer 5 (T5)
0025_{16}	Timer 6 (T6)
0026_{16}	
0027_{16}	Timer 6 PWM register (T6PWM)
0028_{16}	Timer 12 mode register (T12M)
0029_{16}	Timer 34 mode register (T34M)
$002A_{16}$	Timer 56 mode register (T56M)
$002B_{16}$	ϕ output control register (CKOUT)
$002C_{16}$	Timer A register (low) (TAL)
$002D_{16}$	Timer A register (high) (TAH)
$002E_{16}$	Compare register (low) (CONAL)
$002F_{16}$	Compare register (high) (CONAH)
0030_{16}	Timer A mode register (TAM)
0031_{16}	Timer A control register (TACON)
0032_{16}	A-D control register (ADCON)
0033_{16}	A-D conversion register (low) (ADL)
0034_{16}	A-D conversion register (high) (ADH)
0035_{16}	
0036_{16}	
0037_{16}	
0038_{16}	Segment output enable register (SEG)
0039_{16}	LCD mode register (LM)
$003A_{16}$	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)

$0F01_{16}$	ROM correct enable register 1 **(Note)**
$0F02_{16}$	ROM correct high-order address register 1 **(Note)**
$0F03_{16}$	ROM correct low-order address register 1 **(Note)**
$0F04_{16}$	ROM correct high-order address register 2 **(Note)**
$0F05_{16}$	ROM correct low-order address register 2 **(Note)**
$0F06_{16}$	ROM correct high-order address register 3 **(Note)**
$0F07_{16}$	ROM correct low-order address register 3 **(Note)**
$0F08_{16}$	ROM correct high-order address register 4 **(Note)**
$0F09_{16}$	ROM correct low-order address register 4 **(Note)**

$0F0A_{16}$	ROM correct high-order address register 5 **(Note)**
$0F0B_{16}$	ROM correct low-order address register 5 **(Note)**
$0F0C_{16}$	ROM correct high-order address register 6 **(Note)**
$0F0D_{16}$	ROM correct low-order address register 6 **(Note)**
$0F0E_{16}$	ROM correct high-order address register 7 **(Note)**
$0F0F_{16}$	ROM correct low-order address register 7 **(Note)**
$0F10_{16}$	ROM correct high-order address register 8 **(Note)**
$0F11_{16}$	ROM correct low-order address register 8 **(Note)**

Note: This register is valid only in mask ROM version.

Fig. 9 Memory map of special function register (SFR)

I/O PORTS [Direction Registers (ports P2, P4, P5$_0$, P5$_2$–P5$_7$, and P6–P8)]

The I/O ports P2, P4, P5$_0$, P5$_2$–P5$_7$, and P6–P8 have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, 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 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.

[Direction Registers (ports P0 and P1)]

Ports P0 and P1 have direction registers which determine the input/output direction of each individual port.

Each port in a direction register corresponds to one port, each port can be set to be input or output.

When "0" is written to the bit 0 of a direction register, that port becomes an input port. When "1" is written to that port, that port becomes an output port. Bits 1 to 7 of ports P0 and P1 direction registers are not used.

Pull-up/Pull-down Control

By setting the PULL register A (address 0016$_{16}$) or the PULL register B (address 0017$_{16}$), ports except for ports P3 and P5$_1$ can control either pull-down or pull-up (pins that are shared with the segment output pins for LCD are pull-down; all other pins are pull-up) with a program.

However, the contents of PULL register A and PULL register B do not affect ports programmed as the output ports.

Port P8 Output Selection

Ports P8$_0$ to P8$_7$ can be switched to N-channel open-drain output by setting "1" to the port P8 output selection register.

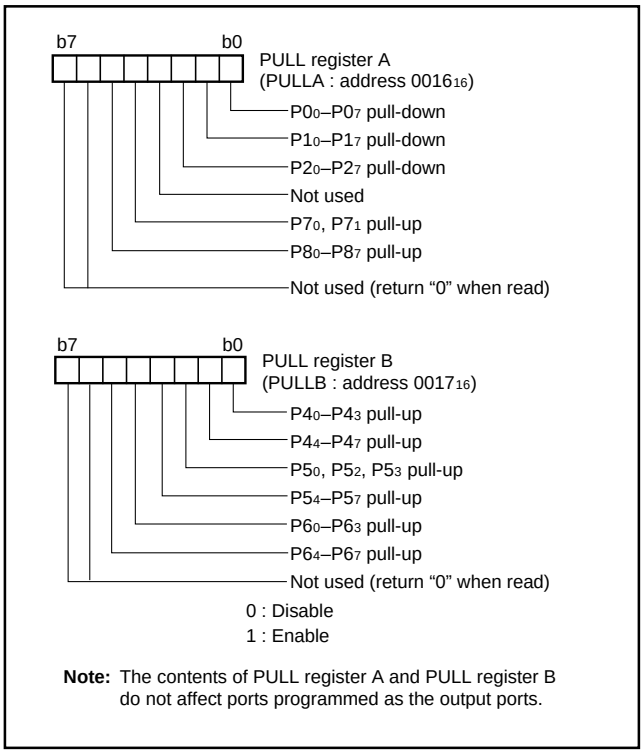


Fig. 10 Structure of PULL register A and PULL register B

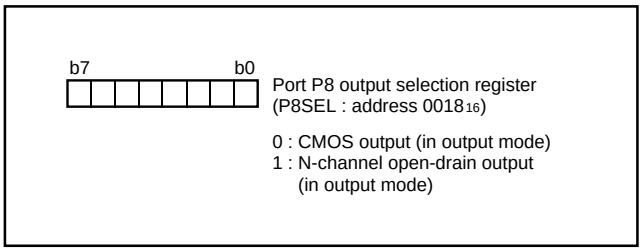


Fig. 11 Structure of port P8 output selection register

Table 6 List of I/O port function (1)

Pin	Name	Input/Output	I/O format	Non-port function	Related SFRs	Ref. No.
P0$_0$/SEG$_8$ – P0$_7$/SEG$_{15}$	Port P0	Input/Output, port unit	CMOS compatible input level CMOS 3-state output	LCD segment output	PULL register A Segment output enable register	(1)
P1$_0$/SEG$_{16}$ – P1$_7$/SEG$_{23}$	Port P1	Input/Output, port unit	CMOS compatible input level CMOS 3-state output	LCD segment output	PULL register A Segment output enable register	
P2$_0$/SEG$_0$ – P2$_7$/SEG$_7$	Port P2	Input/Output, individual bits	CMOS compatible input CMOS 3-state output	LCD segment output	PULL register A Segment output enable register	
P3$_0$/SEG$_{24}$ – P3$_7$/SEG$_{31}$	Port P3	Output, individual bits	CMOS 3-state output	LCD segment output	Segment output enable register	(2)

Table 7 List of I/O port function (2)

Pin	Name	Input/Output	I/O format	Non-port function	Related SFRs	Ref. No.
P40/SCLK2	Port P4	Input/Output, individual bits	CMOS compatible input level CMOS 3-state output	Serial I/O function I/O	Serial I/O control registers 1, 2 PULL register B	(3)
P41/T1OUT				Timer output	Timer 12 mode register PULL register B	(4)
P42/T3OUT				Timer output	Timer 34 mode register PULL register B	(4)
P43/φ				φ clock output	φ output control register PULL register B	(5)
P44/SIN				Serial I/O function I/O	Serial I/O control registers 1, 2 PULL register B	(6)
P45/SOUT						(7)
P46/SCLK1						(8)
P47/SRDY						(9)
P50/TAOUT	Port P5	Input/Output, individual bits	CMOS compatible input level CMOS 3-state output	Timer A output	Timer A mode register Timer A control register PULL register B	(10)
P51		Input	CMOS compatible input level			(11)
P52/PWM1		Input/Output, individual bits	CMOS compatible input level CMOS 3-state output	PWM output	Timer 56 mode register PULL register B	(4)
P53/CNTR0 P54/CNTR1				External count I/O	Interrupt edge selection register PULL register B	(12)
P55/INT0 P56/INT1 P57/INT2				External interrupt input	Interrupt edge selection register PULL register B	(12)
P60/AN0 – P67/AN7	Port P6	Input/Output, individual bits	CMOS compatible input level CMOS 3-state output	A-D converter input	A-D control register PULL register B	(13)
P70/XCIN	Port P7	Input/Output, individual bits	CMOS compatible input level CMOS 3-state output	Sub-clock generating circuit I/O	CPU mode register PULL register A	(14)
P71/XCOUT						(15)
P80 – P87	Port P8	Input/Output, individual bits	CMOS compatible input level CMOS 3-state output	Key input (key-on wake-up) interrupt input	Interrupt control register 2 PULL register A	(17)
COM0 – COM3	Common	Output	LCD common output		LCD mode register	(16)

Notes 1: 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.
2: For details of how to use double function ports as function I/O ports, refer to the applicable sections.

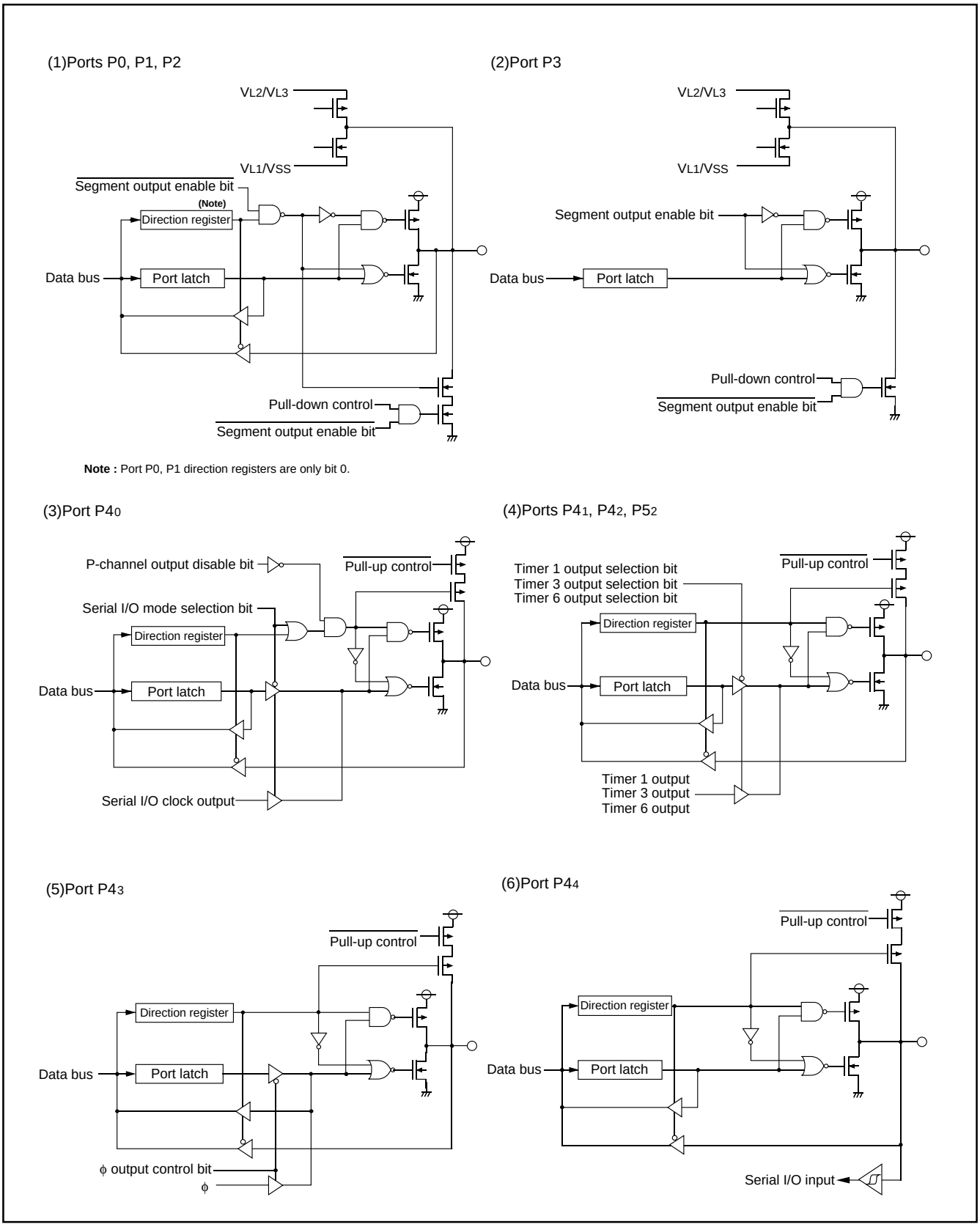
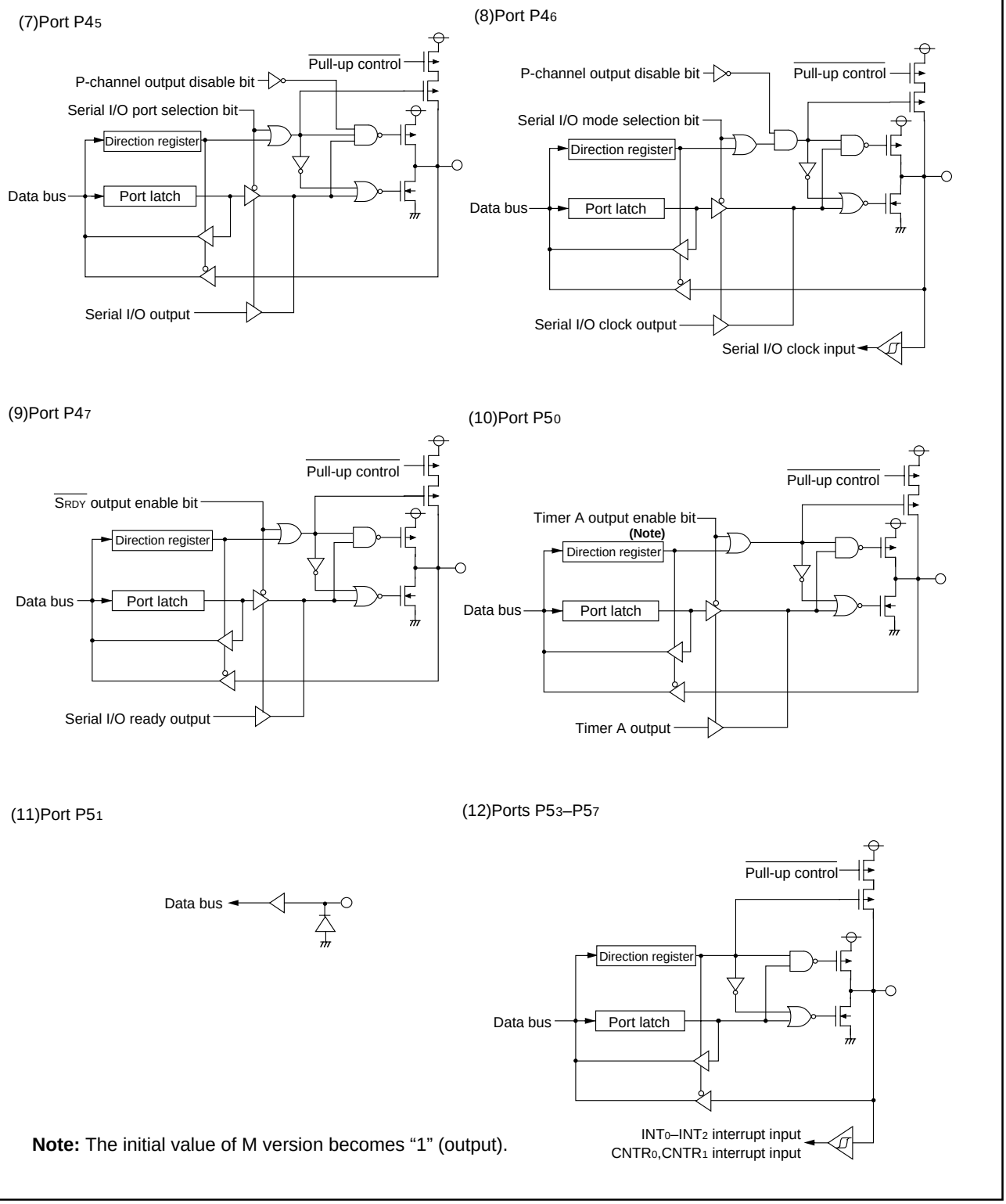


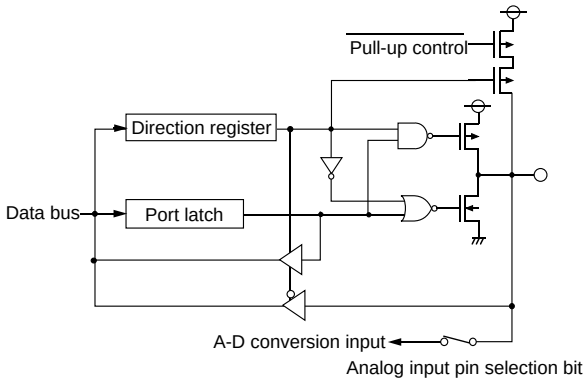
Fig. 12 Port block diagram (1)



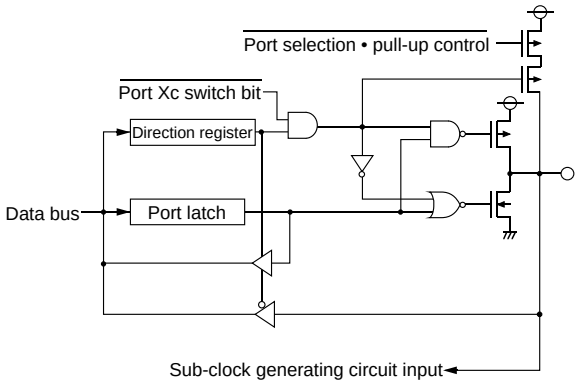
Note: The initial value of M version becomes “1” (output).

Fig. 13 Port block diagram (2)

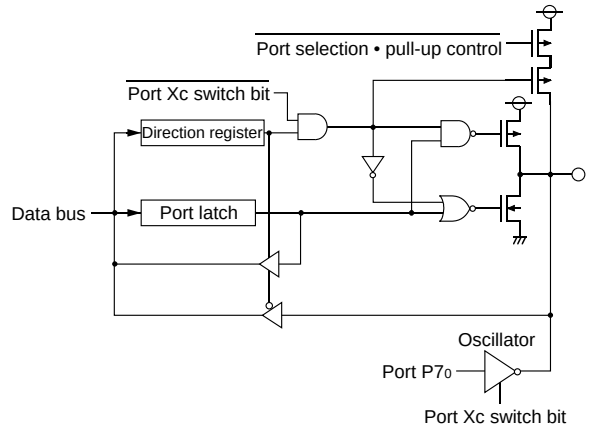
(13)Port P6



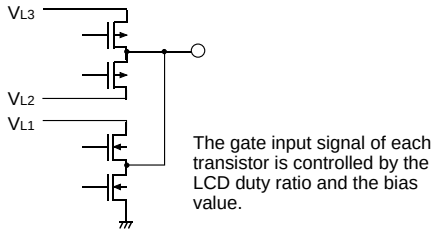
(14)Port P7$_0$



(15)Port P7$_1$



(16)COM$_0$–COM$_3$



(17)Port P8

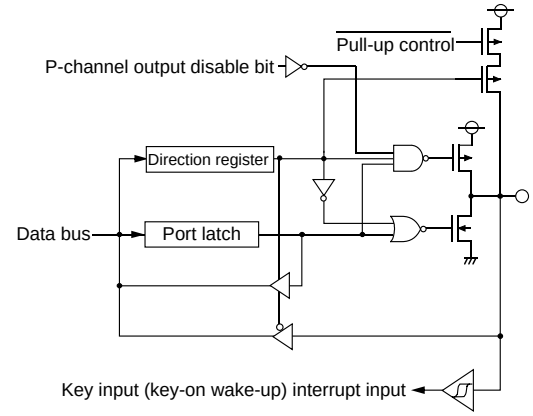


Fig. 14 Port block diagram (3)

INTERRUPTS

Interrupts occur by sixteen sources: six external, nine internal, and one software.

Interrupt Control

Each interrupt except the BRK instruction interrupt have both an interrupt request bit and an interrupt enable bit, and is controlled by the interrupt disable flag. 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 interrupt and reset cannot be disabled with any flag or bit. The I flag disables all interrupts except the BRK instruction interrupt and reset. If several interrupts requests occurs at the same time the interrupt with highest priority is accepted first.

Interrupt Operation

By acceptance of an interrupt, the following operations are automatically performed:

1. The processing being executed is stopped.
2. The contents of the program counter and processor status register are automatically pushed onto the stack.
3. The interrupt disable flag is set and the corresponding interrupt request bit is cleared.
4. The interrupt jump destination address is read from the vector table into the program counter.

■Notes on Interrupts

When the active edge of an external interrupt (INT_0 – INT_2, $CNTR_0$ or $CNTR_1$) is set or an vector interrupt source where several interrupt source is assigned to the same vector address is switched, the corresponding interrupt request bit may also be set. Therefore, take following sequence:

(1) Disable the interrupt.
(2) Change the active edge in interrupt edge selection register.
(3) Clear the set interrupt request bit to “0.”
(4) Enable the interrupt.

Table 8 Interrupt vector addresses and priority

Interrupt Source	Priority	Vector Addresses **(Note 1)**		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset **(Note 2)**	1	FFFD16	FFFC16	At reset	Non-maskable
INT0	2	FFFB16	FFFA16	At detection of either rising or falling edge of INT0 input	External interrupt (active edge selectable)
INT1	3	FFF916	FFF816	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
INT2	4	FFF716	FFF616	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
Serial I/O	5	FFF516	FFF416	At completion of serial I/O data transmit/receive	Valid when serial I/O is selected
Timer A	6	FFF316	FFF216	At timer A underflow	
Timer 1	7	FFF116	FFF016	At timer 1 underflow	
Timer 2	8	FFEF16	FFEE16	At timer 2 underflow	STP release timer underflow
Timer 3	9	FFED16	FFEC16	At timer 3 underflow	
Timer 4	10	FFEB16	FFEA16	At timer 4 underflow	
Timer 5	11	FFE916	FFE816	At timer 5 underflow	
Timer 6	12	FFE716	FFE616	At timer 6 underflow	
CNTR0	13	FFE516	FFE416	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
CNTR1	14	FFE316	FFE216	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
Key input (Key-on wake-up)	15	FFE116	FFE016	At falling of port P8 (at input) input logical level AND	External interrupt (falling valid)
A-D conversion	16	FFDF16	FFDE16	At completion of A-D conversion	Valid when A-D conversion interrupt is selected
BRK instruction	17	FFDD16	FFDC16	At BRK instruction execution	Non-maskable software interrupt

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

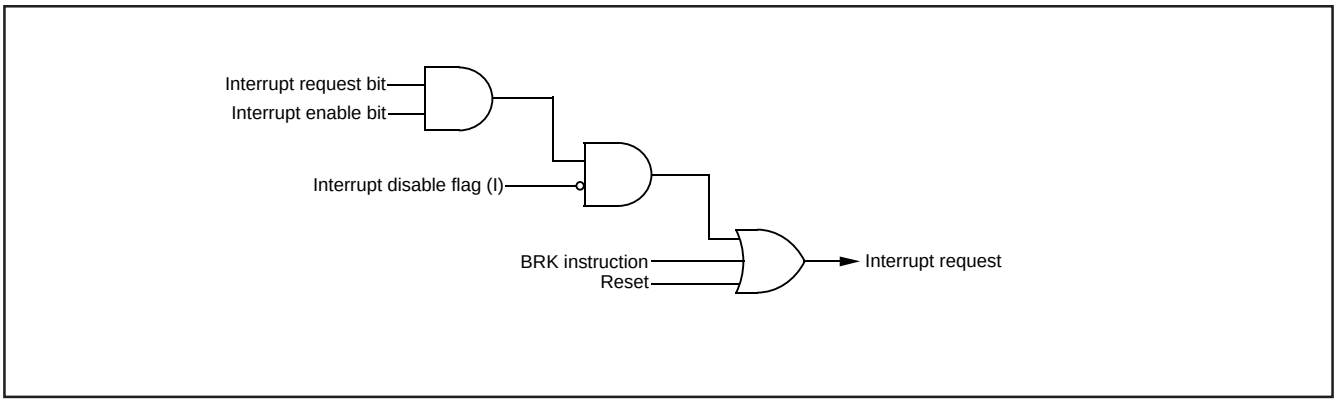


Fig. 15 Interrupt control

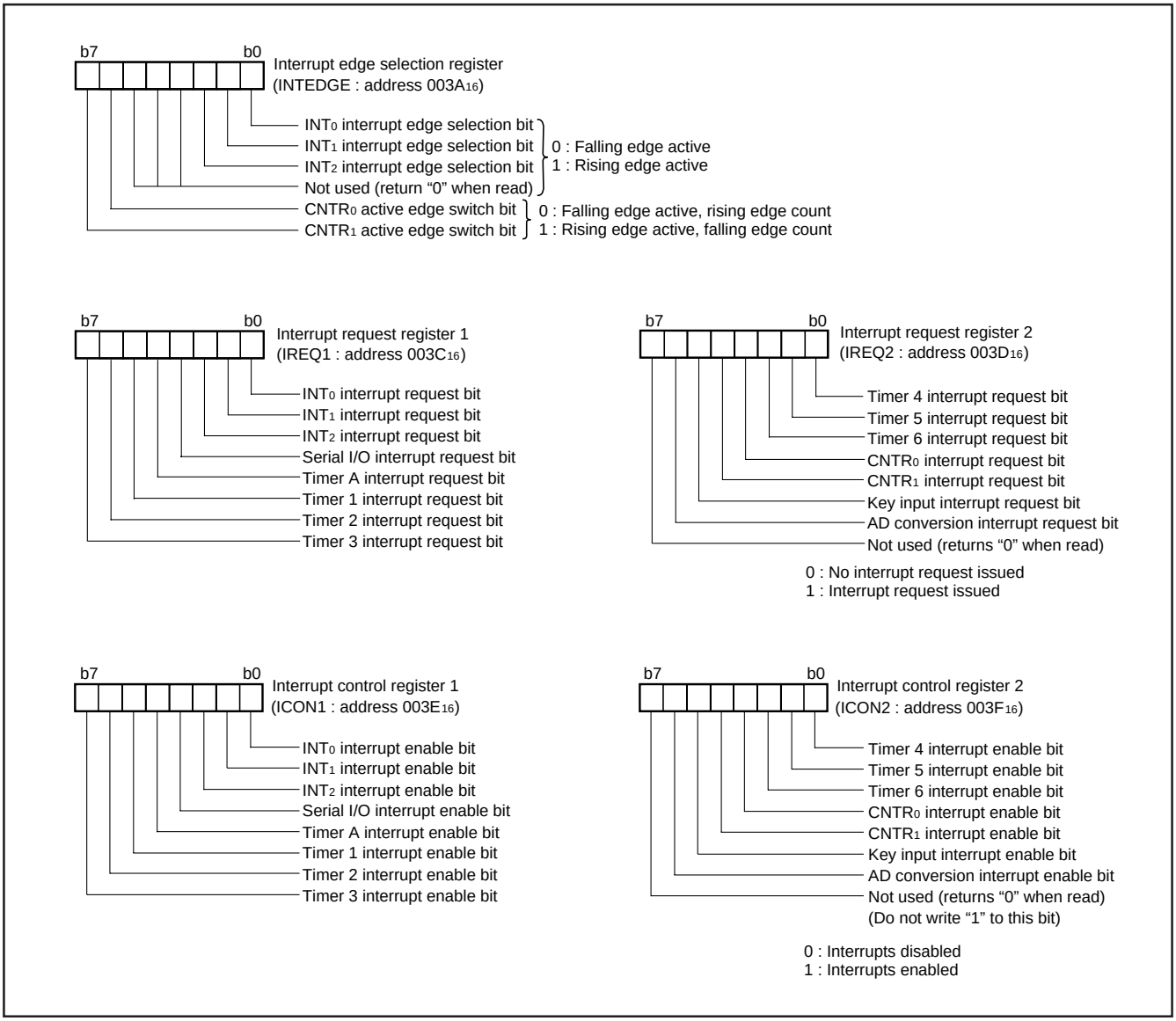


Fig. 16 Structure of interrupt-related registers

Key Input Interrupt (Key-on Wake-Up)

A key input interrupt request is generated by applying "L" level to any pin of port P8 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 17, where an interrupt request is generated by pressing one of the keys consisted as an active-low key matrix which inputs to ports P80–P83.

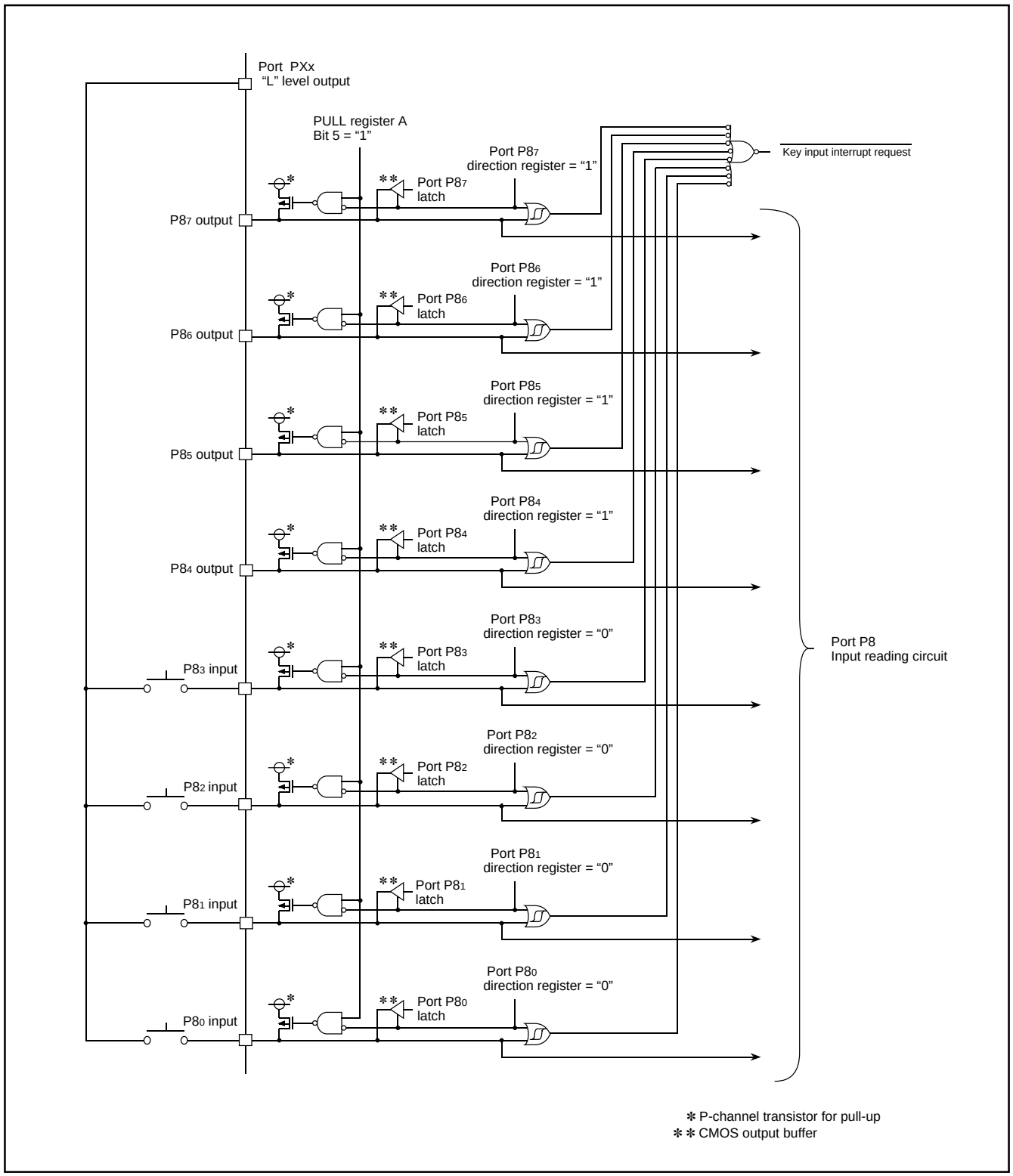


Fig. 17 Connection example when using key input interrupt and port P8 block diagram

TIMERS
8-Bit Timer

The 38C3 group has six built-in timers : Timer 1, Timer 2, Timer 3, Timer 4, Timer 5, and Timer 6.

Each timer has the 8-bit timer latch. All timers are down-counters. When the timer reaches "00$_{16}$," an underflow occurs with the next count pulse. Then the contents of the timer latch is reloaded into the timer and the timer continues down-counting. When a timer underflows, the interrupt request bit corresponding to that timer is set to "1."

The count can be stopped by setting the stop bit of each timer to "1." The system clock ϕ can be set to either the high-speed mode or low-speed mode with the CPU mode register. At the same time, timer internal count source is switched to either f(XIN) or f(XCIN).

●Timer 1, Timer 2

The count sources of timer 1 and timer 2 can be selected by setting the timer 12 mode register. A rectangular waveform of timer 1 underflow signal divided by 2 is output from the P4$_1$/T1OUT pin. The waveform polarity changes each time timer 1 overflows. The active edge of the external clock CNTR$_0$ can be switched with the bit 6 of the interrupt edge selection register.

At reset or when executing the STP instruction, all bits of the timer 12 mode register are cleared to "0," timer 1 is set to "FF$_{16}$," and timer 2 is set to "01$_{16}$."

●Timer 3, Timer 4

The count sources of timer 3 and timer 4 can be selected by setting the timer 34 mode register. A rectangular waveform of timer 3 underflow signal divided by 2 is output from the P4$_2$/T3OUT pin. The waveform polarity changes each time timer 3 overflows. The active edge of the external clock CNTR$_1$ can be switched with the bit 7 of the interrupt edge selection register.

●Timer 5, Timer 6

The count sources of timer 5 and timer 6 can be selected by setting the timer 56 mode register. A rectangular waveform of timer 6 underflow signal divided by 2 can be output from the P5$_2$/PWM$_1$ pin.

●Timer 6 PWM$_1$ Mode

Timer 6 can output a rectangular waveform with "H" duty cycle n/(n+m) from the P5$_2$/PWM$_1$ pin by setting the timer 56 mode register (refer to Figure 20). The n is the value set in timer 6 latch (address 0025$_{16}$) and m is the value in the timer 6 PWM register (address 0027$_{16}$). If n is "0," the PWM output is "L," if m is "0," the PWM output is "H" (n = 0 is prior than m = 0). In the PWM mode, interrupts occur at the rising edge of the PWM output.

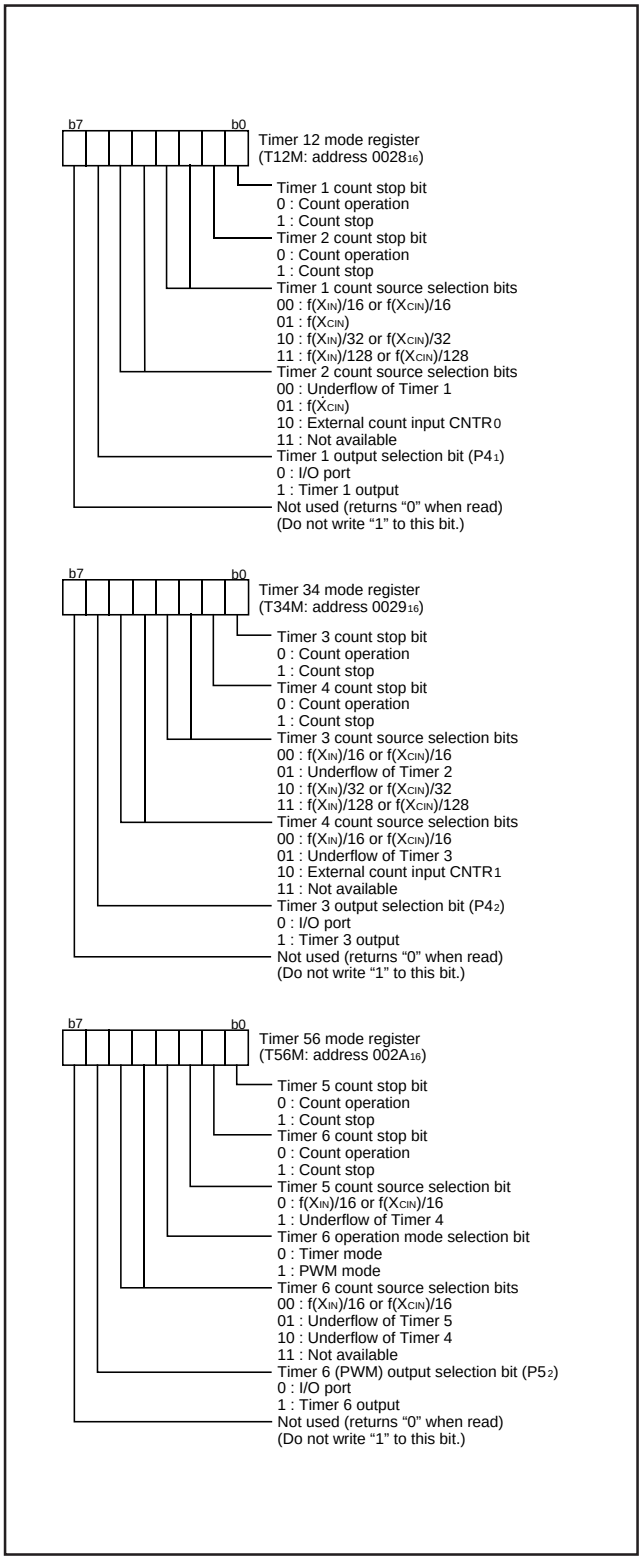


Fig. 18 Structure of timer related register

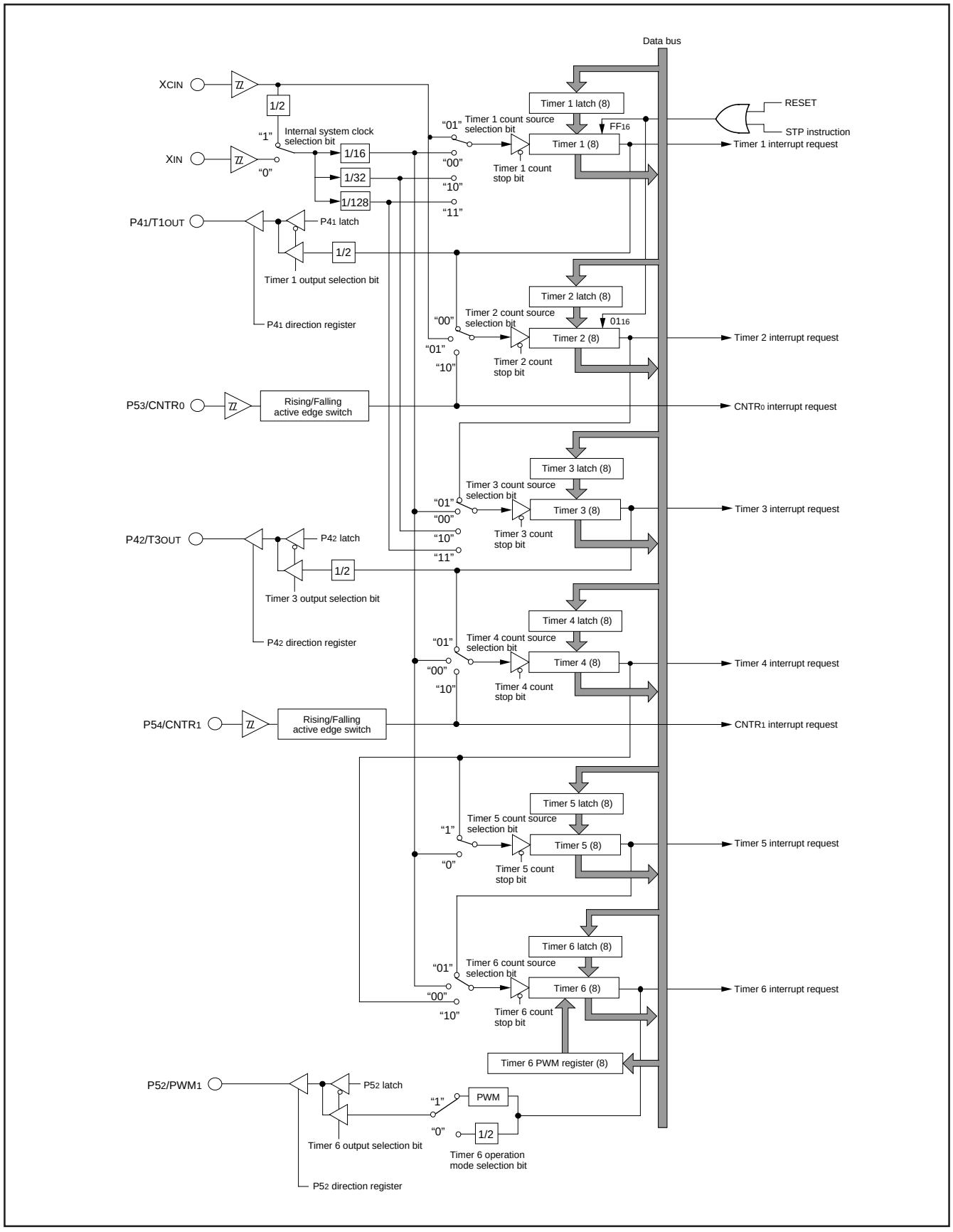
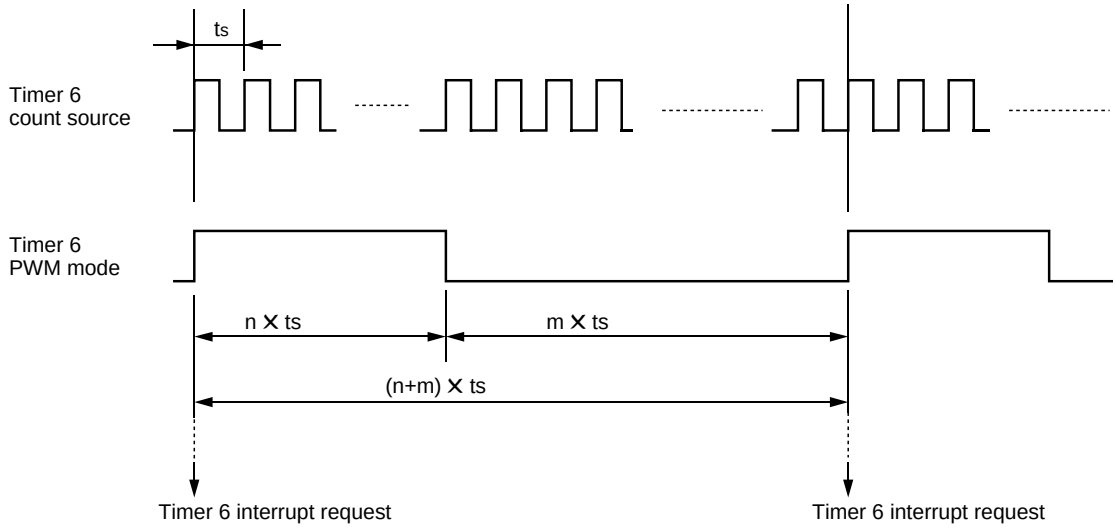


Fig. 19 Block diagram of timer



Note: PWM waveform (duty : n/(n+m) and period : (n+m) × ts) is output.
n: setting value of Timer 6
m: setting value of Timer 6 PWM register
ts: period of Timer 6 count source

Fig. 20 Timing chart of timer 6 PWM1 mode

16-bit Timer

Timer A is a 16-bit timer that can be selected in one of four modes by the timer A mode register and the timer A control register.

●Timer A

The timer A operates as down-count. When the timer contents reach "0000_{16}", an underflow occurs at the next count pulse and the timer latch contents are reloaded. After that, the timer continues count-down. When the timer underflows, the interrupt request bit corresponding to the timer A is set to "1".

(1) Timer mode

The count source can be selected by setting the timer A mode register.

(2) Pulse output mode

Pulses of which polarity is inverted each time the timer underflows are output from the TA_{OUT} pin. Except for that, this mode operates just as in the timer mode.

When using this mode, set port $P5_0$ sharing the TA_{OUT} pin to output mode.

(3) IGBT output mode

After dummy output from the TA_{OUT} pin, count starts with the INT_0 pin input as a trigger. When the trigger is detected or the timer A underflows, "H" is output from the the TA_{OUT} pin.

When the count value corresponds with the compare register value, the TA_{OUT} output becomes "L". When the INT_0 signal becomes "H", the TA_{OUT} output is forced to become "L".

After noise is cleared by noise filters, judging continuous 4-time same levels with sampling clocks to be signals, the INT_0 signal can use 4 types of delay time by a delay circuit.

When using this mode, set port $P5_5$ sharing the INT_0 pin to input mode and set port $P5_0$ sharing the TA_{OUT} pin to output mode.

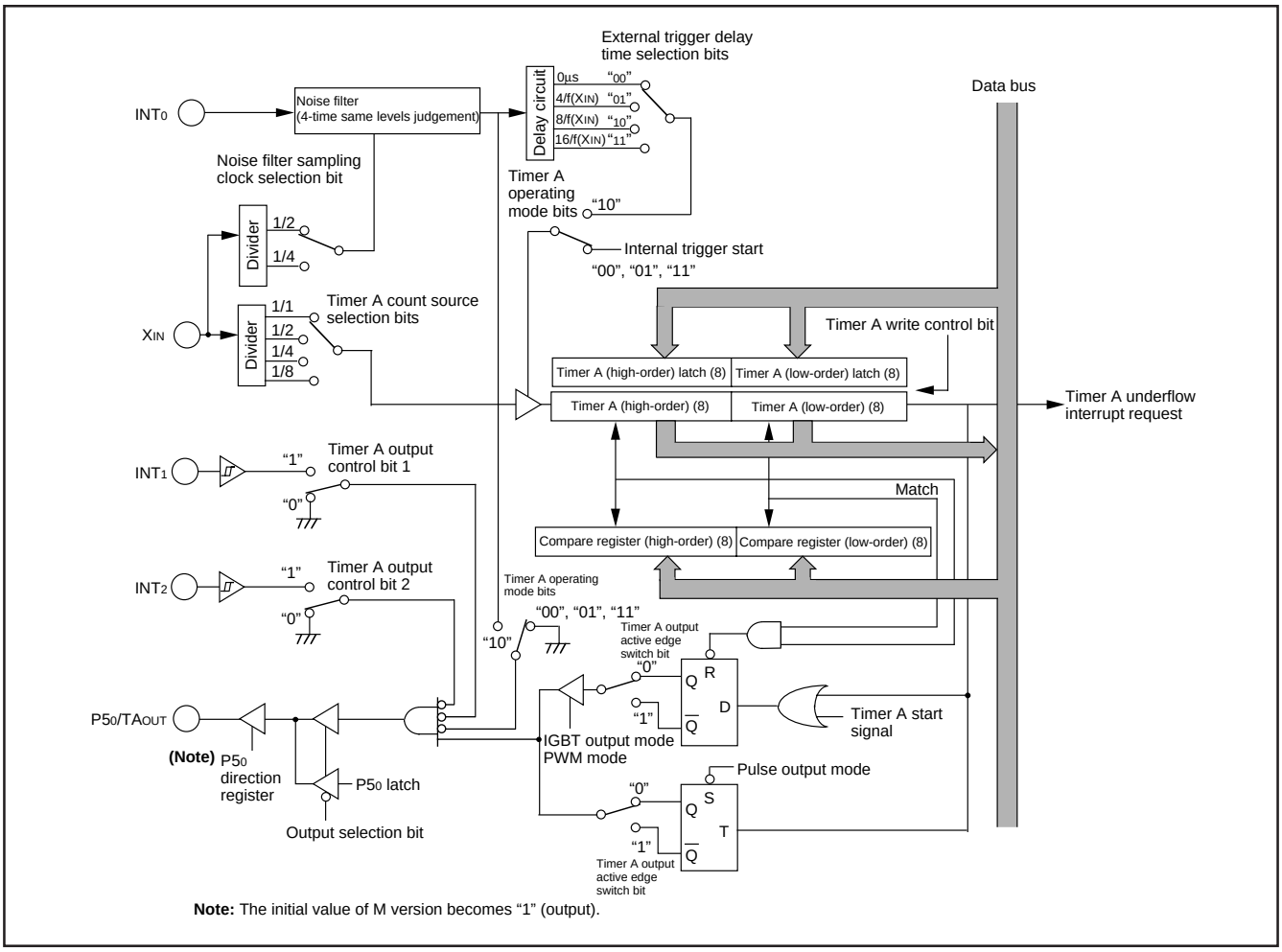
It is possible to force the timer A output to be "L" using pins INT_1 and INT_2 by the timer A control register.

(4) PWM mode

IGBT dummy output, an external trigger with the INT_0 pin and output control with pins INT_1 and INT_2 are not used. Except for those, this mode operates just as in the IGBT output mode.

The period of PWM waveform is specified by the timer A set value. The "H" term is specified by the compare register set value.

When using this mode, set port $P5_0$ sharing the TA_{OUT} pin to output mode.



Note: The initial value of M version becomes "1" (output).

Fig. 21 Block diagram of timer A

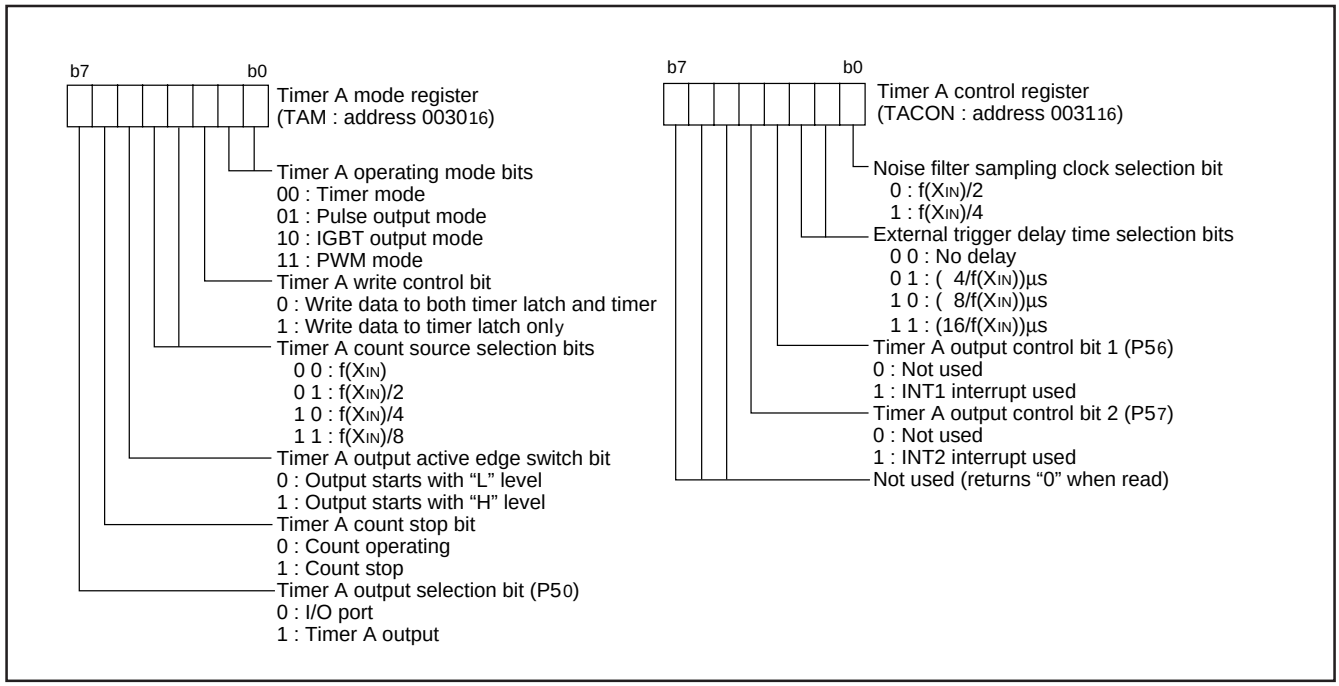


Fig. 22 Structure of timer A related registers

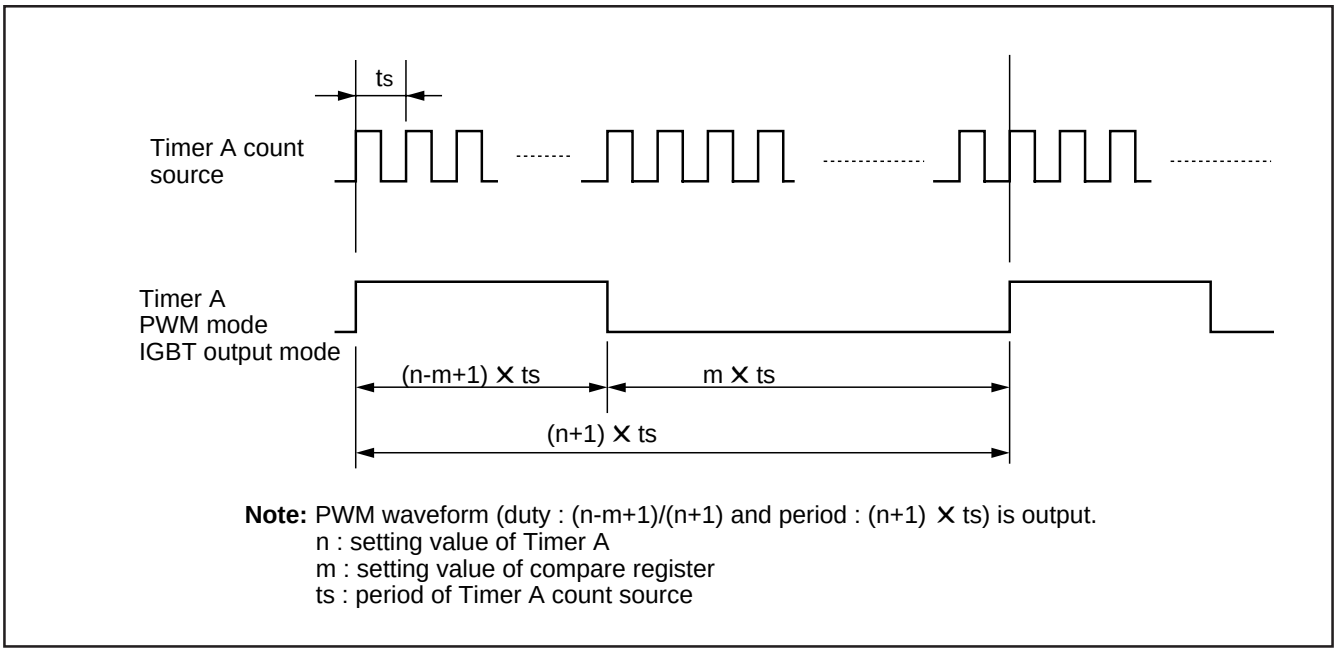


Fig. 23 Timing chart of timer A PWM, IGBT output modes

■Notes on Timer A

(1) Write order to timer A

- In the timer and pulse output modes, write to the timer A register (low-order) first and to the timer A register (high-order) next. Do not write to only one side.
- In the IGBT and PWM modes, write to the registers as follows:
  the compare register (high- and low-order)
  the timer A register (low-order)
  the timer A register (high-order).
  It is possible to use whichever order to write to the compare register (high- and low-order). However, write both the compare register and the timer A register at the same time.

(2) Read order to timer A

- In all modes, read to the timer A register (high-order) first and to the timer A register (low-order) next. Read order to the compare register is not specified.
- If reading to the timer A register during write operation or writing to it during read operation, normal operation will not be performed.

(3) Write to timer A

- When writing a value to the timer A address to write to the latch only, the value is set into the reload latch and the timer is updated at the next underflow. Normally, when writing a value to the timer A address, the value is set into the timer and the timer latch at the same time, because they are written at the same time.
  When writing to the latch only, if the write timing to the high-order reload latch and the underflow timing are almost the same, an expected value may be set in the high-order counter.
- Do not switch the timer count source during timer count operation. Stop the timer count before switching it. Additionally, when performing write to the latch and the timer at the same time, the timer count value may change large.

(4) Set of timer A mode register

Set the write control bit to "1" (write to the latch only) when setting the IGBT and PWM modes.
Output waveform simultaneously reflects the contents of both registers at the next underflow after writing to the timer A register (high-order).

(5) Output control function of timer A

When using the output control function (INT_1 and INT_2) in the IGBT mode, set the levels of INT_1 and INT_2 to "H" in the falling edge active or to "L" in the rising edge active before switching to the IGBT mode.

SERIAL I/O

The 38C3 group has a built-in 8-bit clock synchronous serial I/O. The I/O pins of serial I/O also operate as I/O port P4, and their function is selected by the serial I/O control register 1 (address 0019$_{16}$).

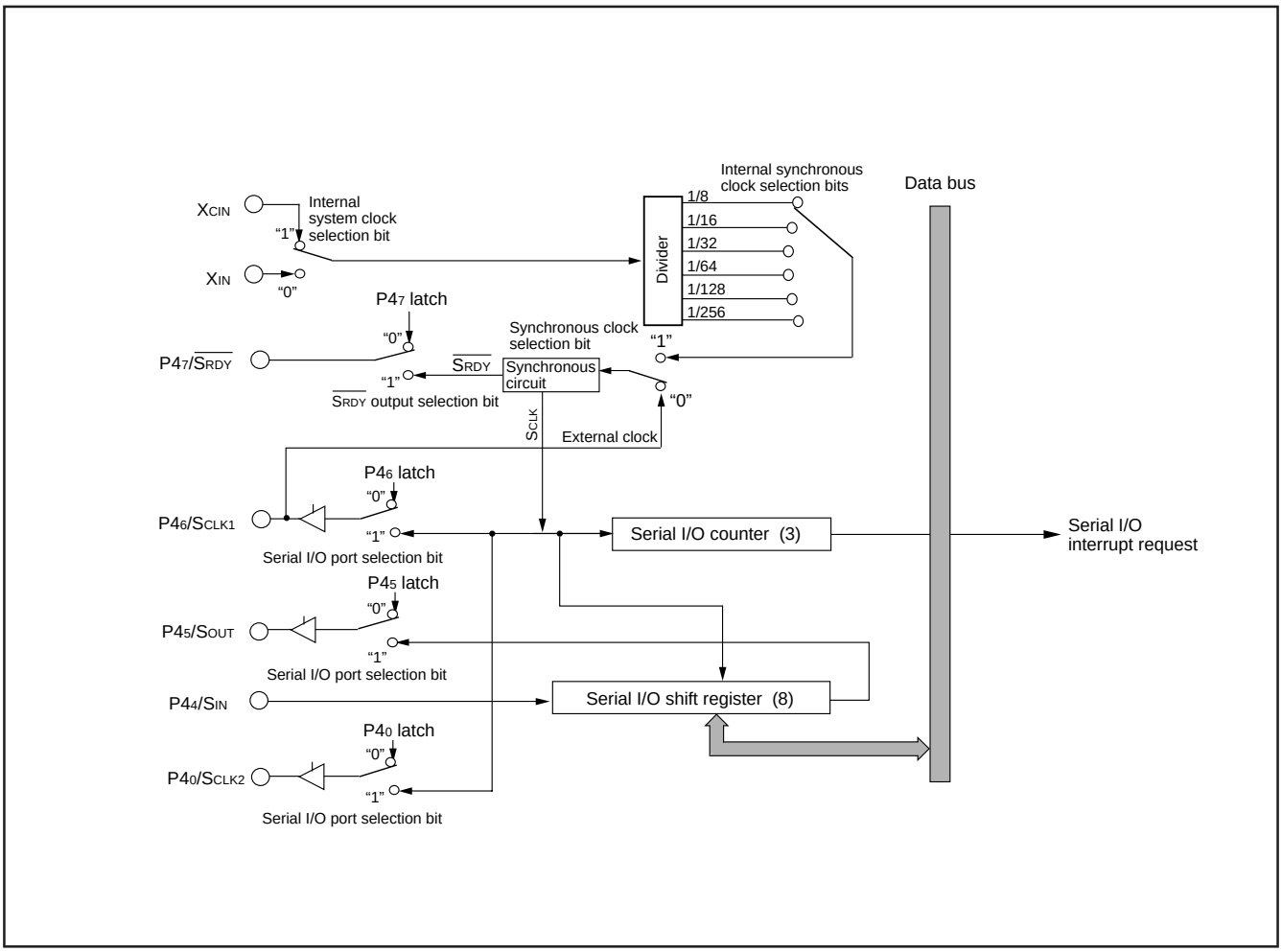


Fig. 24 Block diagram of serial I/O

[Serial I/O Control Registers 1, 2 (SIOCON1, SIOCON2)] 0019$_{16}$, 001A$_{16}$

Each of the serial I/O control registers 1, 2 contains 8 bits that select various control parameters of serial I/O.

●Operation in serial I/O mode

Either an internal clock or an external clock can be selected as the synchronous clock for serial I/O transfer. A dedicated divider is built-in as the internal clock, giving a choice of six clocks.

When internal clock is selected, serial I/O starts to transfer by a write signal to the serial I/O register (address 001B$_{16}$). After 8 bits have been transferred, the SOUT pin goes to high impedance.

When external clock is selected, the clock must be controlled externally because the contents of the serial I/O register continue to shift while the transfer clock is input. In this case, the SOUT pin does not go to high impedance at the completion of data transfer.

The interrupt request bit is set at the end of the transfer of 8 bits, regardless of whether the internal or external clock is selected.

When selecting internal clock and setting "1" to SIOCON2$_0$, the P4$_0$ pin can be also used as synchronous clock output pin SCLK$_2$. At this time, the SCLK$_1$ pin can be used as I/O port.

Table 9 Function of P4$_6$/SCLK$_1$ and P4$_0$/SCLK$_2$

SIOCON1$_6$	SIOCON1$_3$	SIOCON2$_0$	P4$_6$/SCLK$_1$	P4$_0$/SCLK$_2$
1	1	0	SCLK$_1$	P4$_0$
		1	P4$_6$	SCLK$_2$

SIOCON1$_3$: Serial I/O port selection bit
SIOCON1$_6$: Synchronous clock selection bit
SIOCON2$_0$: Synchronous clock output pin selection bit

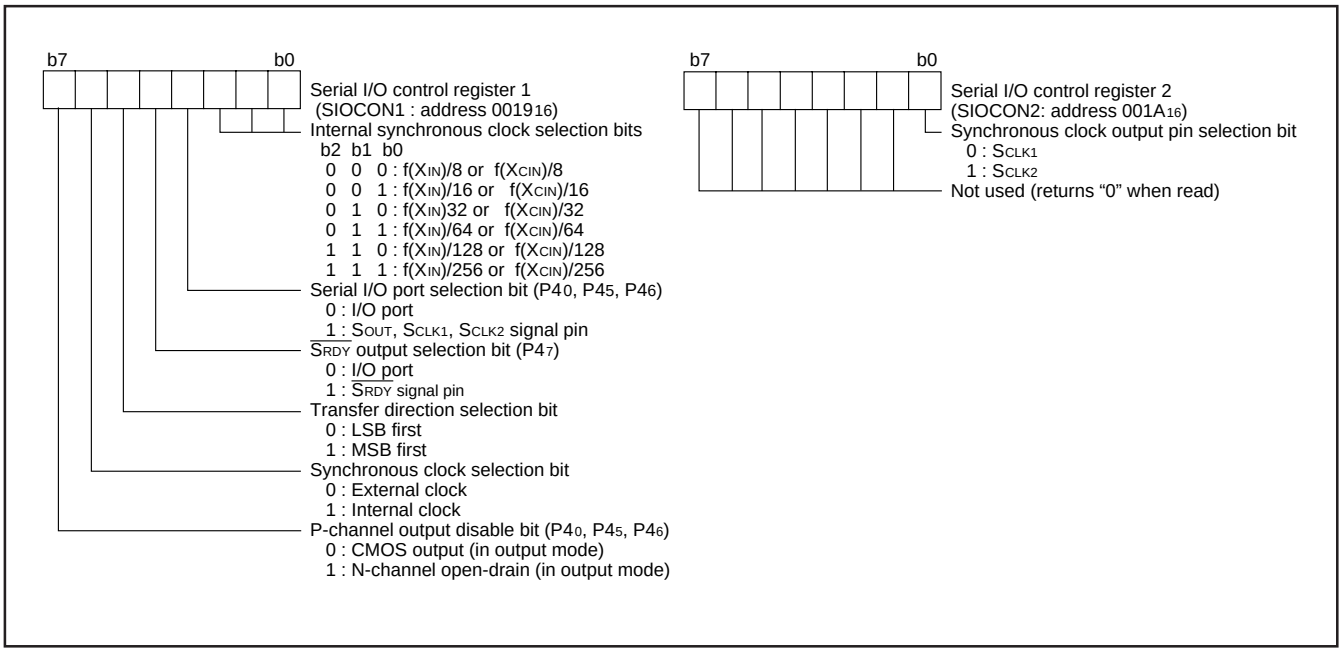


Fig. 25 Structure of serial I/O control register

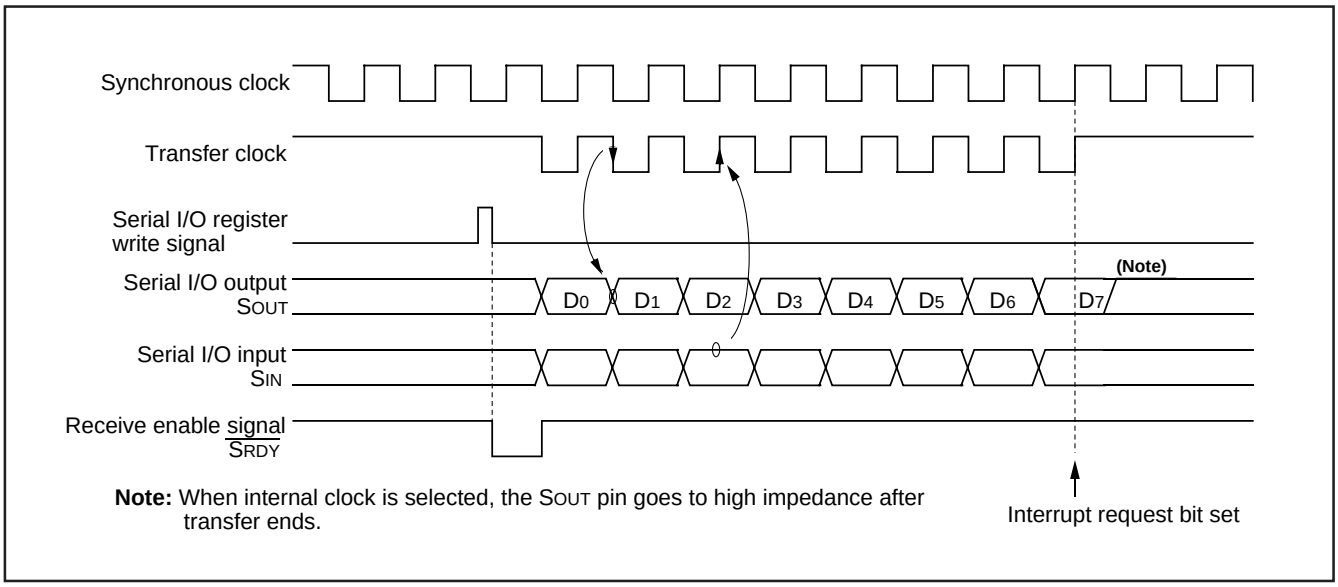


Fig. 26 Serial I/O timing (for LSB first)

A-D CONVERTER

The 38C3 group has a 10-bit A-D converter. The A-D converter performs successive approximation conversion.

[A-D Conversion Register (AD)] 0033$_{16}$, 0034$_{16}$

One of these registers is a high-order register, and the other is a low-order register. The high-order 8 bits of a conversion result is stored in the A-D conversion register (high-order) (address 0034$_{16}$), and the low-order 2 bits of the same result are stored in bit 7 and bit 6 of the A-D conversion register (low-order) (address 0033$_{16}$).
During A-D conversion, do not read these registers.

[A-D Control Register (ADCON)] 0032$_{16}$

This register controls A-D converter. Bits 2 to 0 are analog input pin selection bits. Bit 4 is an AD conversion completion bit and "0" during A-D conversion. This bit is set to "1" upon completion of A-D conversion.
A-D conversion is started by setting "0" in this bit.

[Comparison Voltage Generator]

The comparison voltage generator divides the voltage between AVSS and VREF, and outputs the divided voltages.

[Channel Selector]

The channel selector selects one of the input ports P67/AN7–P60/AN0 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 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(XIN) to at least 500 kHz during A-D conversion. Use a CPU system clock dividing the main clock XIN as the internal system clock.

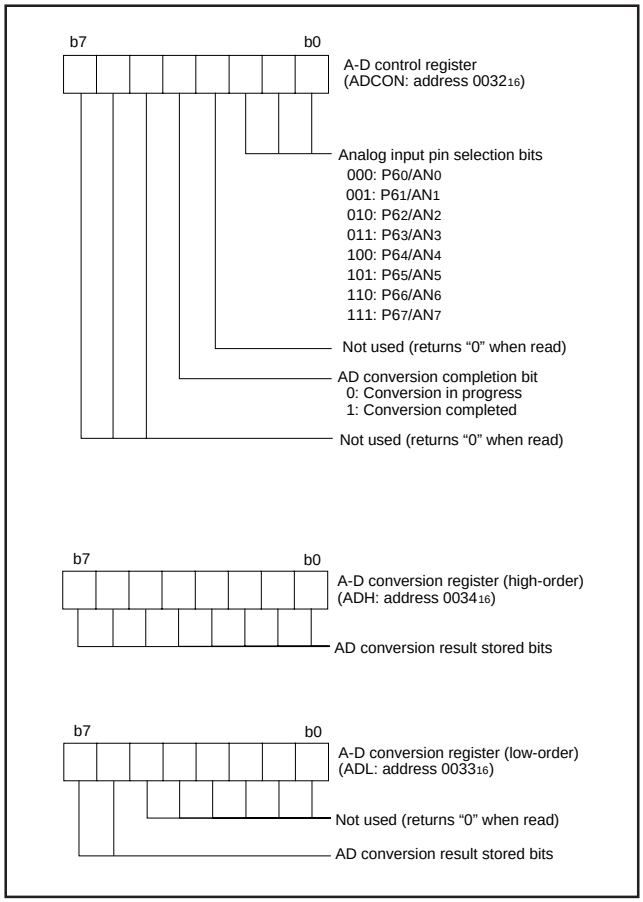


Fig. 27 Structure of A-D control register

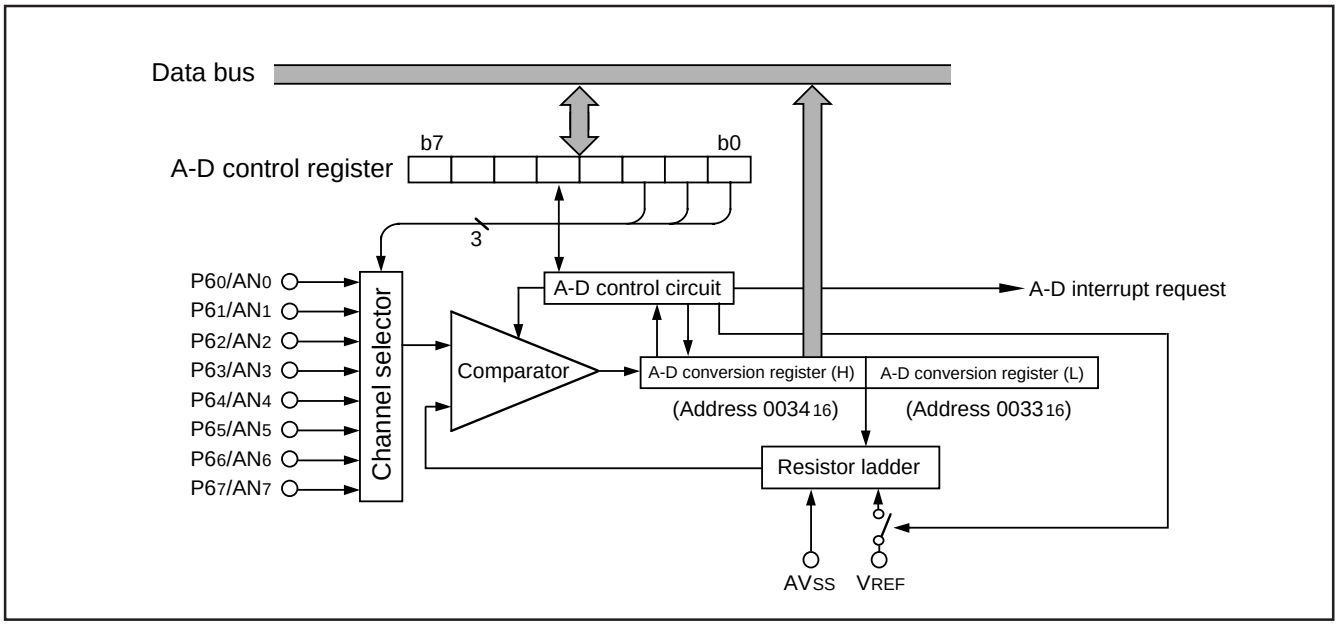


Fig. 28 Block diagram of A-D converter

LCD DRIVE CONTROL CIRCUIT

The 38C3 group has the built-in Liquid Crystal Display (LCD) drive control circuit consisting of the following.

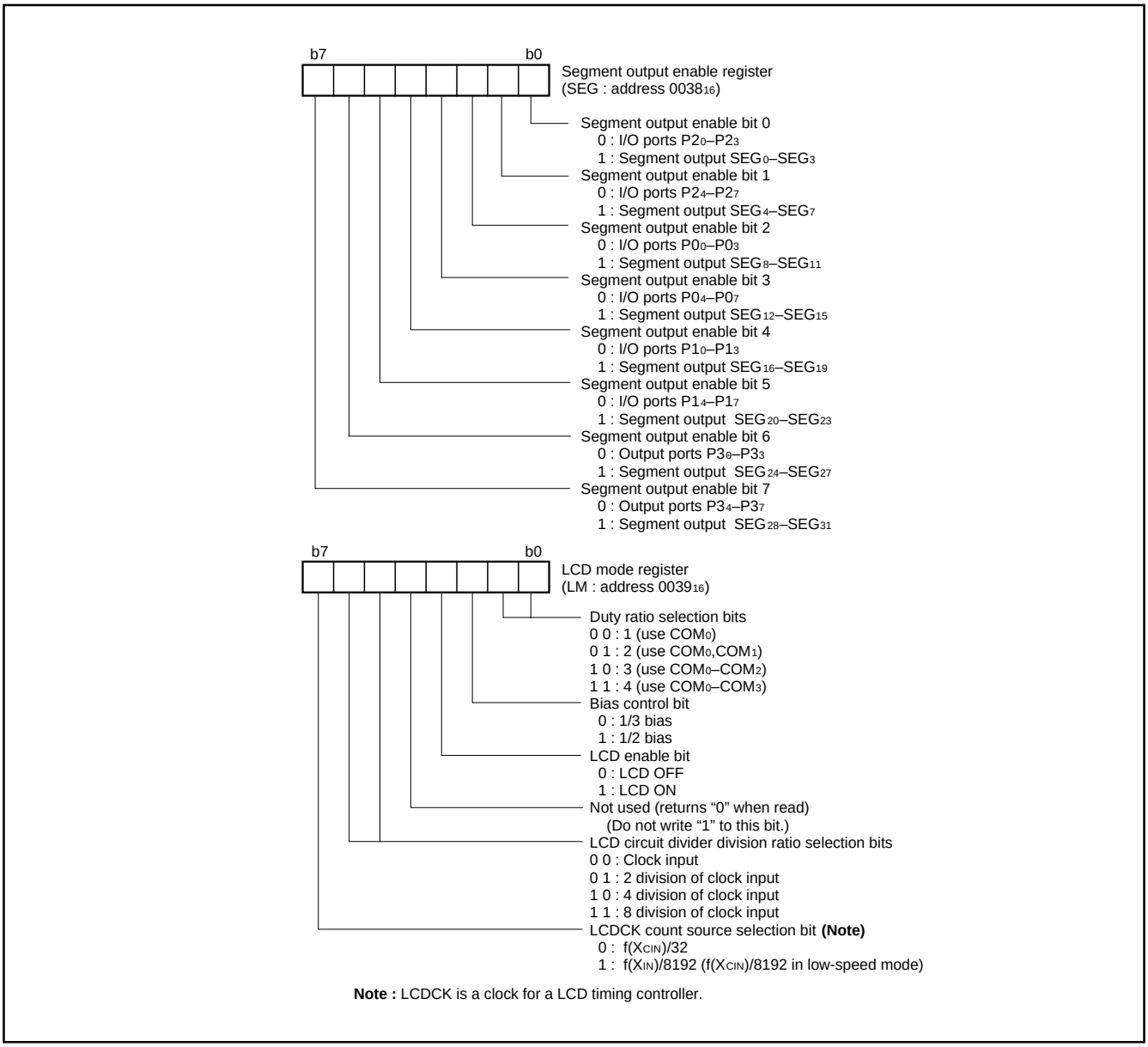
- LCD display RAM
- Segment output enable register
- LCD mode register
- Selector
- Timing controller
- Common driver
- Segment driver
- Bias control circuit

A maximum of 32 segment output pins and 4 common output pins can be used.

Up to 128 pixels can be controlled for a LCD display. When the LCD enable bit is set to “1” after data is set in the LCD mode register, the segment output enable register, and the LCD display RAM, the LCD drive control circuit starts reading the display data automatically, performs the bias control and the duty ratio control, and displays the data on the LCD panel.

Table 10 Maximum number of display pixels at each duty ratio

Duty ratio	Maximum number of display pixels
1	32 dots or 8 segment LCD 4 digits
2	64 dots or 8 segment LCD 8 digits
3	96 dots or 8 segment LCD 12 digits
4	128 dots or 8 segment LCD 16 digits



Note : LCDCK is a clock for a LCD timing controller.

Fig. 29 Structure of LCD related registers

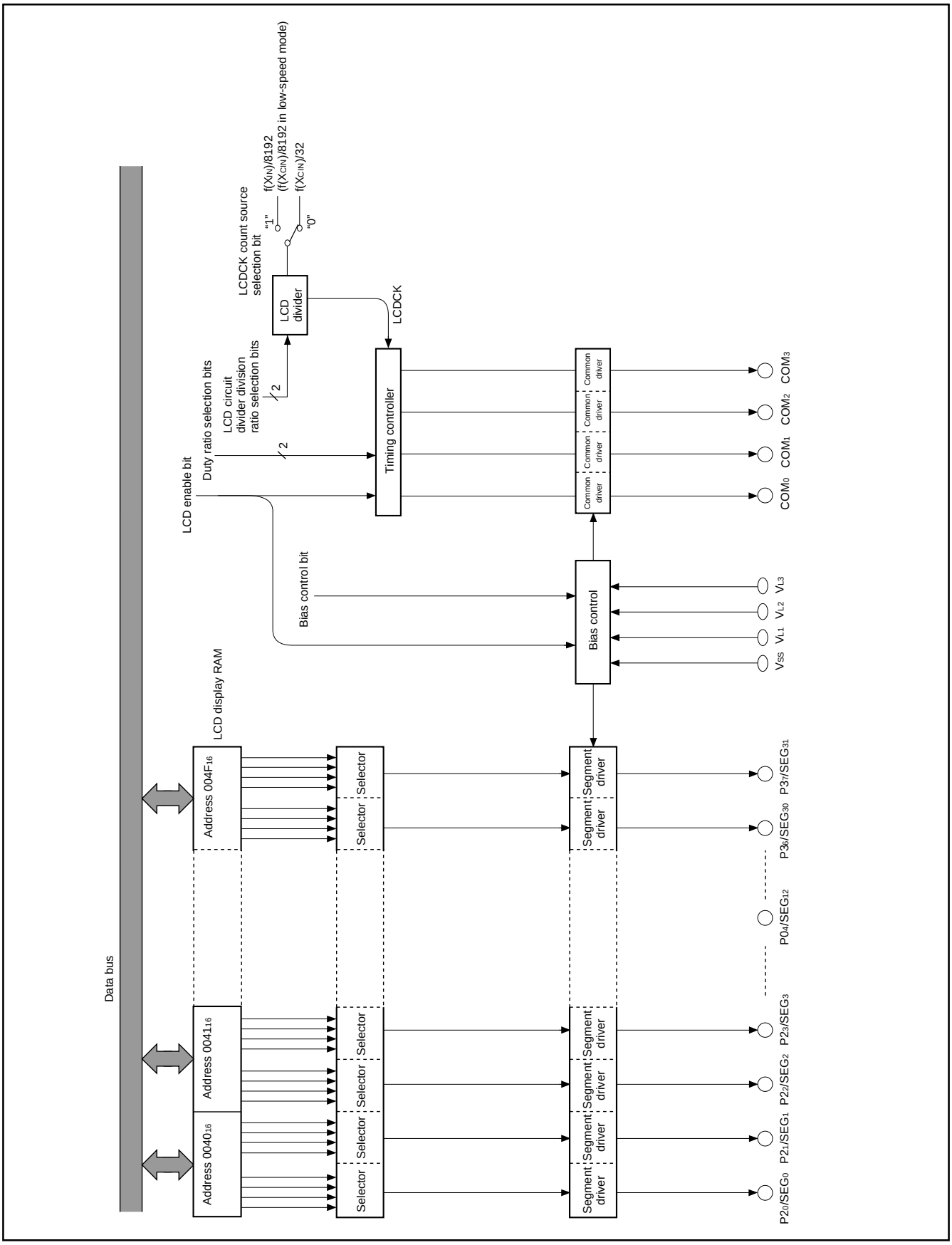


Fig. 30 Block diagram of LCD controller/driver

Bias Control and Applied Voltage to LCD Power Input Pins

To the LCD power input pins (V_{L1}–V_{L3}), apply the voltage value shown in Table 11 according to the bias value.

Select a bias value by the bias control bit (bit 2 of the LCD mode register).

Common Pin and Duty Ratio Control

The common pins (COM_0–COM_3) to be used are determined by duty ratio.

Select duty ratio by the duty ratio selection bits (bits 0 and 1 of the LCD mode register).

When selecting 1-duty ratio, 1/1 bias can be used.

Table 11 Bias control and applied voltage to V_{L1}–V_{L3}

Bias value	Voltage value
1/3 bias	$V_{L3}=V_{LCD}$ $V_{L2}=2/3\ V_{LCD}$ $V_{L1}=1/3\ V_{LCD}$
1/2 bias	$V_{L3}=V_{LCD}$ $V_{L2}=V_{L1}=1/2\ V_{LCD}$
1/1 bias (1-duty ratio)	$V_{L3}=V_{LCD}$ $V_{L2}=V_{L1}=V_{SS}$

Note 1: V_{LCD} is the maximum value of supplied voltage for the LCD panel.

Table 12 Duty ratio control and common pins used

Duty ratio	Duty ratio selection bit		Common pins used
	Bit 1	Bit 0	
1	0	0	COM_0 **(Note 1)**
2	0	1	COM_0, COM_1 **(Note 2)**
3	1	0	COM_0–COM_2 **(Note 3)**
4	1	1	COM_0–COM_3

Notes 1: COM_1, COM_2, and COM_3 are open.
2: COM_2 and COM_3 are open.
3: COM_3 is open.

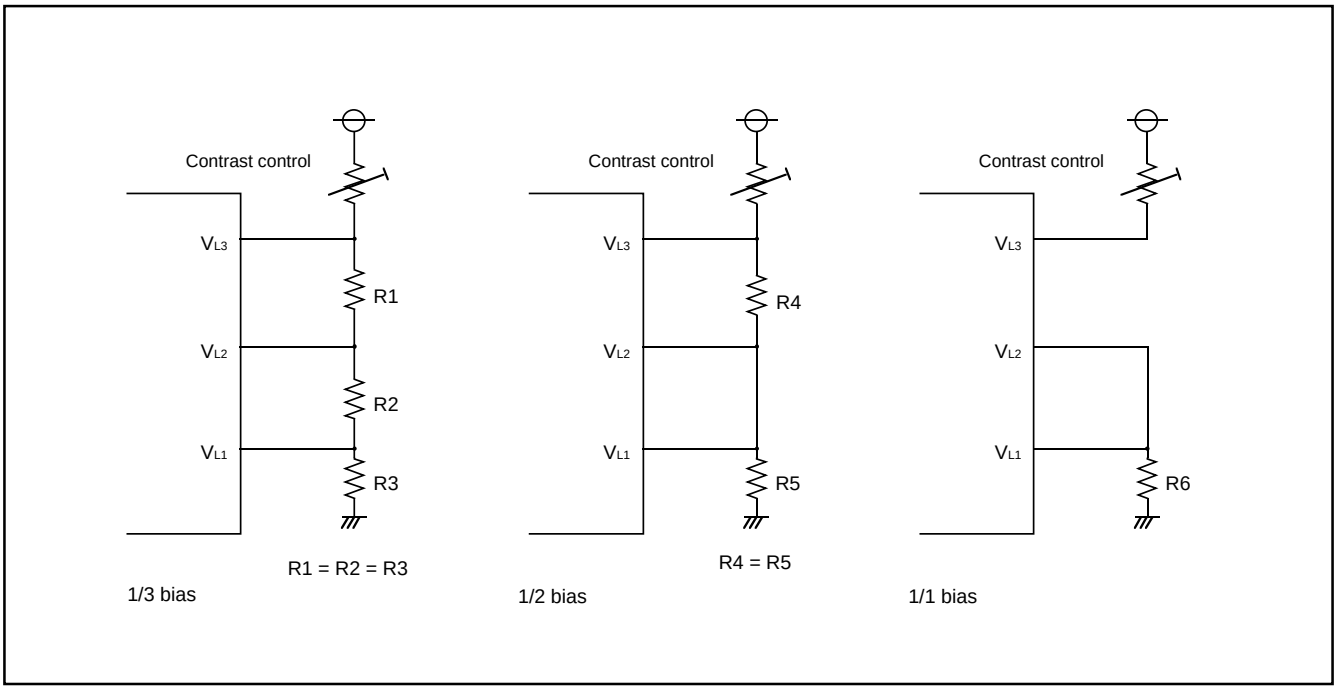


Fig. 31 Example of circuit at each bias

LCD Display RAM

Address 0040_{16} to $004F_{16}$ is the designated RAM for the LCD display. When "1" are written to these addresses, the corresponding segments of the LCD display panel are turned on.

LCD Drive Timing

The LCDCK timing frequency (LCD drive timing) is generated internally and the frame frequency can be determined with the following equation;

$$f(LCDCK) = \frac{\text{(frequency of count source for LCDCK)}}{\text{(divider division ratio for LCD)}}$$

$$\text{Frame frequency} = \frac{f(LCDCK)}{\text{duty ratio}}$$

Address \ Bit	7	6	5	4	3	2	1	0
0040_{16}	SEG_1				SEG_0			
0041_{16}	SEG_3				SEG_2			
0042_{16}	SEG_5				SEG_4			
0043_{16}	SEG_7				SEG_6			
0044_{16}	SEG_9				SEG_8			
0045_{16}	SEG_{11}				SEG_{10}			
0046_{16}	SEG_{13}				SEG_{12}			
0047_{16}	SEG_{15}				SEG_{14}			
0048_{16}	SEG_{17}				SEG_{16}			
0049_{16}	SEG_{19}				SEG_{18}			
$004A_{16}$	SEG_{21}				SEG_{20}			
$004B_{16}$	SEG_{23}				SEG_{22}			
$004C_{16}$	SEG_{25}				SEG_{24}			
$004D_{16}$	SEG_{27}				SEG_{26}			
$004E_{16}$	SEG_{29}				SEG_{28}			
$004F_{16}$	SEG_{31}				SEG_{30}			
	COM_3	COM_2	COM_1	COM_0	COM_3	COM_2	COM_1	COM_0

Fig. 32 LCD display RAM map

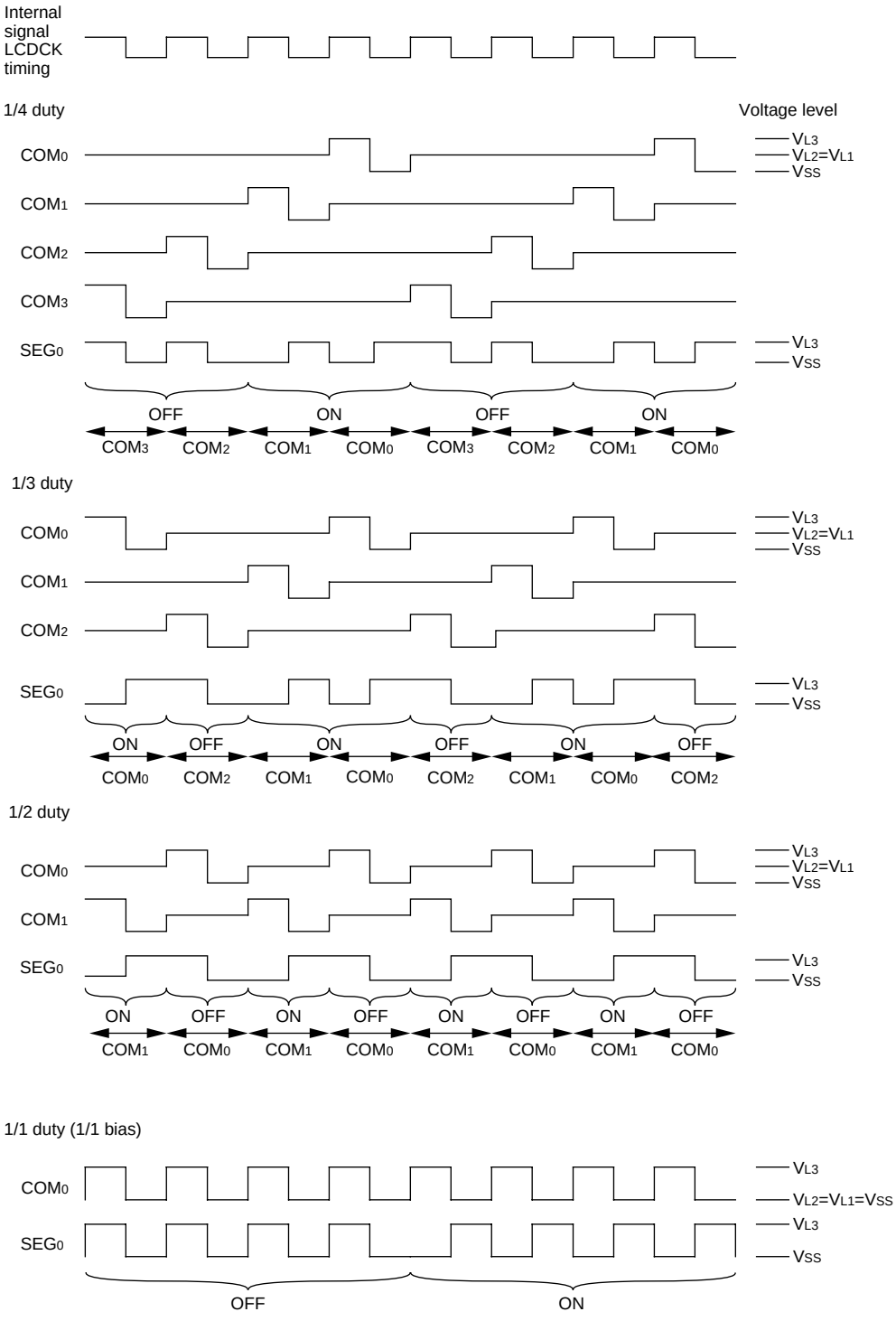


Fig. 33 LCD drive waveform (1/2 bias)

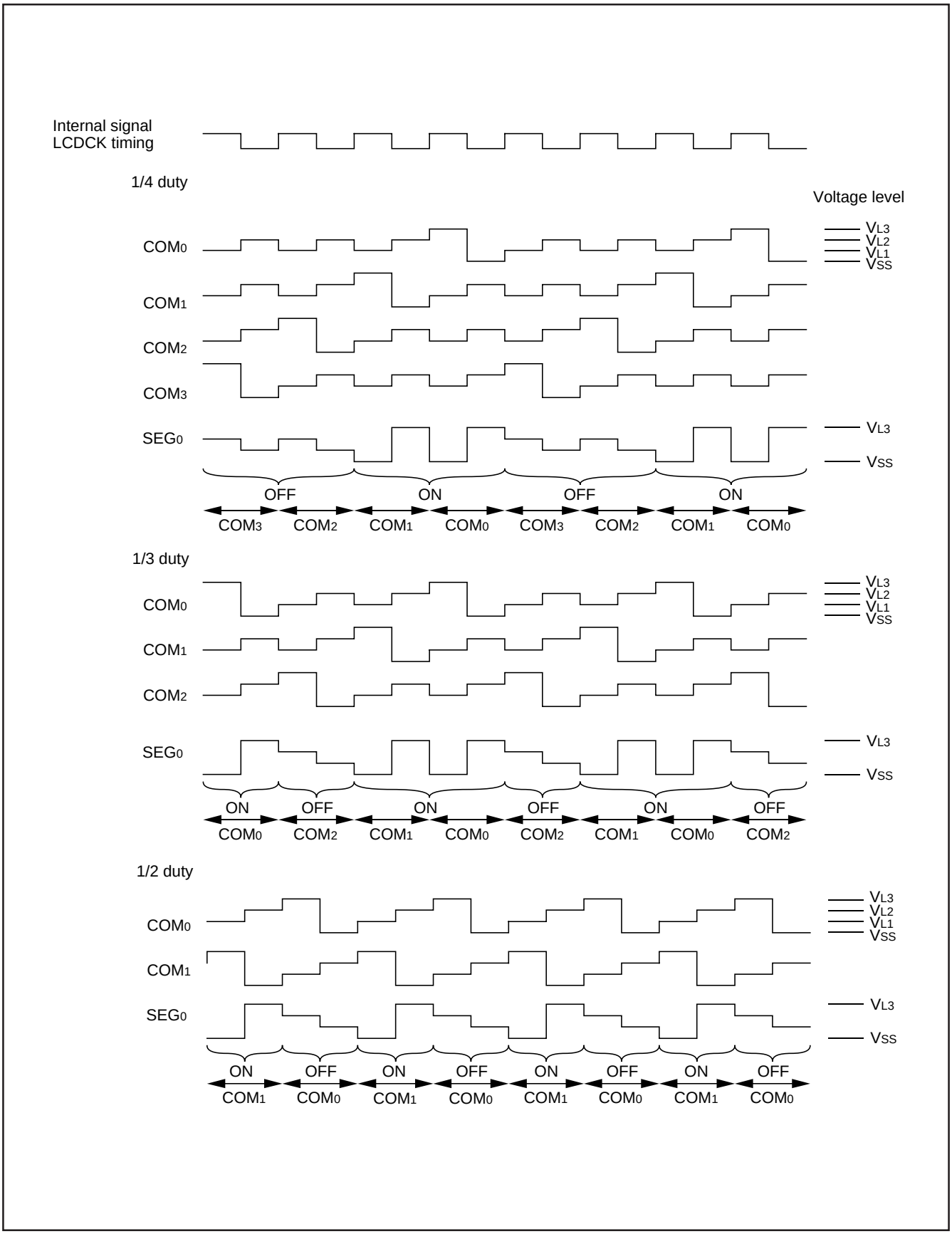


Fig. 34 LCD drive waveform (1/3 bias)

φ CLOCK OUTPUT FUNCTION

The internal system clock φ can be output from port P43 by setting the φ output control register. Set “1” to bit 3 of the port P4 direction register when outputting φ clock.

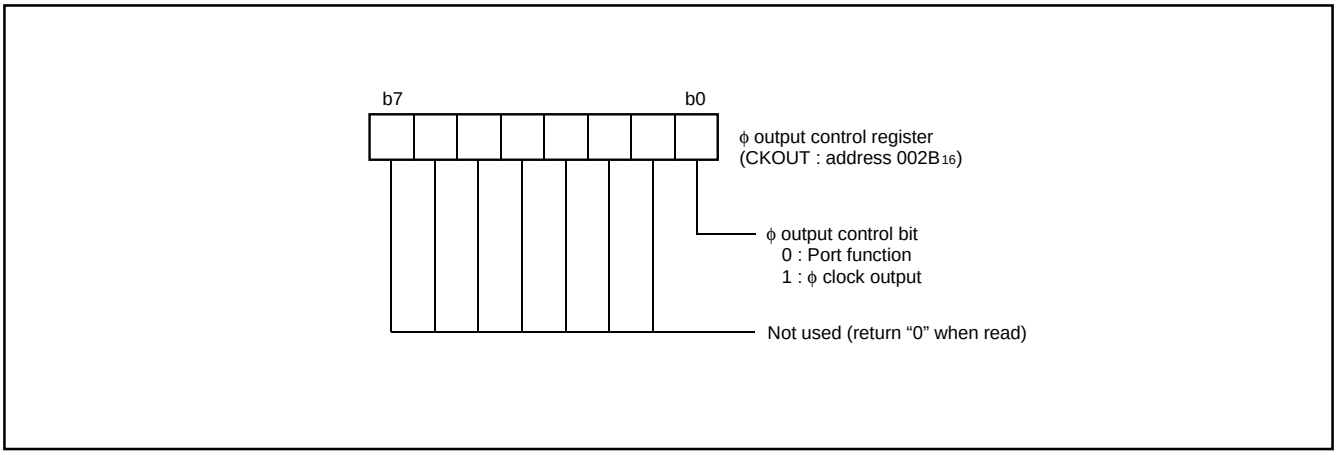


Fig. 35 Structure of φ output control register

ROM CORRECTION FUNCTION (Mask ROM version only)

The 38C3 group has the ROM correction function correcting data at the arbitrary addresses in the ROM area.

[ROM correct address register] 0F02$_{16}$ – 0F11$_{16}$

This is the register to store the address performing ROM correction. There are two types of registers to correct up to 8 addresses: one is the register to store the high-order address and the other is to store the low-order address.

[ROM correct enable register 1 (RC1)] 0F01$_{16}$

This is the register to enable the ROM correction function. When setting the bit corresponding to the ROM correction address to "1", the ROM correction function is enabled.
It becomes invalid to the addresses of which corresponding bit is "0".
All bits are "0" at the initial state.

[ROM correct data]

This is the register to store a correct data for the address specified by the ROM correct address register.

■Notes on ROM correction function

1. To use the ROM correction function, transfer data to each ROM correct data register in the initial setting.
2. Do not specify the same addresses in the ROM correct address register.

0F02$_{16}$	ROM correct high-order address register 1
0F03$_{16}$	ROM correct low-order address register 1
0F04$_{16}$	ROM correct high-order address register 2
0F05$_{16}$	ROM correct low-order address register 2
0F06$_{16}$	ROM correct high-order address register 3
0F07$_{16}$	ROM correct low-order address register 3
0F08$_{16}$	ROM correct high-order address register 4
0F09$_{16}$	ROM correct low-order address register 4
0F0A$_{16}$	ROM correct high-order address register 5
0F0B$_{16}$	ROM correct low-order address register 5
0F0C$_{16}$	ROM correct high-order address register 6
0F0D$_{16}$	ROM correct low-order address register 6
0F0E$_{16}$	ROM correct high-order address register 7
0F0F$_{16}$	ROM correct low-order address register 7
0F10$_{16}$	ROM correct high-order address register 8
0F11$_{16}$	ROM correct low-order address register 8

Fig. 36 Structure of ROM correct address register

0050$_{16}$	ROM correct data 1
0051$_{16}$	ROM correct data 2
0052$_{16}$	ROM correct data 3
0053$_{16}$	ROM correct data 4
0054$_{16}$	ROM correct data 5
0055$_{16}$	ROM correct data 6
0056$_{16}$	ROM correct data 7
0057$_{16}$	ROM correct data 8

Fig. 37 Structure of ROM correct data

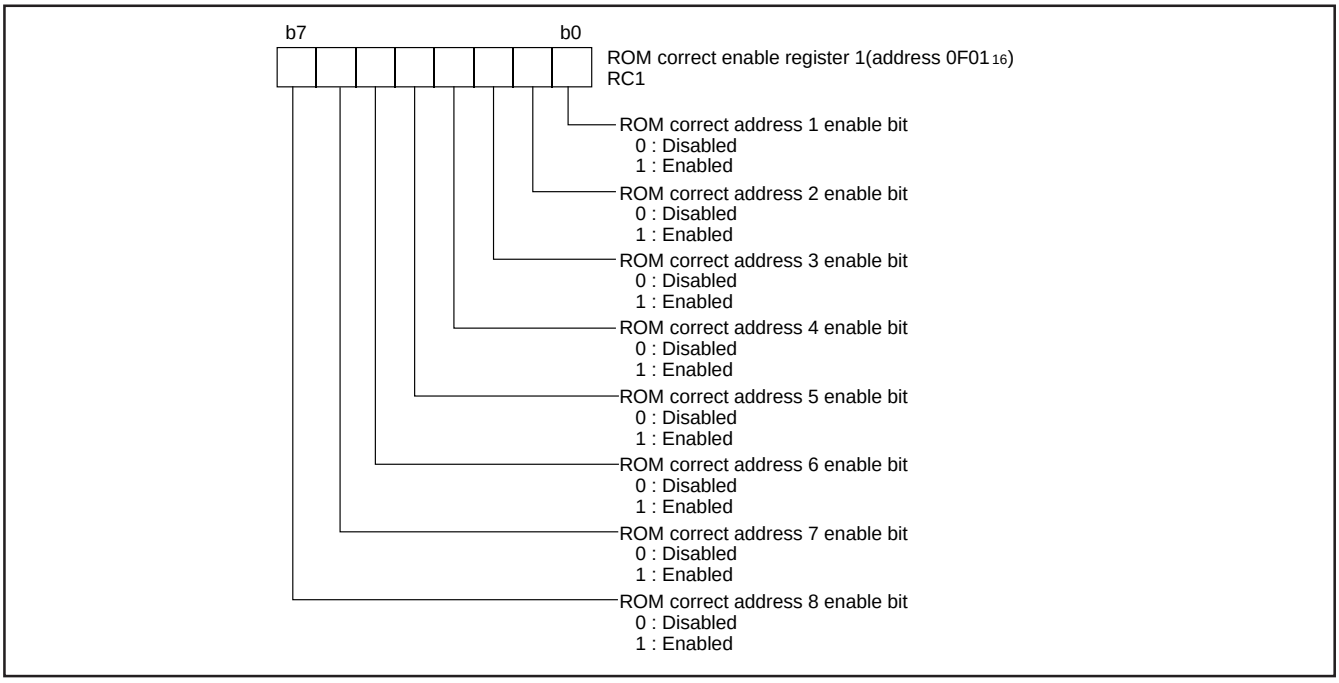


Fig. 38 Structure of ROM correct enable register 1

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 2.5 V and 5.5 V (M version: 2.2* V to 5.5 V), and the oscillation should be stable), reset is released. After the reset is completed, the program starts from the address contained in address FFFD16 (high-order byte) and address FFFC16 (low-order byte). Make sure that the reset input voltage is less than 0.5 V for Vcc of 2.5 V (M version: less than 0.44 V for Vcc of 2.2* V) when switching to the high-speed mode, a power source voltage must be between 4.0 V and 5.5 V.

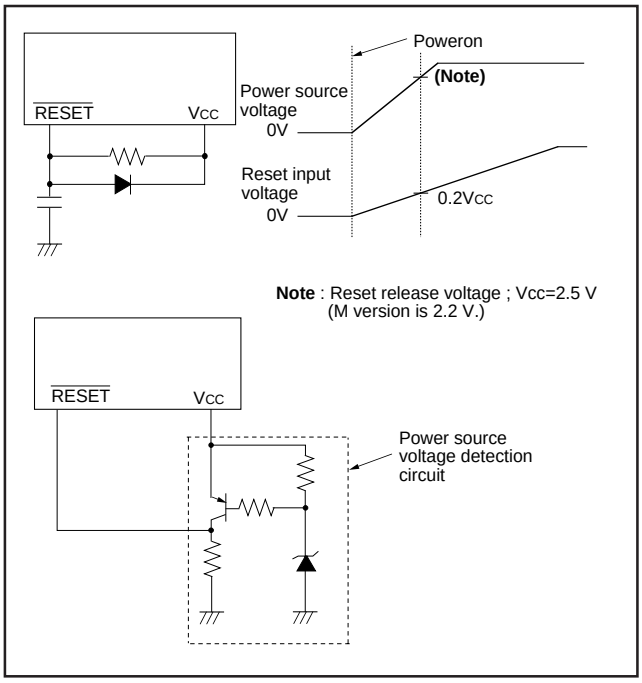
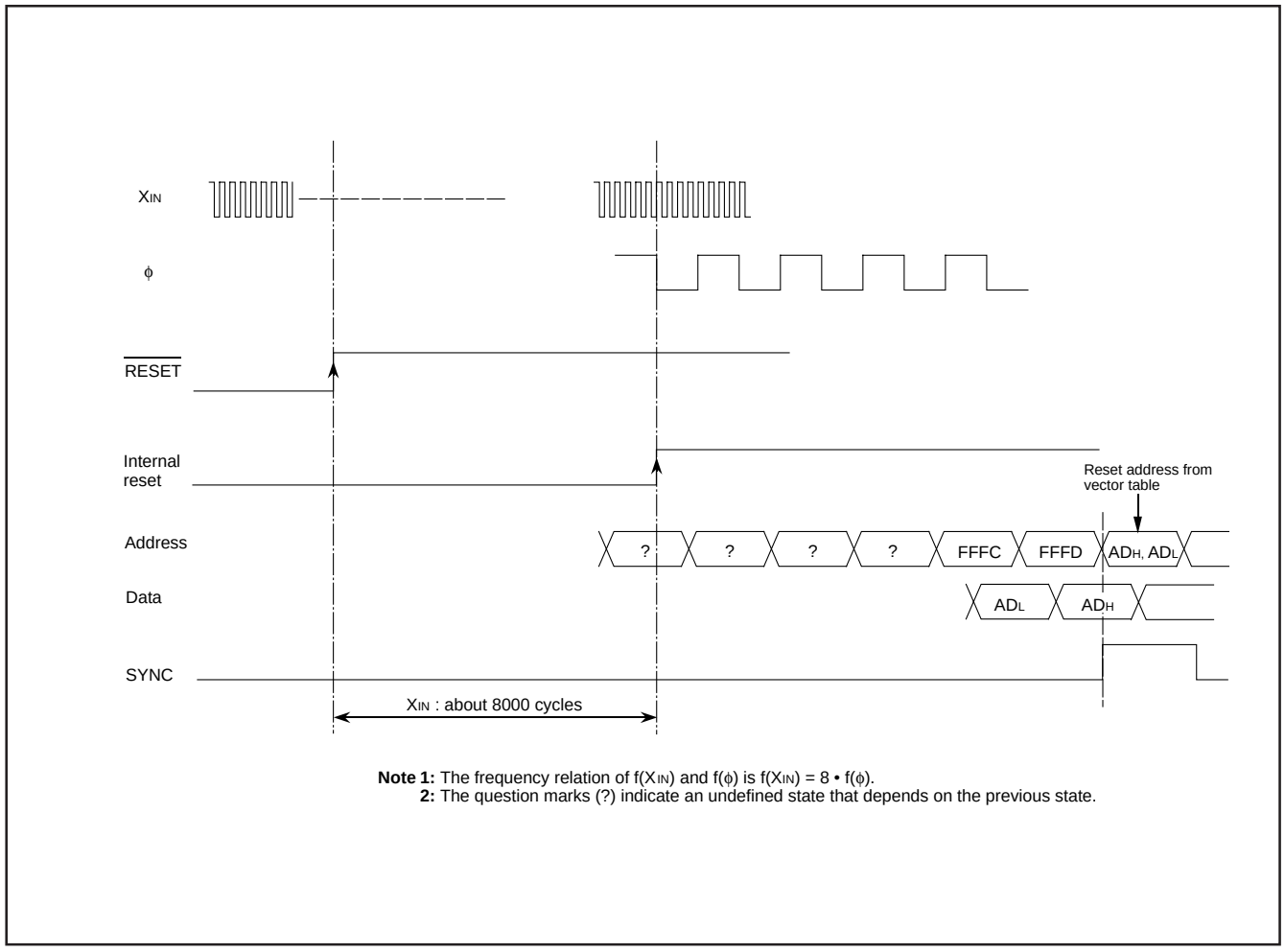


Fig. 39 Reset circuit example



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

Fig. 40 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 P4	0008_{16}	00_{16}
(9) Port P4 direction register	0009_{16}	00_{16}
(10) Port P5	$000A_{16}$	00_{16}
(11) Port P5 direction register	$000B_{16}$	00_{16}
(12) Port P6	$000C_{16}$	00_{16}
(13) Port P6 direction register	$000D_{16}$	00_{16}
(14) Port P7	$000E_{16}$	00_{16}
(15) Port P7 direction register	$000F_{16}$	00_{16}
(16) Port P8	0010_{16}	00_{16}
(17) Port P8 direction register	0011_{16}	00_{16}
(18) PULL register A	0016_{16}	$0F_{16}$
(19) PULL register B	0017_{16}	00_{16}
(20) Port P8 output selection register	0018_{16}	00_{16}
(21) Serial I/O control register 1	0019_{16}	00_{16}
(22) Serial I/O control register 2	$001A_{16}$	00_{16}
(23) Timer 1	0020_{16}	FF_{16}
(24) Timer 2	0021_{16}	01_{16}
(25) Timer 3	0022_{16}	FF_{16}
(26) Timer 4	0023_{16}	FF_{16}
(27) Timer 5	0024_{16}	FF_{16}
(28) Timer 6	0025_{16}	FF_{16}
(29) Timer 12 mode register	0028_{16}	00_{16}
(30) Timer 34 mode register	0029_{16}	00_{16}
(31) Timer 56 mode register	$002A_{16}$	00_{16}
(32) ϕ output control register	$002B_{16}$	00_{16}
(33) Timer A register (low-order)	$002C_{16}$	FF_{16}
(34) Timer A register (high-order)	$002D_{16}$	FF_{16}
(35) Compare register (low-order)	$002E_{16}$	00_{16}
(36) Compare register (high-order)	$002F_{16}$	00_{16}
(37) Timer A mode register	0030_{16}	00_{16}
(38) Timer A control register	0031_{16}	00_{16}
(39) A-D control register	0032_{16}	10_{16}
(40) Segment output enable register	0038_{16}	00_{16}
(41) LCD mode register	0039_{16}	00_{16}
(42) Interrupt edge selection register	$003A_{16}$	00_{16}
(43) CPU mode register	$003B_{16}$	0 1 0 0 1 0 0 0
(44) Interrupt request register 1	$003C_{16}$	00_{16}
(45) Interrupt request register 2	$003D_{16}$	00_{16}
(46) Interrupt control register 1	$003E_{16}$	00_{16}
(47) Interrupt control register 2	$003F_{16}$	00_{16}
(48) ROM correct enable register 1	$0F01_{16}$	00_{16}
(49) ROM correct high-order address register 1	$0F02_{16}$	FF_{16}
(50) ROM correct low-order address register 1	$0F03_{16}$	FF_{16}
(51) ROM correct high-order address register 2	$0F04_{16}$	FF_{16}
(52) ROM correct low-order address register 2	$0F05_{16}$	FF_{16}
(53) ROM correct high-order address register 3	$0F06_{16}$	FF_{16}
(54) ROM correct low-order address register 3	$0F07_{16}$	FF_{16}
(55) ROM correct high-order address register 4	$0F08_{16}$	FF_{16}
(56) ROM correct low-order address register 4	$0F09_{16}$	FF_{16}
(57) ROM correct high-order address register 5	$0F0A_{16}$	FF_{16}
(58) ROM correct low-order address register 5	$0F0B_{16}$	FF_{16}
(59) ROM correct high-order address register 6	$0F0C_{16}$	FF_{16}
(60) ROM correct low-order address register 6	$0F0D_{16}$	FF_{16}
(61) ROM correct high-order address register 7	$0F0E_{16}$	FF_{16}
(62) ROM correct low-order address register 7	$0F0F_{16}$	FF_{16}
(63) ROM correct high-order address register 8	$0F10_{16}$	FF_{16}
(64) ROM correct low-order address register 8	$0F11_{16}$	FF_{16}
(65) Processor status register	(PS)	X X X X X 1 X X
(66) Program counter	(PC_H)	$FFFD_{16}$ contents
	(PC_L)	$FFFC_{16}$ contents

X: Not fixed

Since the initial values for other than above mentioned registers and RAM contents are indefinite at reset, they must be set.

In the M version, bit 0 of the port P5 direction register becomes “1.”

Fig. 41 Internal status at reset

CLOCK GENERATING CIRCUIT

The 38C3 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 feedback resistor exists on-chip. However, an external feedback 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 system clock is the frequency of XIN divided by 8. After reset, this mode is selected.

(2) High-speed mode

The internal system clock is the frequency of XIN divided by 2.

(3) Low-speed mode

The internal system clock is the frequency of XCIN divided by 2.

■Notes on clock generating circuit

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

Oscillation control
(1) Stop mode

If the STP instruction is executed, the internal system clock stops at an "H" level, and XIN and XCIN oscillators stop. Timer 1 is set to "FF16" and timer 2 is set to "0116."

Either XIN divided by 16 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 12 mode register are cleared to "0." Set the interrupt enable bits of the timer 1 and timer 2 to disabled ("0") before executing the STP instruction. Oscillator restarts when an external interrupt is received, but the internal system clock is not supplied to the CPU until timer 2 underflows. This allows time for the clock circuit oscillation to stabilize.

(2) Wait mode

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

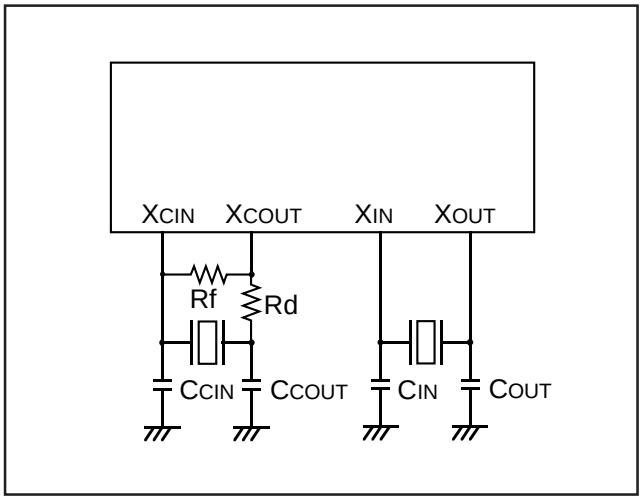


Fig. 42 Ceramic resonator circuit

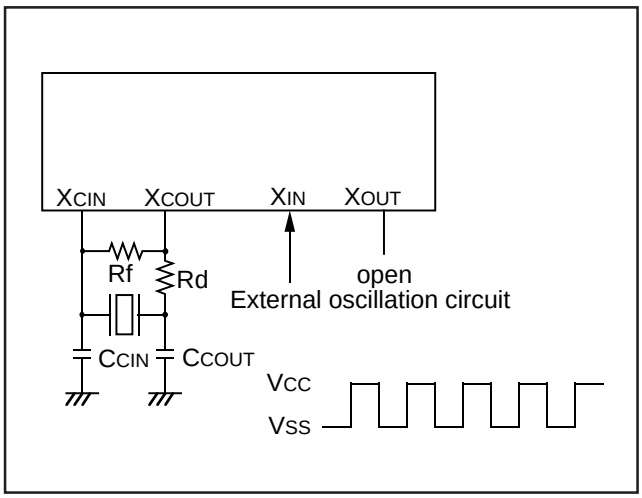
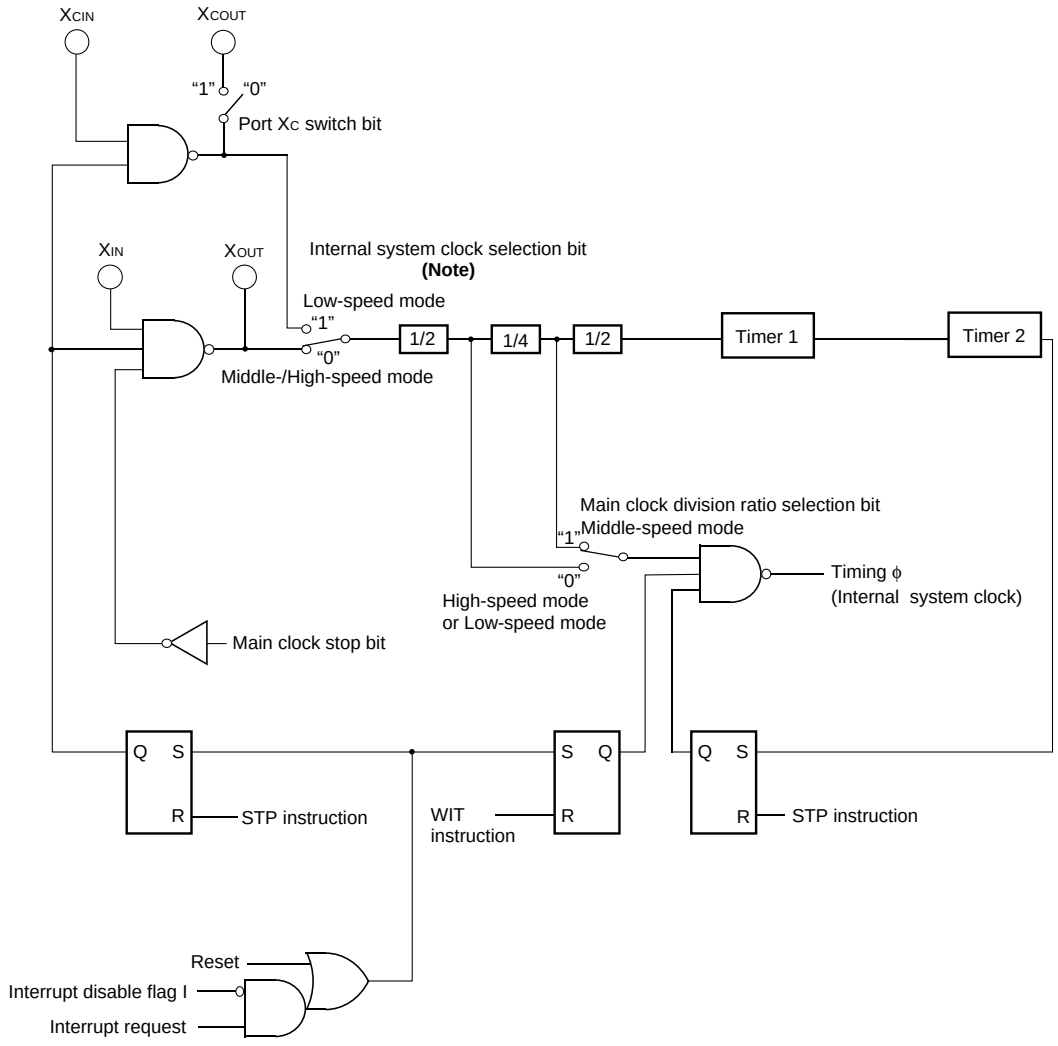
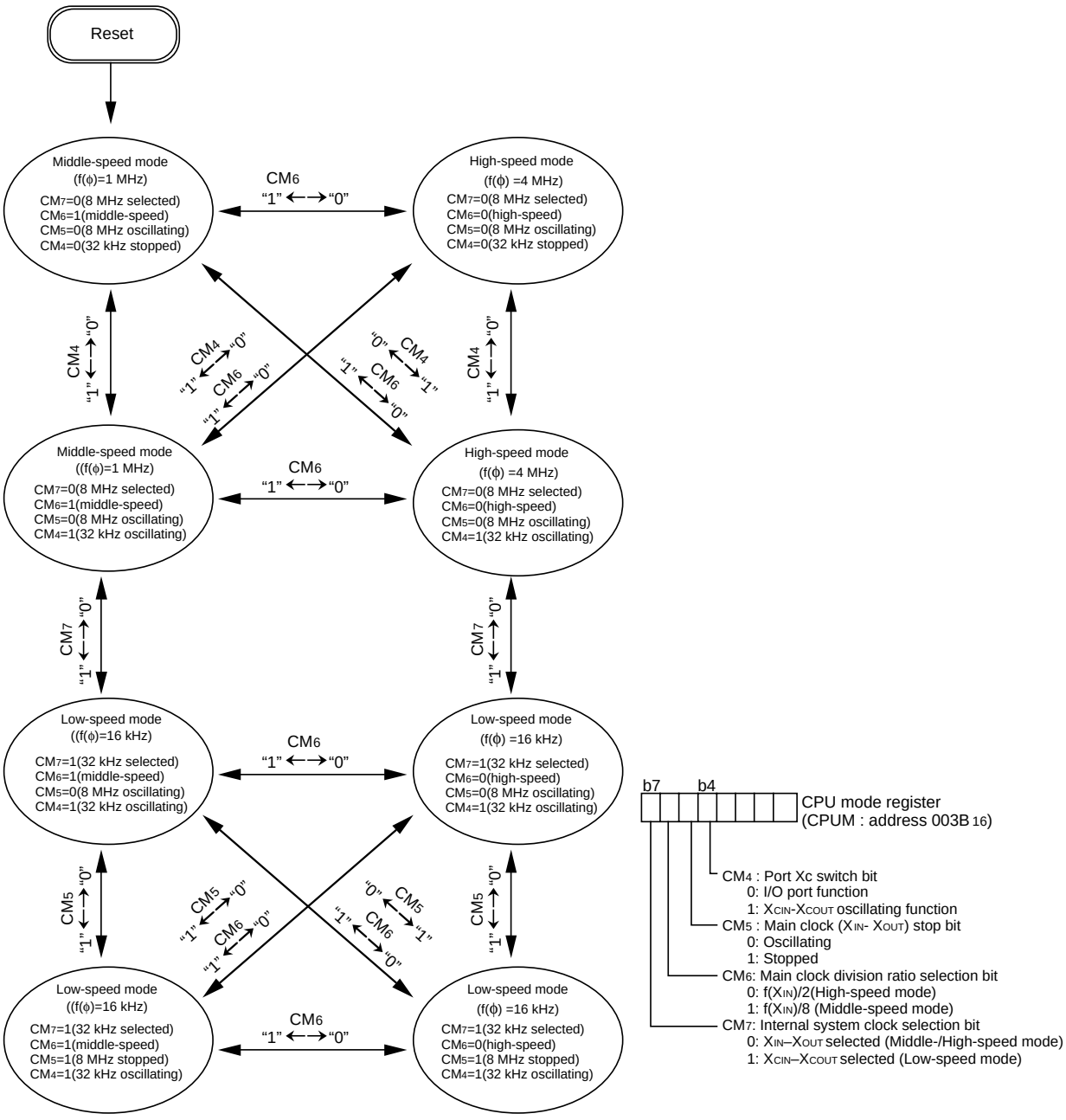


Fig. 43 External clock input circuit



Note : When using the low-speed mode, set the port XC switch bit to "1" .

Fig. 44 Clock generating circuit block diagram



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

Fig. 45 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.

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/O

- Using an external clock
  When using an external clock, input "H" to the external clock input pin and clear the serial I/O interrupt request bit before executing serial I/O transfer and serial I/O automatic transfer.
- Using an internal clock
  When using an internal clock, set the synchronous clock to the internal clock, then clear the serial I/O interrupt request bit before executing a serial I/O transfer and serial I/O automatic transfer.

A-D Converter

The comparator uses internal capacitors 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 system clock 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 system clock is the same half of the X_{IN} frequency in high-speed mode.

At STP Instruction Release

At the STP instruction release, all bits of the timer 12 mode register are cleared.

NOTES ON USE

Notes on Built-in EPROM Version

The $P5_1$ pin of the One Time PROM version or the EPROM version functions as the power source input pin of the internal EPROM.
Therefore, this pin is set at low input impedance, thereby being affected easily by noise.
To prevent a malfunction due to noise, insert a resistor (approx. 5 kΩ) in series with the $P5_1$ pin.

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 built-in PROM of the blank One Time PROM version and built-in EPROM version can be read or programmed with a general-purpose PROM programmer using a special programming adapter.

Table 13 Programming adapter

Package	Name of Programming Adapter
80P6N-A	PCA4738F-80A
80D0	PCA4738L-80A

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 46 is recommended to verify programming.

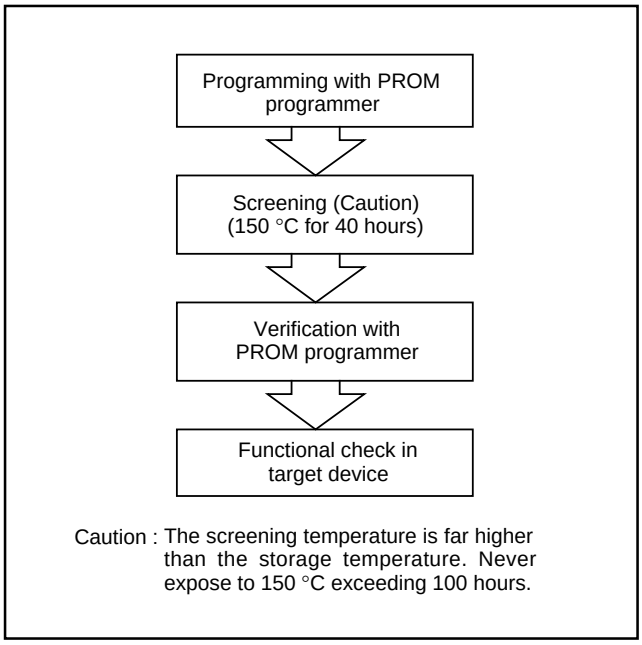


Fig. 46 Programming and testing of One Time PROM version

FUNCTIONAL DESCRIPTION SUPPLEMENT

Interrupt

38C3 group permits interrupts on the basis of 16 sources.
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 interrupt sources, vector addresses and interrupt priority, refer to Table 14.

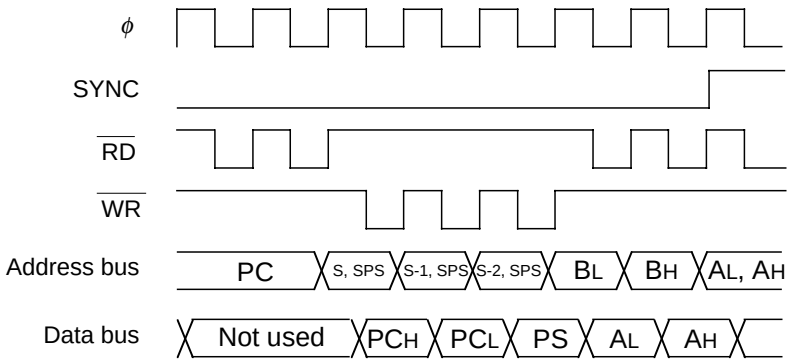
Table 14 Interrupt sources, vector addresses and interrupt priority

Interrupt Source	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)
INT_2	4	$FFF7_{16}$	$FFF6_{16}$	At detection of either rising or falling edge of INT_2 input	External interrupt (active edge selectable)
Serial I/O	5	$FFF5_{16}$	$FFF4_{16}$	At completion of serial I/O data transmit/receive	Valid when serial I/O is selected
Timer A	6	$FFF3_{16}$	$FFF2_{16}$	At timer A underflow	
Timer 1	7	$FFF1_{16}$	$FFF0_{16}$	At timer 1 underflow	
Timer 2	8	$FFEF_{16}$	$FFEE_{16}$	At timer 2 underflow	STP release timer underflow
Timer 3	9	$FFED_{16}$	$FFEC_{16}$	At timer 3 underflow	
Timer 4	10	$FFEB_{16}$	$FFEA_{16}$	At timer 4 underflow	
Timer 5	11	$FFE9_{16}$	$FFE8_{16}$	At timer 5 underflow	
Timer 6	12	$FFE7_{16}$	$FFE6_{16}$	At timer 6 underflow	
$CNTR_0$	13	$FFE5_{16}$	$FFE4_{16}$	At detection of either rising or falling edge of $CNTR_0$ input	External interrupt (active edge selectable)
$CNTR_1$	14	$FFE3_{16}$	$FFE2_{16}$	At detection of either rising or falling edge of $CNTR_1$ input	External interrupt (active edge selectable)
Key input (Key-on wake-up)	15	$FFE1_{16}$	$FFE0_{16}$	At falling of port P8 (at input) input logical level AND	External interrupt (falling valid)
A-D conversion	16	$FFDF_{16}$	$FFDE_{16}$	At completion of A-D conversion	Valid when A-D conversion interrupt 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.

Timing After Interrupt

The interrupt processing routine begins with the machine cycle following the completion of the instruction that is currently in execution. Figure 47 shows a timing chart after an interrupt occurs, and Figure 48 shows the time up to execution of the interrupt processing routine.



SYNC : CPU operation code fetch cycle
(This is an internal signal which cannot be observed from the external unit.)
BL, BH : Vector address of each interrupt
AL, AH : Jump destination address of each interrupt
SPS : "00_{16}" or "01_{16}"

Fig. 47 Timing chart after interrupt occurs

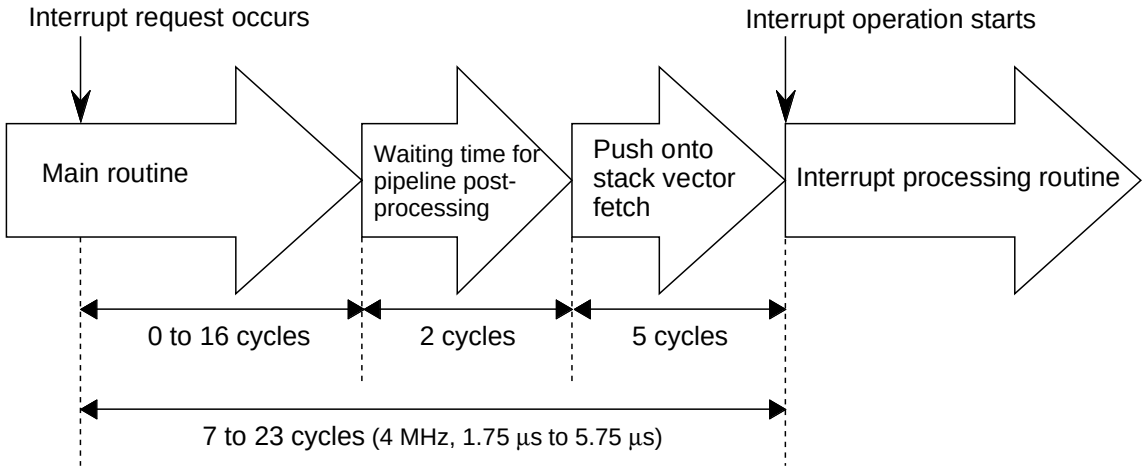


Fig. 48 Time up to execution of interrupt processing routine

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 61 clock cycles (15.25 μs at $f(X_{IN})$ = 8 MHz) 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."

Table 15 Relative formula for a reference voltage V_{REF} of A-D converter and V_{ref}

When n = 0	$V_{ref} = 0$
When n = 1 to 1023	$V_{ref} = \frac{V_{REF}}{1024} \times n$

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

Table 16 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 0	0
First comparison	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$
Second comparison	∗1 1 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4}$
Third comparison	∗1 ∗2 1 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8}$
≈	≈	≈
After completion of tenth comparison	A result of A-D conversion ∗1 ∗2 ∗3 ∗4 ∗5 ∗6 ∗7 ∗8 ∗9 ∗10	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \bullet\bullet\bullet\bullet \pm \frac{V_{REF}}{1024}$

∗1–∗10: A result of the first comparison to the tenth comparison

Figures 49 shows the A-D conversion equivalent circuit, and Figure 50 shows the A-D conversion timing chart.

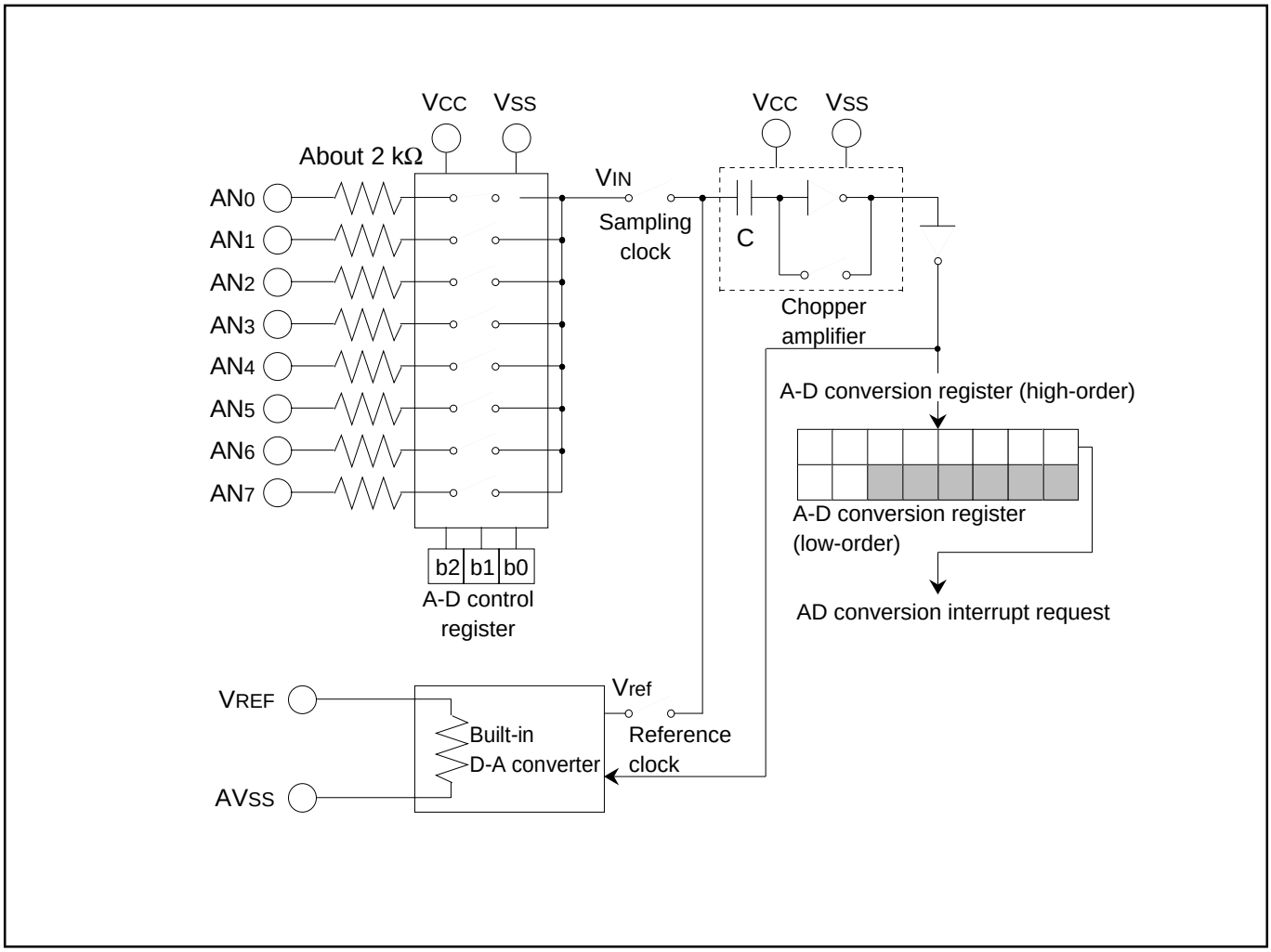


Fig. 49 A-D conversion equivalent circuit

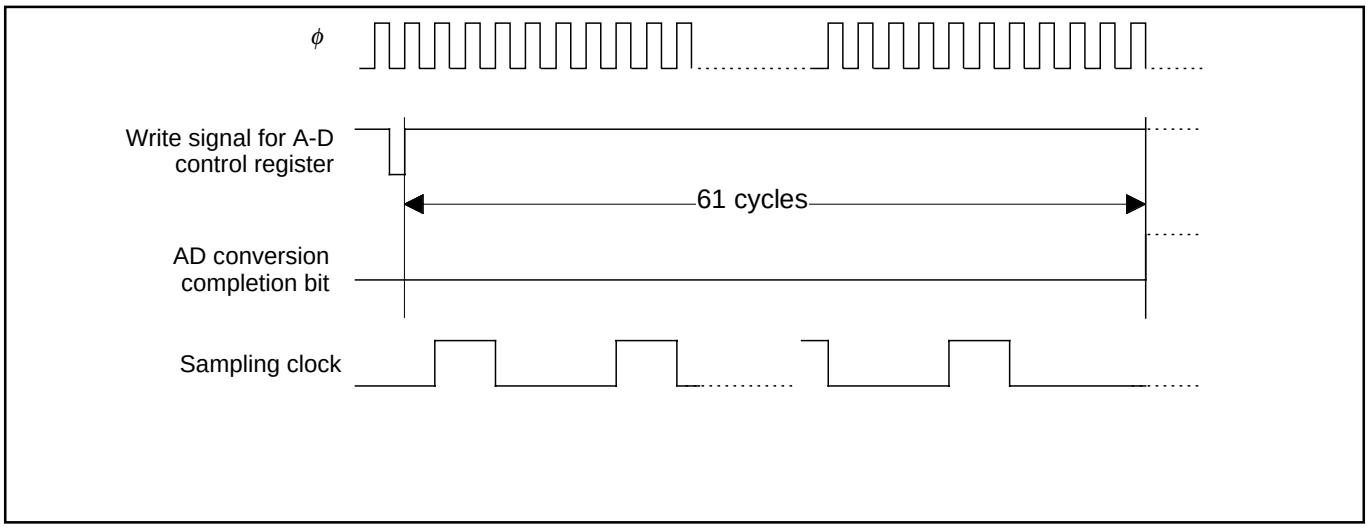


Fig. 50 A-D conversion timing chart

MEMORANDUM

CHAPTER 2

APPLICATION

2.1 I/O port

This paragraph describes the setting method of I/O port relevant registers, notes etc.

2.1.1 Memory map

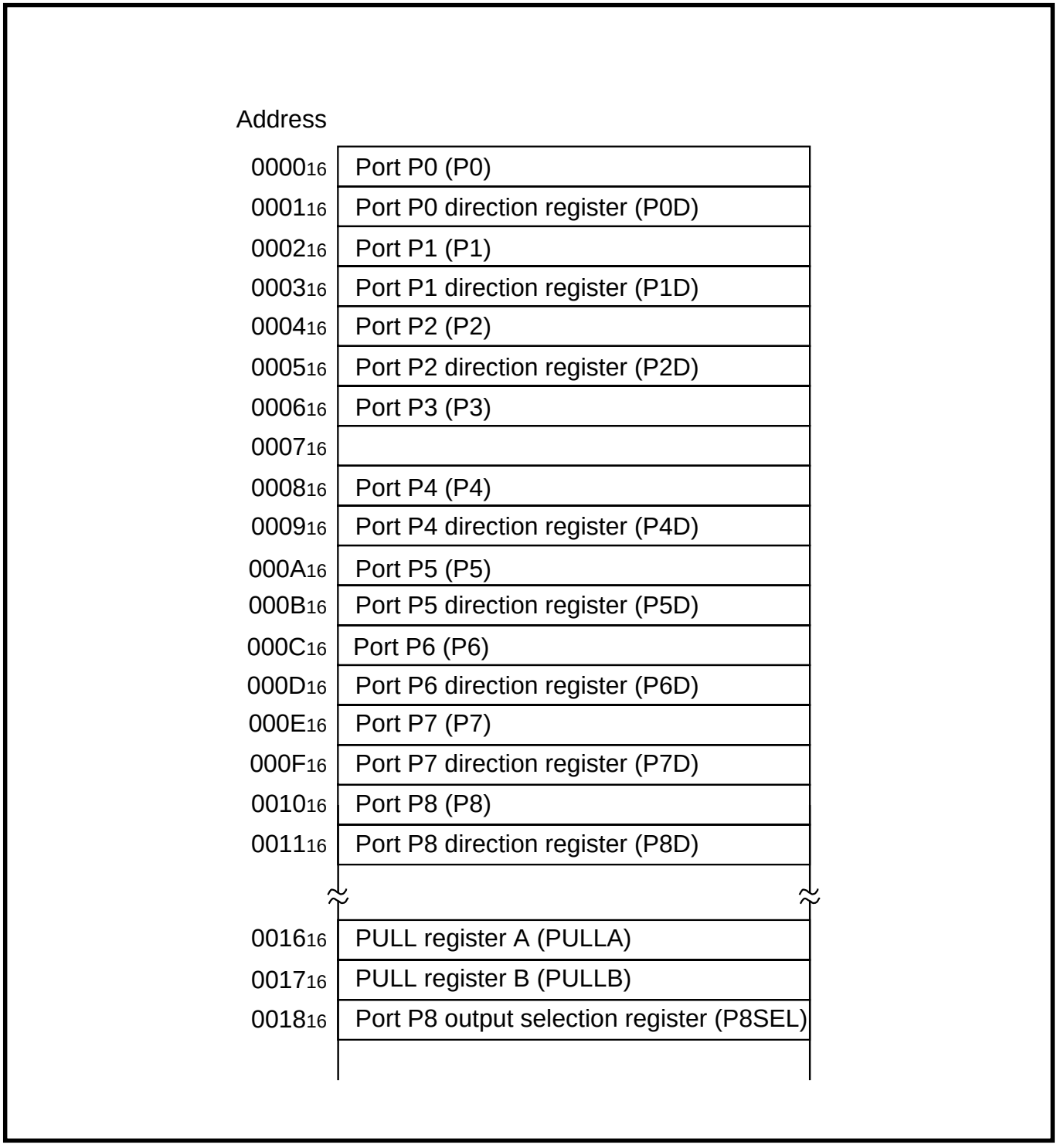


Fig. 2.1.1 Memory map of I/O port relevant registers

2.1.2 Relevant registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (i = 0, 1, 2, 3, 4, 5, 6, 8)
(Pi: addresses 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}, $0A_{16}$, $0C_{16}$, 10_{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. 2.1.2 Structure of port Pi (i = 0, 1, 2, 3, 4, 5, 6, 8)

Port P7

b7 b6 b5 b4 b3 b2 b1 b0

Port P7
(P7: address $0E_{16}$)

b	Name	Functions	At reset	R	W
0	Port $P7_0$	●In output mode Write •••••••• Port latch Read •••••••• Port latch ●In input mode Write •••••••• Port latch Read •••••••• Value of pin	0	○	○
1	Port $P7_1$		0	○	○
2	Nothing is arranged for these bits. When these bits are read out, the contents are undefined.		0	×	×
3			0	×	×
4			0	×	×
5			0	×	×
6			0	×	×
7			0	×	×

Fig. 2.1.3 Structure of port P7

Port P0 direction register, Port P1 direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port P0 direction register (P0D: address 01_{16})
Port P1 direction register (P1D: address 03_{16})

b	Name	Functions	At reset	R	W
0	Ports P0/P1 direction register	0 : All bits of ports P0/P1 input mode 1 : All bits of ports P0/P1 output mode	0	×	○
1	Nothing is arranged for these bits. When these bits are read out, the contents are undefined.		0	×	×
2			0	×	×
3			0	×	×
4			0	×	×
5			0	×	×
6			0	×	×
7			0	×	×

Note: Ports P0 and P1 are switched to input and output by each port. When b0 of corresponding port direction register is set to "0", all 8 bits of port become input port. When b0 of corresponding port direction register is set to "1", all 8 bits of port become output port. Nothing is arranged for b1 to b7 of port P0 and port P1 direction registers. These are write disabled bits.

Fig. 2.1.4 Structure of Port P0 direction register and port P1 direction register

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

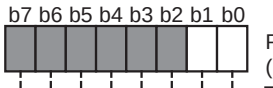
Port Pi direction register (i = 2, 4, 5, 6, 8)
(PiD: addresses 05_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, 11_{16})

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 **(Note)**	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	×	○

Note: Bit 1 of the port P5 direction register (address $0B_{16}$) does not have direction register function, because $P5_1$ is an input port. When writing to bit 1 of the port P5 direction register, write "0" to the bit.

Fig. 2.1.5 Structure of Port Pi direction register (i = 2, 4, 5, 6, 8)

Port P7 direction register

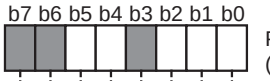


Port P7 direction register
(P7D: address $0F_{16}$)

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	Nothing is arranged for these bits. When these bits are read out, the contents are undefined.		0	×	×
3			0	×	×
4			0	×	×
5			0	×	×
6			0	×	×
7			0	×	×

Fig. 2.1.6 Structure of Port P7 direction register

PULL register A



PULL register A
(PULLA: address 16_{16})

b	Name	Functions	At reset	R	W
0	Port $P0_0$–$P0_7$ pull-down control	0: No pull-down control 1: Pull-down control	1	○	○
1	Port $P1_0$–$P1_7$ pull-down control	0: No pull-down control 1: Pull-down control	1	○	○
2	Port $P2_0$–$P2_7$ pull-down control	0: No pull-down control 1: Pull-down control	1	○	○
3	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “1”.		1	○	×
4	Port $P7_0$, $P7_1$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
5	Port $P8_0$–$P8_7$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
6	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Note: The pin which is set to output port is cut off from pull-up control.

Fig. 2.1.7 Structure of PULL register A

2.1 I/O port

PULL register B

b7 b6 b5 b4 b3 b2 b1 b0

PULL register B
(PULLB: address 17_{16})

b	Name	Functions	At reset	R	W
0	Port $P4_0$–$P4_3$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
1	Port $P4_4$–$P4_7$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
2	Port $P5_0$, $P5_2$, $P5_3$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
3	Port $P5_4$–$P5_7$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
4	Port $P6_0$–$P6_3$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
5	Port $P6_4$–$P6_7$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
6	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Note: The pin which is set to output port is cut off from pull-up control.

Fig. 2.1.8 Structure of PULL register B

Port P8 output selection register

b7 b6 b5 b4 b3 b2 b1 b0

Port P8 output selection register (P8SEL: address 18_{16})

b	Name	Functions	At reset	R	W
0	Port P8 output selection register	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
1		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
2		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
3		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
4		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
6		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
7		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○

Fig. 2.1.9 Structure of Port P8 output selection register

2.1.3 Terminate unused pins

Table 2.1.1 Termination of unused pins

Pins	Termination
P3	Open at "H" output state.
P0, P1, P2, P4, $P5_0$, $P5_2$–$P5_7$, P6, P7, P8	• Set to the input mode and connect each to V_{CC} or V_{SS} through a resistor of 1 kΩ to 10 kΩ. • Set to the output mode and open at "L" or "H" output state.
$P5_1$	Connect to V_{CC} or V_{SS} through a resistor of 1 kΩ to 10 kΩ.
VL_1–VL_3	Connect to Vss (GND).
COM_0–COM_3	Open
V_{REF}	Open
X_{OUT}	Open (only when using external clock)
AV_{SS}	Connect to V_{SS} (GND).

2.1 I/O port

2.1.4 Notes on I/O port

(1) Notes in standby state

In standby state [1] for low-power dissipation, do not make input levels of an input port and an I/O port "undefined".

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

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 standby state: stop mode by executing **STP** instruction
wait mode by executing **WIT** instruction

(2) 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 [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 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.

2 Bit managing instructions: **SEB** and **CLB** instructions

(3) Pull-up/Pull-down control

When each port which has built-in pull-up/pull-down resistor (P0, P1, P2, P4, $P5_0$, $P5_2$–$P5_7$, P6, P7, P8) is set to output port, pull-up/pull-down control of corresponding port become invalid. (Pull-up/Pull-down cannot be set.)

Reason

Pull-up control is valid only when each direction register is set to the input mode.

2.1.5 Termination of unused pins

(1) Terminate unused pins

① Output ports : Open

② Input ports :
Connect each pin to VCC or VSS through each resistor of 1 kΩ to 10 kΩ.
As for pins whose potential affects to operation modes such as pin 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 VCC or VSS through each resistor of 1 kΩ to 10 kΩ.
  Ports that permit the selecting of a built-in pull-up resistor can also use this resistor. Set the I/O ports for the output mode and open them at "L" or "H".
- When opening them in the output mode, the input mode of the initial status remains until the mode of the ports is switched over to the output mode by the program after reset. Thus, the potential at these pins is undefined and the power source current may increase in the input mode. With regard to an effects on the system, thoroughly perform system evaluation on the user side.
- Since the direction register setup may be changed because of a program runaway or noise, set direction registers by program periodically to increase the reliability of program.

(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.

2.2 Timer

This paragraph explains the registers setting method and the notes relevant to the timers.

2.2.1 Memory map

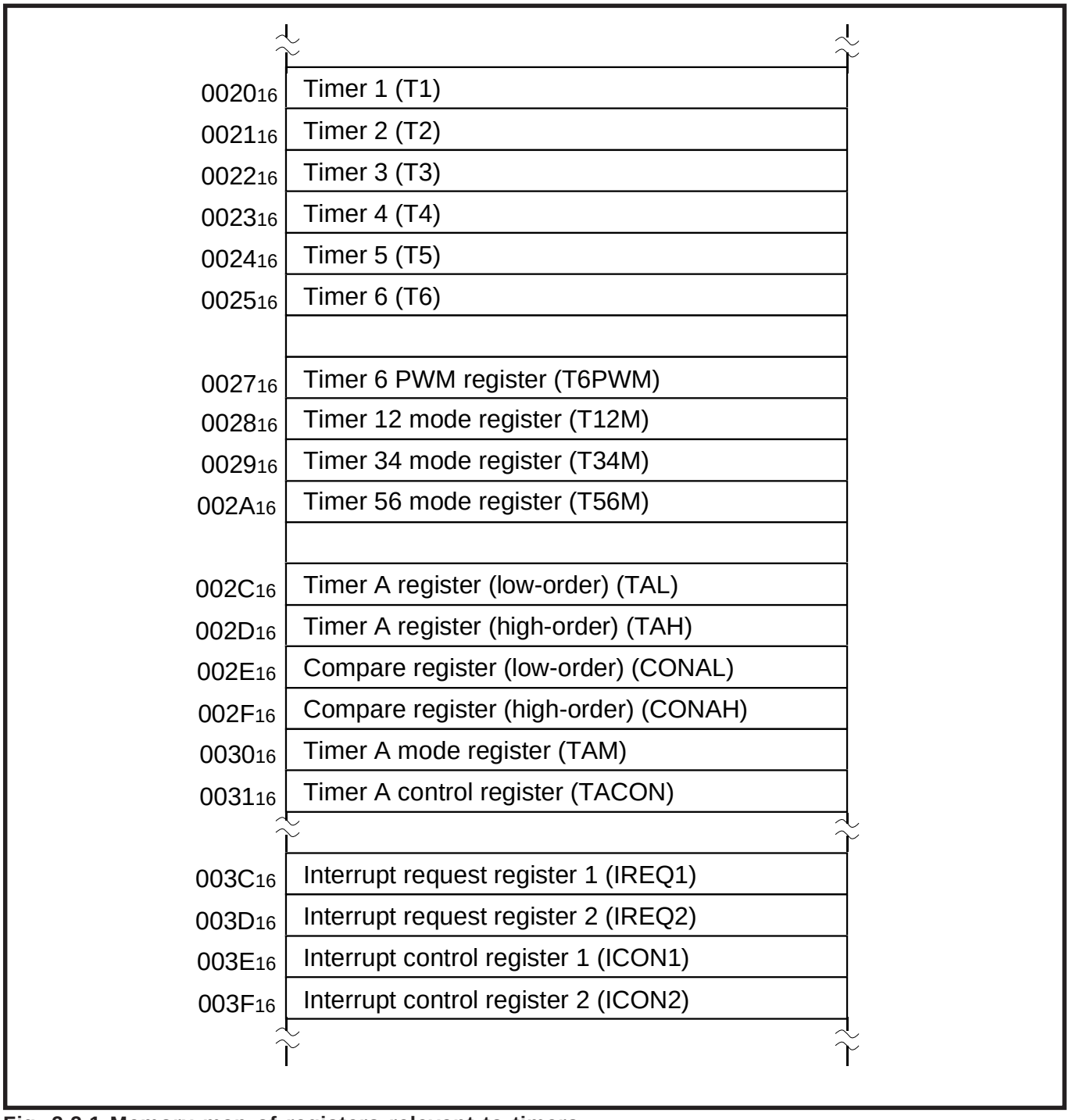


Fig. 2.2.1 Memory map of registers relevant to timers

2.2.2 Relevant registers

(1) 8-bit timer

Timer i

b7 b6 b5 b4 b3 b2 b1 b0

Timer i (i = 1, 3, 4, 5, 6)
(Ti: addresses 20_{16}, 22_{16}, 23_{16}, 24_{16}, 25_{16})

b	Functions	At reset	R	W
0	• Set timer i count value. • The value set in this register is written to both the timer i and the timer i latch at one time. • When the timer i is read out, the count value of the timer i is read out.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 2.2.2 Structure of Timer i (i=1, 3, 4, 5, 6)

Timer 2

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2
(T2: address 21_{16})

b	Functions	At reset	R	W
0	• Set timer 2 count value. • The value set in this register is written to both the timer 2 and the timer 2 latch at one time. • When the timer 2 is read out, the count value of the timer 2 is read out.	1	○	○
1		0	○	○
2		0	○	○
3		0	○	○
4		0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Fig. 2.2.3 Structure of Timer 2

Timer 6 PWM register

b7 b6 b5 b4 b3 b2 b1 b0

Timer 6 PWM register
(T6PWM: address 27_{16})

b	Functions	At reset	R	W
0	• In timer 6 PWM_1 mode "L" level width of PWM rectangular waveform is set. • Duty of PWM rectangular waveform: n/(n + m) Period: (n + m) × ts n = timer 6 set value m = timer 6 PWM register set value ts = timer 6 count source period At n = 0, all PWM output "L". At m = 0, all PWM output "H". (However, n = 0 has priority.) • Selection of timer 6 PWM_1 mode Set "1" to the timer 6 operation mode selection bit.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 2.2.4 Structure of Timer 6 PWM register

Timer 12 mode register

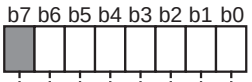
b7 b6 b5 b4 b3 b2 b1 b0

Timer 12 mode register
(T12M: address 28_{16})

b	Name	Functions	At reset	R	W
0	Timer 1 count stop bit	0: Count operation 1: Count stop	0	○	○
1	Timer 2 count stop bit	0: Count operation 1: Count stop	0	○	○
2	Timer 1 count source selection bits	b3 b2 0 0: $f(X_{IN})/16$ or $f(X_{CIN})/16$ 0 1: $f(X_{CIN})$ 1 0: $f(X_{IN})/32$ or $f(X_{CIN})/32$ 1 1: $f(X_{IN})/128$ or $f(X_{CIN})/128$	0	○	○
3			0	○	○
4	Timer 2 count source selection bits	b5 b4 0 0: Timer 1 underflow 0 1: $f(X_{CIN})$ 1 0: External count input $CNTR_0$ 1 1: Not available	0	○	○
5			0	○	○
6	Timer 1 output selection bit ($P4_1$)	0: I/O port 1: Timer 1 output	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×

Fig. 2.2.5 Structure of Timer 12 mode register

Timer 34 mode register

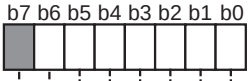


Timer 34 mode register
(T34M: address 29_{16})

b	Name	Functions	At reset	R	W
0	Timer 3 count stop bit	0: Count operation 1: Count stop	0	○	○
1	Timer 4 count stop bit	0: Count operation 1: Count stop	0	○	○
2 3	Timer 3 count source selection bits	b3 b2 0 0: f(XIN)/16 or f(XCIN)/16 0 1: Timer 2 underflow 1 0: f(XIN)/32 or f(XCIN)/32 1 1: f(XIN)/128 or f(XCIN)/128	0	○	○
4	Timer 4 count source selection bits	b5 b4 0 0: f(XIN)/16 or f(XCIN)/16 0 1: Timer 3 underflow 1 0: External count input CNTR1 1 1: Not available	0	○	○
5			0	○	○
6	Timer 3 output selection bit (P42)	0: I/O port 1: Timer 3 output	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Fig. 2.2.6 Structure of Timer 34 mode register

Timer 56 mode register



Timer 56 mode register
(T56M: address $2A_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 5 count stop bit	0: Count operation 1: Count stop	0	○	○
1	Timer 6 count stop bit	0: Count operation 1: Count stop	0	○	○
2	Timer 5 count source selection bit	0: f(XIN)/16 or f(XCIN)/16 1: Timer 4 underflow	0	○	○
3	Timer 6 operation mode selection bit	0: Timer mode 1: PWM mode	0	○	○
4	Timer 6 count source selection bits	b5 b4 0 0: f(XIN)/16 or f(XCIN)/16 0 1: Timer 5 underflow 1 0: Timer 4 underflow 1 1: Not available	0	○	○
5			0	○	○
6	Timer 6 (PWM) output selection bit (P52)	0: I/O port 1: Timer 6 output	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Fig. 2.2.7 Structure of Timer 56 mode register

(2) 16-bit timer

Timer A register (low-order, high-order)

b7 b6 b5 b4 b3 b2 b1 b0

Timer A register (low-order, high-order)
(TAL, TAH: addresses $2C_{16}$, $2D_{16}$)

b	Functions	At reset	R	W
0	• Set timer A count value. • When the timer A write control bit of the timer A mode register is "0", the value is written to timer A and the latch at one time. When the timer A write control bit of the timer A mode register is "1", the value is written only to the latch. • The timer A 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 TAH and TAL following.
3: Write both registers in order of TAL and TAH following.
4: Do not read both registers during a write, and do not write to both registers during a read.

Fig. 2.2.8 Structure of Timer A register (low-order, high-order)

Compare register (low-order, high-order)

b7 b6 b5 b4 b3 b2 b1 b0

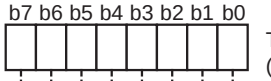
Compare register (low-order, high-order)
(CONAL, CONAH: addresses $2E_{16}$, $2F_{16}$)

b	Functions	At reset	R	W
0	• Set compare register value.	0	○	○
1		0	○	○
2		0	○	○
3		0	○	○
4		0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Note: Write registers in order of CONAH, CONAL, TAL, and TAH following.

Fig. 2.2.9 Structure of Compare register (low-order, high-order)

Timer A mode register

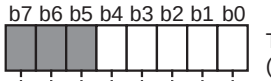


Timer A mode register
(TAM: address 30_{16})

b	Name	Functions	At reset	R	W
0	Timer A operating mode bits	b1b0 0 0: Timer mode 0 1: Pulse output mode 1 0: IGBT output mode 1 1: PWM mode	0	○	○
1			0	○	○
2	Timer A write control bit	0: Write data to both timer latch and timer 1: Write data to timer latch	0	○	○
3	Timer A count source selection bits	b4b3 0 0: f(X_{IN}) 0 1: f(X_{IN})/2 1 0: f(X_{IN})/4 1 1: f(X_{IN})/8	0	○	○
4			0	○	○
5	Timer A output active edge switch bit	0: Output starts with "L" level 1: Output starts with "H" level	0	○	○
6	Timer A count stop bit	0: Count operating 1: Count stop	0	○	○
7	Timer A output selection bit ($P5_0$)	0: I/O port 1: Timer A output	0	○	○

Fig. 2.2.10 Structure of Timer A mode register

Timer A control register



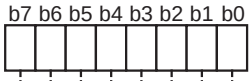
Timer A control register
(TACON: address 31_{16})

b	Name	Functions	At reset	R	W
0	Noise filter sampling clock selection bit	0: f(X_{IN})/2 1: f(X_{IN})/4	0	○	○
1	External trigger delay time selection bits	b2b1 0 0: No delay 0 1: (4/f(X_{IN}))μs 1 0: (8/f(X_{IN}))μs 1 1: (16/f(X_{IN}))μs	0	○	○
2			0	○	○
3	Timer A output control bit 1 ($P5_6$)	0: Not used 1: INT1 interrupt used	0	○	○
4	Timer A output control bit 2 ($P5_7$)	0: Not used 1: INT2 interrupt used	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
6			0	○	×
7			0	○	×

Fig. 2.2.11 Structure of Timer A control register

(3) 8-bit timer, 16-bit timer

Interrupt request register 1



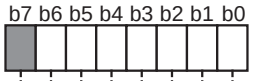
Interrupt request register 1
(IREQ1 : address $3C_{16}$)

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	INT_2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer A interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer 1 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. 2.2.12 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	Timer 4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Timer 5 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Timer 6 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	$CNTR_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	CNTR1 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	AD conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

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

Fig. 2.2.13 Structure of Interrupt request register 2

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

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	INT_2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer A interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 1 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. 2.2.14 Structure of Interrupt control register 1

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0

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

b	Name	Functions	At reset	R	W
0	Timer 4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Timer 5 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
2	Timer 6 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
3	$CNTR_0$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
4	$CNTR_1$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key input interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD conversion interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Fig. 2.2.15 Structure of Interrupt control register 2

2.2.3 Timer application examples

(1) Basic functions and uses

[Function 1] Control of event interval (Timer 1 to Timer 6, Timer A: timer mode)

When a certain time, by setting a count value to each timer, has passed, the timer interrupt request occurs.

<Use>

•Generating of an output signal timing

•Generating of a wait time

[Function 2] Control of cyclic operation (Timer 1 to Timer 6, Timer A: timer mode)

The value of the timer latch is automatically written to the corresponding timer each time the timer underflows, and each timer interrupt request occurs in cycles.

<Use>

•Generating of cyclic interrupts

•Clock function (measurement of 1 s); see "(2) Timer application example 1"

•Control of a main routine cycle

[Function 3] Output of rectangular waveform (Timer 1, Timer 3, Timer 6, Timer A: pulse output mode)

The output level of the $T1_{OUT}$ pin, $T3_{OUT}$ pin, PWM_1 pin or TA_{OUT} pin is inverted each time the timer underflows.

<Use>

•Piezoelectric buzzer output; see "(3) Timer application example 2"

•Generating of the remote control carrier waveforms

[Function 4] Count of external pulses (Timer 2, Timer 4)

External pulses input to the $CNTR_0$ pin, $CNTR_1$ pin are counted as the timer count source (in the event counter mode).

<Use>

•Frequency measurement; see "(4) Timer application example 3"

•Division of external pulses

•Generating of interrupts due to a cycle using external pulses as the count source; count of a reel pulse

[Function 5] Output of PWM signal (Timer 6)

"H" interval and "L" interval are specified, respectively, and the output of pulses from $P5_2/PWM_1$ pin is repeated.

<Use>

•Control of electric volume

[Function 6] Output of IGBT control signal (Timer A: IGBT output mode)

The external signal which is input to INT_0 pin is used as trigger, and the period and "H" interval are specified, respectively, and the output of pulses from $P5_0/TA_{OUT}$ pin is repeated.

<Use>

•IGBT control of IH heat equipment; see "(5) Timer application example 4"

•IGBT control to magnetron

[Function 7] Output of PWM signal (Timer A: PWM mode)

The cycle and "H" interval are specified, respectively, and the output of pulses from $P5_0/TA_{OUT}$ pin is repeated.

<Use>

•Control of electric volume

•IGBT control of IH heat equipment

(2) Timer application example 1: Clock function (measurement of 1 s)

Outline: The input clock is divided by the timer so that the clock can count up at 1 s intervals.

Specifications: •The clock f(X_{IN}) = 4.19 MHz (2^{22} Hz) is divided by the timer.
•The timer 3 interrupt request bit is checked in main routine, and if the interrupt request is issued, the clock is counted up.
• The timer 1 interrupt occurs every 244 μs to execute processing of other interrupts.

Figure 2.2.16 shows the timers connection and setting of division ratios; Figure 2.2.17 shows the relevant registers setting; Figure 2.2.18 shows the control procedure.

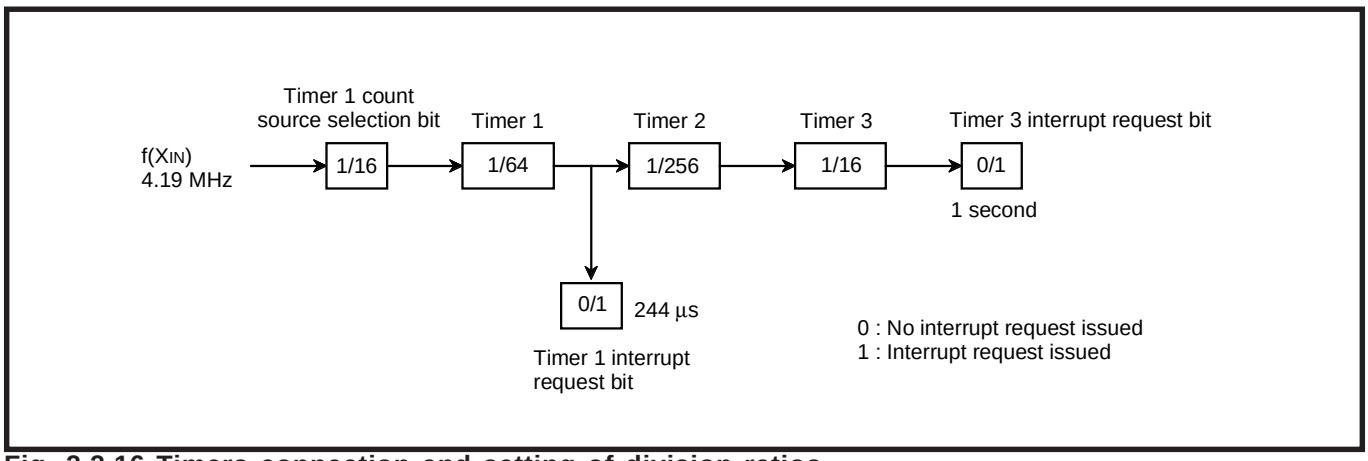


Fig. 2.2.16 Timers connection and setting of division ratios

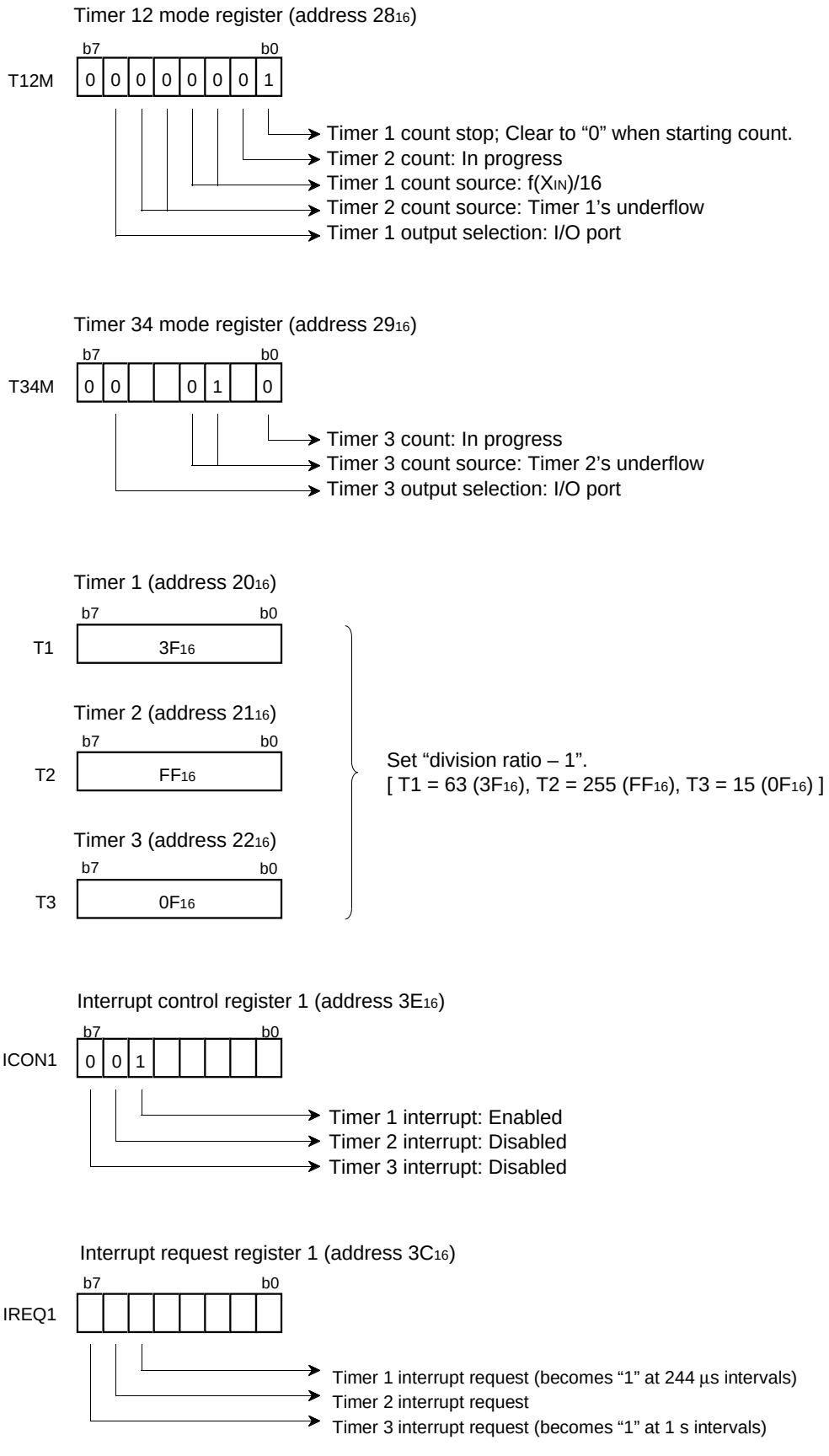
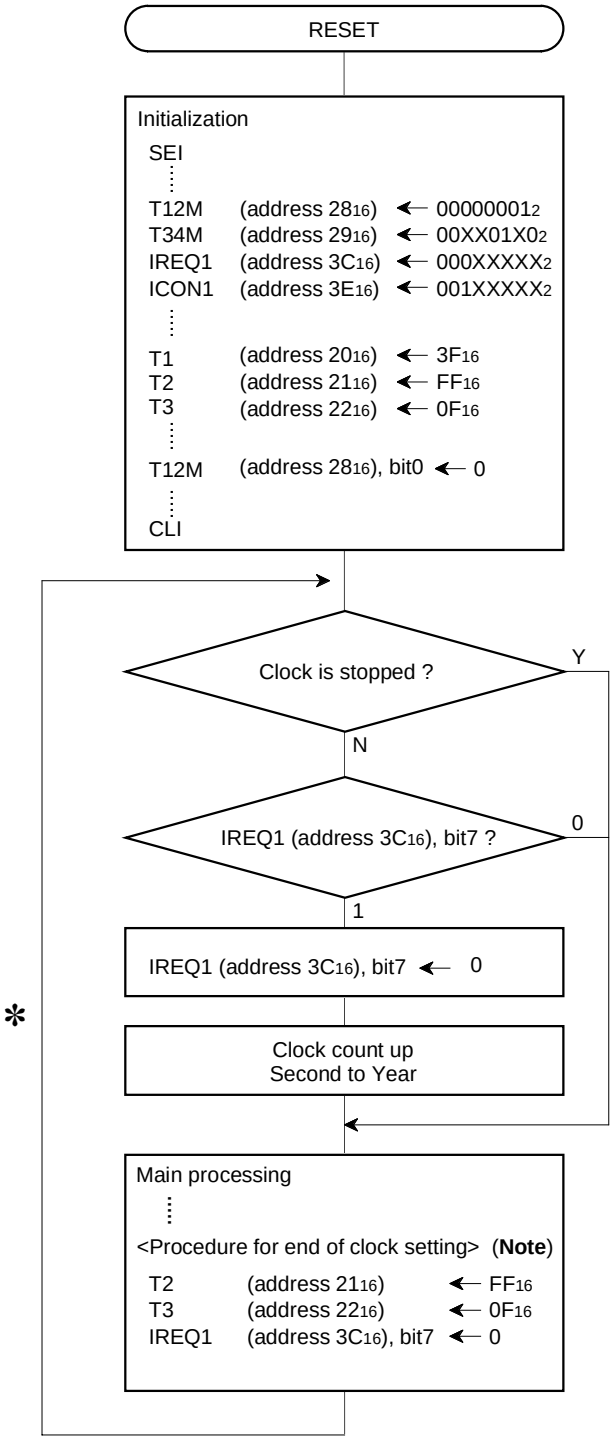


Fig. 2.2.17 Relevant registers setting



● X: This bit is not used here. Set it to “0” or “1” arbitrarily.

•All interrupts disabled

•Connection of Timers 1 to 3

•Setting of Interrupt request bits of Timers 1 to 3 to “0”
•Timer 1 interrupt enabled, Timers 2 and 3 interrupts disabled

•Setting “Division ratio – 1” to Timers 1 to 3

•Timer count start

•Interrupts enabled

•Judgment whether time is not set or time is being set

•Confirmation that 1 sec. has passed
(Check of Timer 3 interrupt request bit)

•Interrupt request bit cleared
(Clear it by software when not using the interrupt.)

•Clock count up

•Adjust the main processing so that all processing in the loop ✻ will be processed within 1 second period.

•Set Timers again when starting clock from 0 second after end of clock setting.
The procedure is Timer 2 setting followed by Timer 3 setting.
•Do not set Timer 1 again because Timer 1 is used to generate the interrupt at 244 μs intervals.

Note : Perform procedure for end of clock setting only when end of clock setting.

Fig. 2.2.18 Control procedure

(3) Timer application example 2: Piezoelectric buzzer output

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

Specifications: •The rectangular waveform, dividing the clock f(X_{IN}) = 4.19 MHz (2^{22} Hz) into about 2 kHz (2048 Hz), is output from the $P4_2$/$T3_{OUT}$ pin.

•The level of the $P4_2$/$T3_{OUT}$ pin is fixed to "H" while a piezoelectric buzzer output stops.

Figure 2.2.19 shows a peripheral circuit example, and Figure 2.2.20 shows the timers connection and setting of division ratios. Figures 2.2.21 shows the relevant registers setting, and Figure 2.2.22 shows the control procedure.

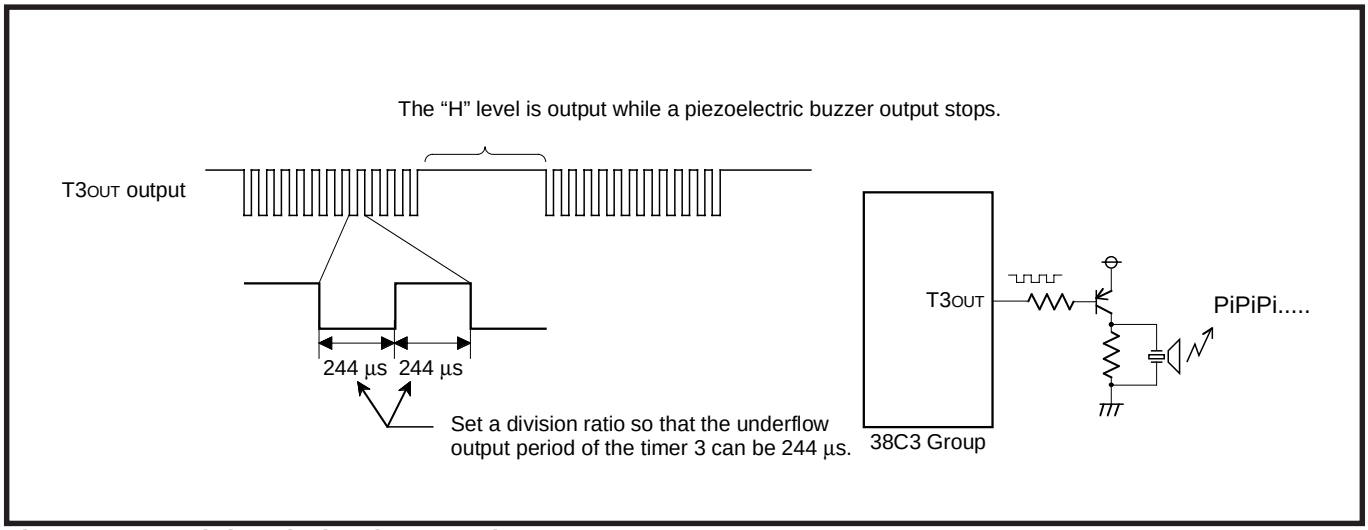


Fig. 2.2.19 Peripheral circuit example

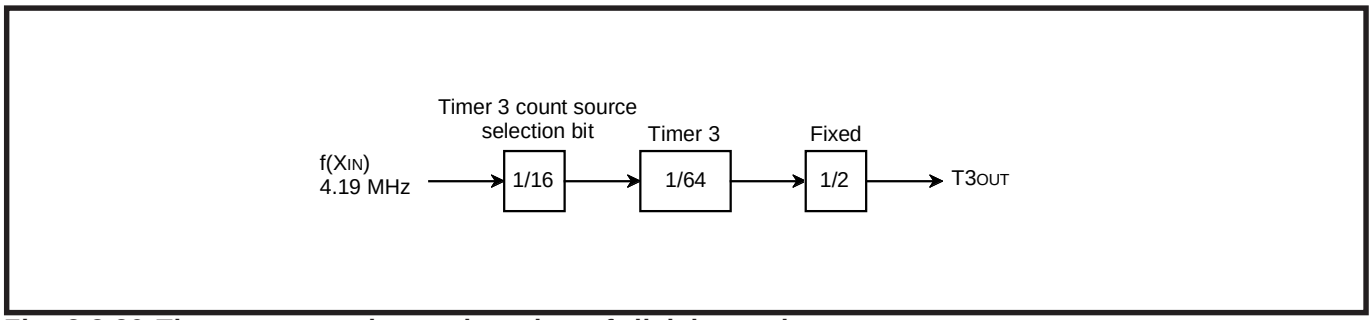


Fig. 2.2.20 Timers connection and setting of division ratios

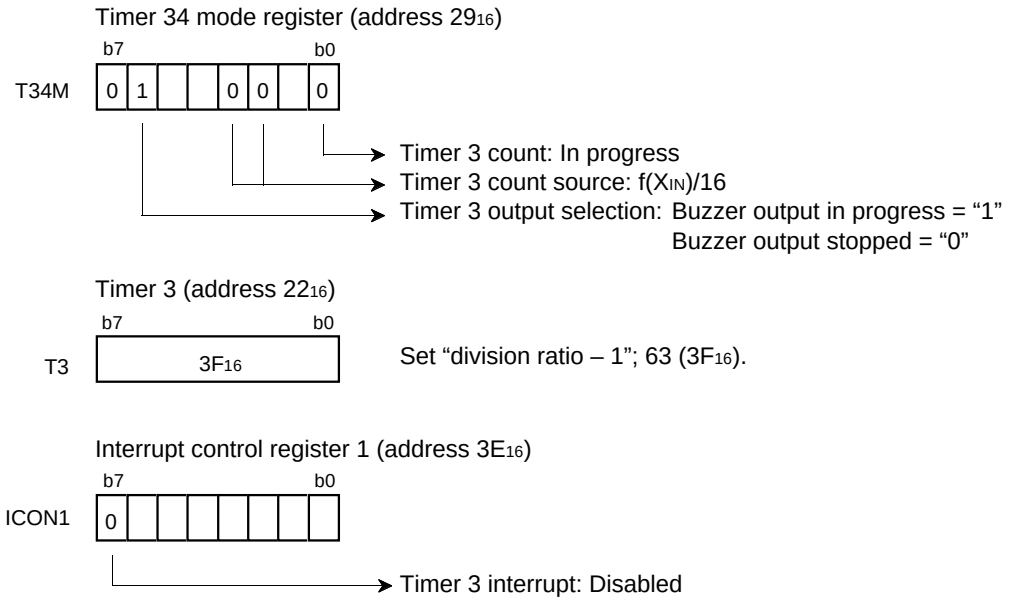


Fig. 2.2.21 Relevant registers setting

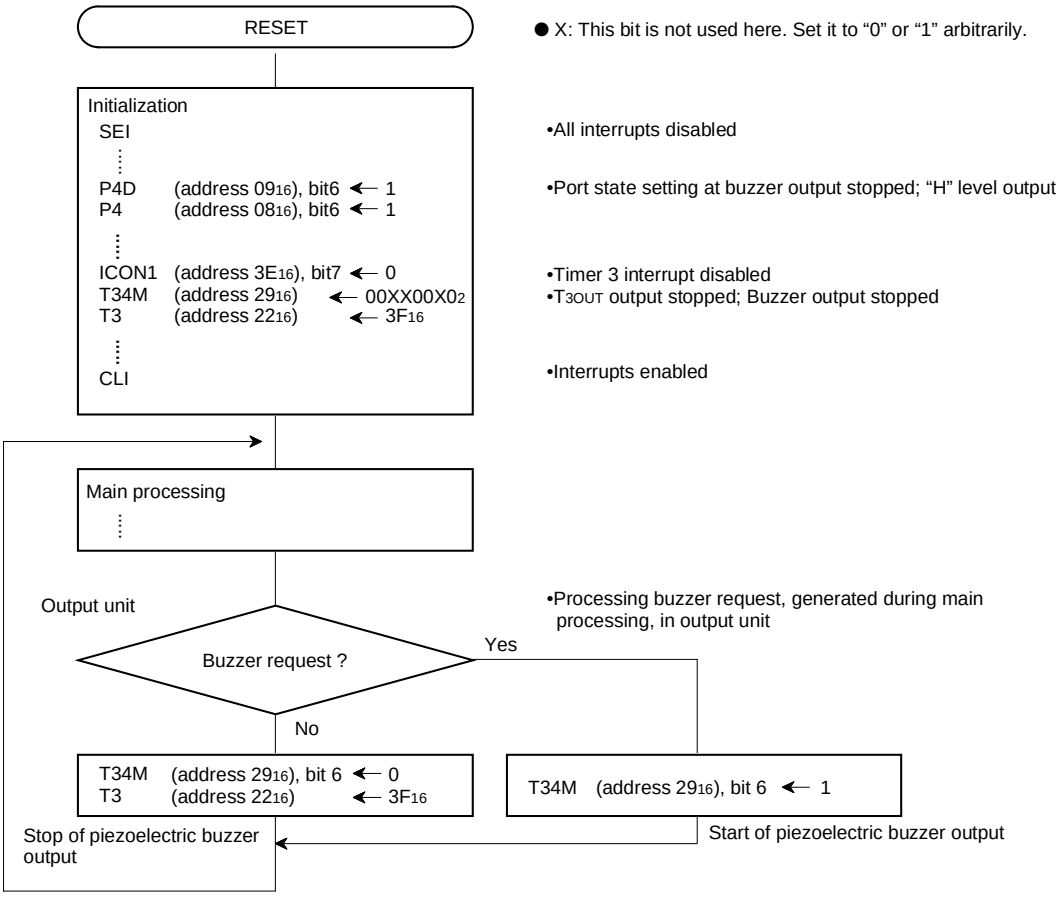


Fig. 2.2.22 Control procedure

(4) Timer application example 3: Frequency measurement

Outline: The following two values are compared to judge whether the frequency is within a valid range.

- A value by counting pulses input to $P5_4/CNTR_1$ pin with the timer.
- A reference value

Specifications: •The pulse is input to the $P5_4/CNTR_1$ pin and counted by the timer 4.

•A count value of timer 4 is read out at about 2 ms intervals, the timer 1 interrupt interval. When the count value is 28 to 40, it is judged that the input pulse is valid.

•Because the timer is a down-counter, the count value is compared with 227 to 215 (**Note**).

Note: 227 to 215 = {255 (initial value of counter) – 28} to {255 – 40}; 28 to 40 means the number of valid count.

Figure 2.2.23 shows the judgment method of valid/invalid of input pulses; Figure 2.2.24 shows the relevant registers setting; Figure 2.2.25 shows the control procedure.

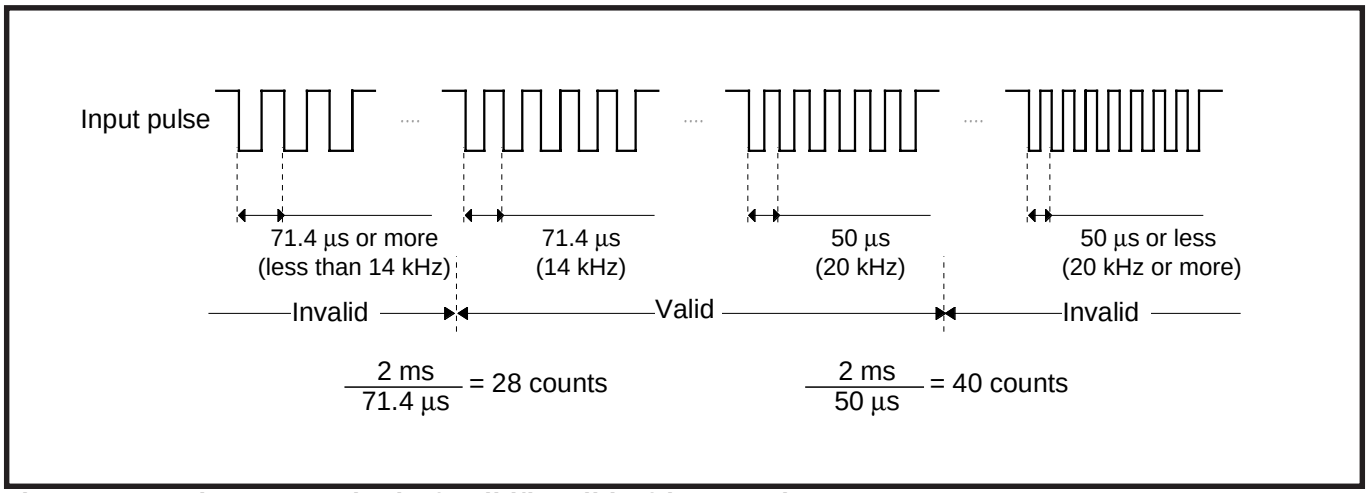


Fig. 2.2.23 Judgment method of valid/invalid of input pulses

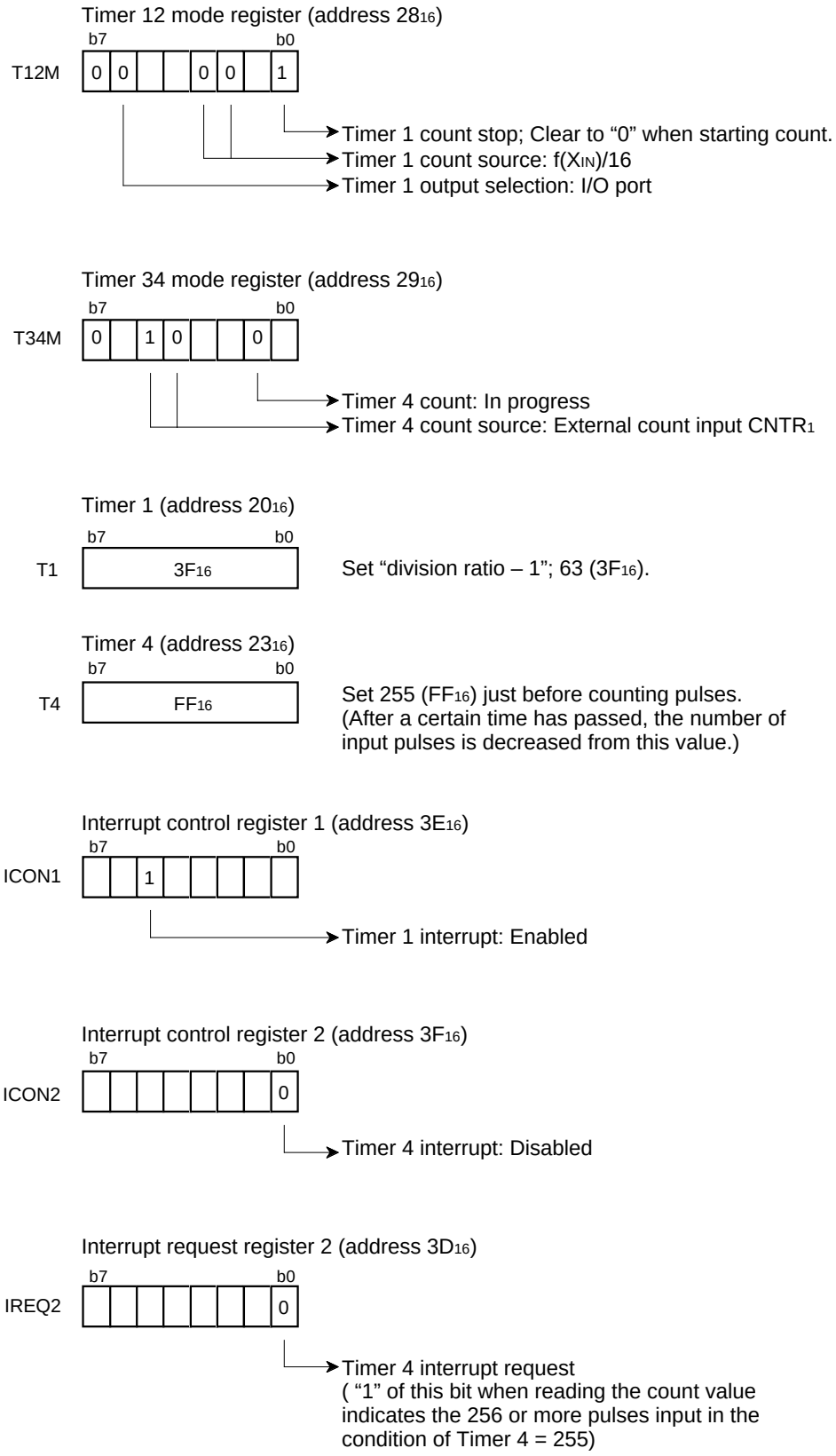


Fig. 2.2.24 Relevant registers setting

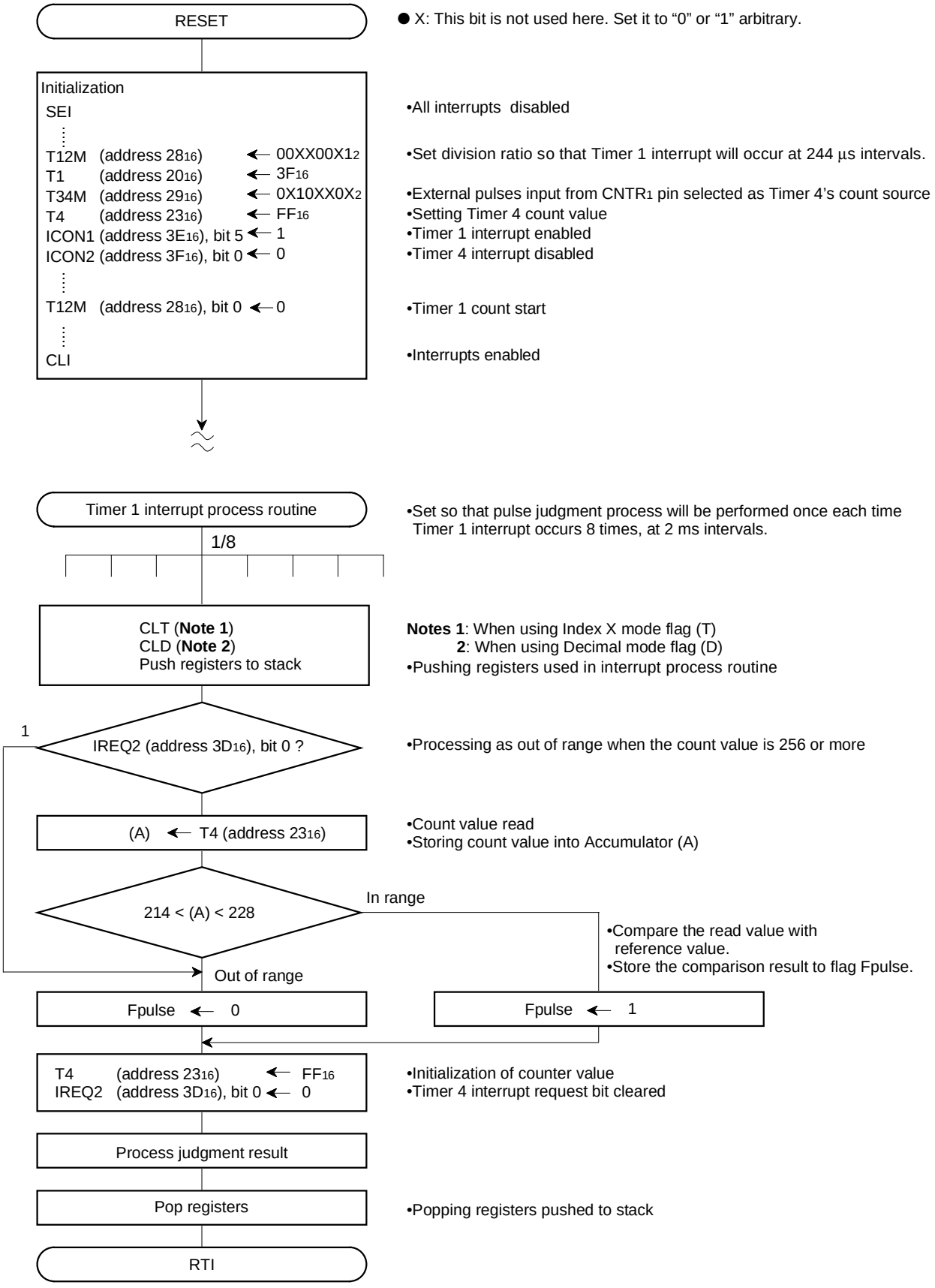


Fig. 2.2.25 Control procedure

(5) Timer application example 4: Output of IGBT control signal

Outline: Synchronized variable PWM signal is output in "H" term.
When a signal is input to INT_0 pin before timer underflow during "L" output, timer restarts.

Specifications: •The signal, of which "H" level width is 5 μs and cycle is 20 μs, is output from the $P5_0/TA_{OUT}$ pin.
However, if "H" is input to INT_0 pin during "L" output, timer restarts from "H" output.

<Example>
When $f(X_{IN})$ = 8 MHz, the count source is 125 ns.

Figure 2.2.26 shows the timers connection and setting of division ratio; Figure 2.2.27 shows the relevant registers setting; Figure 2.2.28 shows the control procedure.

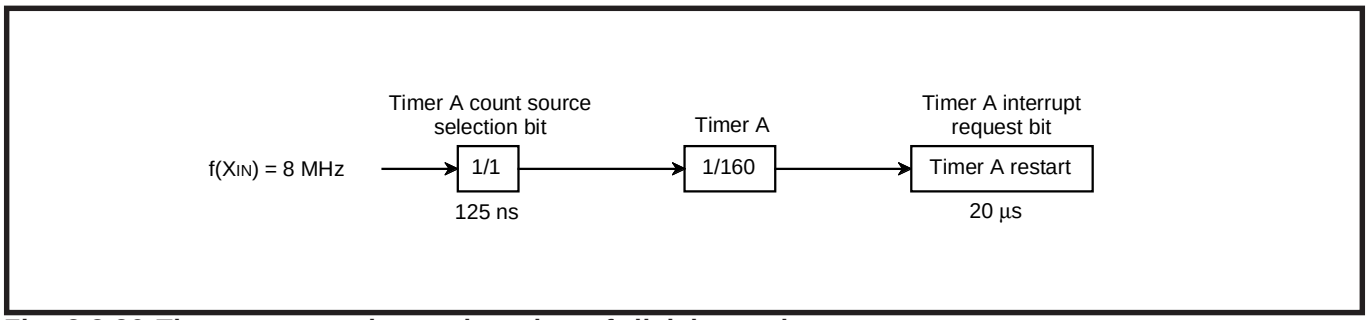
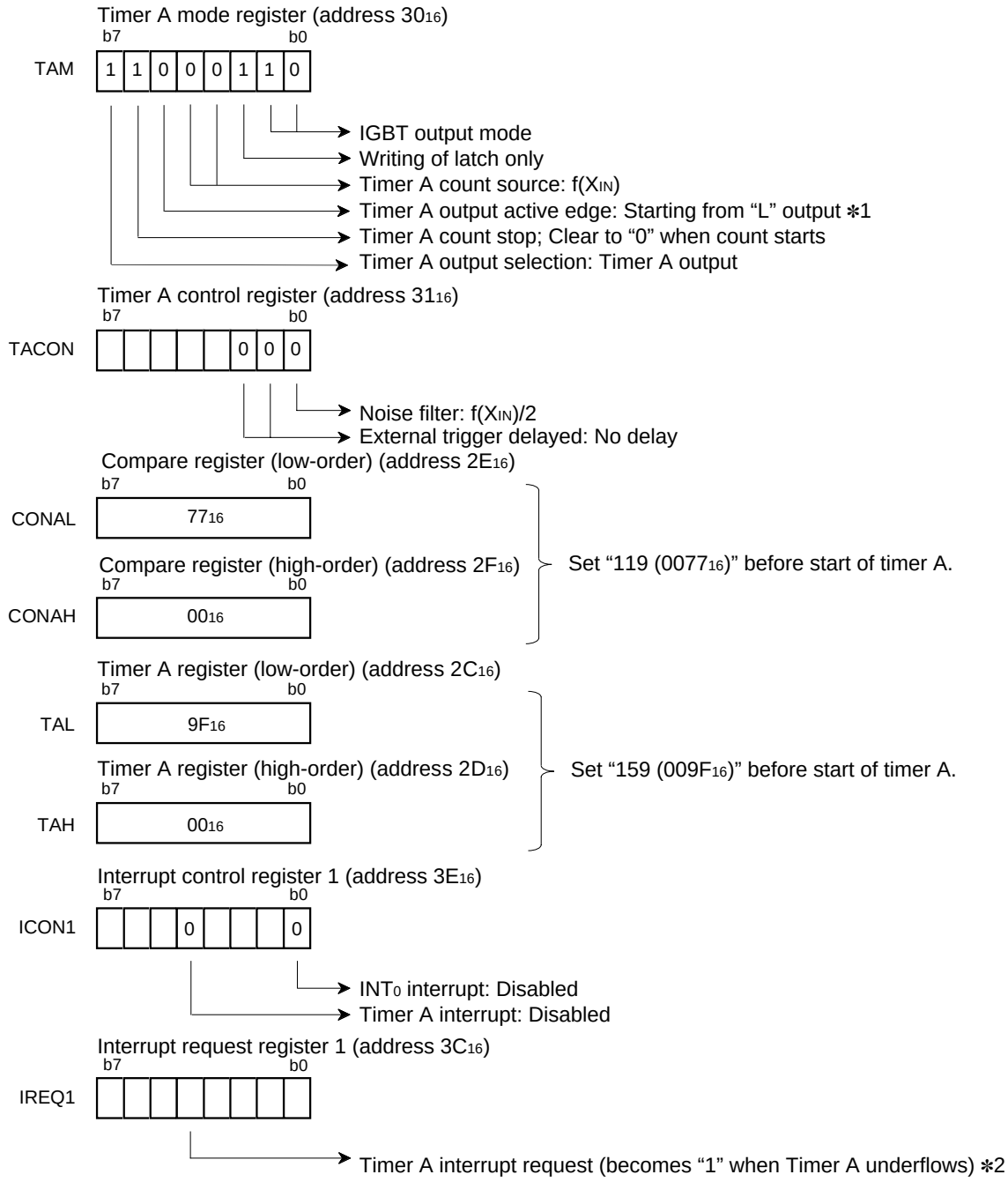


Fig. 2.2.26 Timers connection and setting of division ratios



∗1 Use this bit with "0" (start from "L" output) in the IGBT output mode.
∗2 This bit becomes "1" even when a signal is input from the INT_0 pin in the IGBT output mode.

Fig. 2.2.27 Relevant registers setting

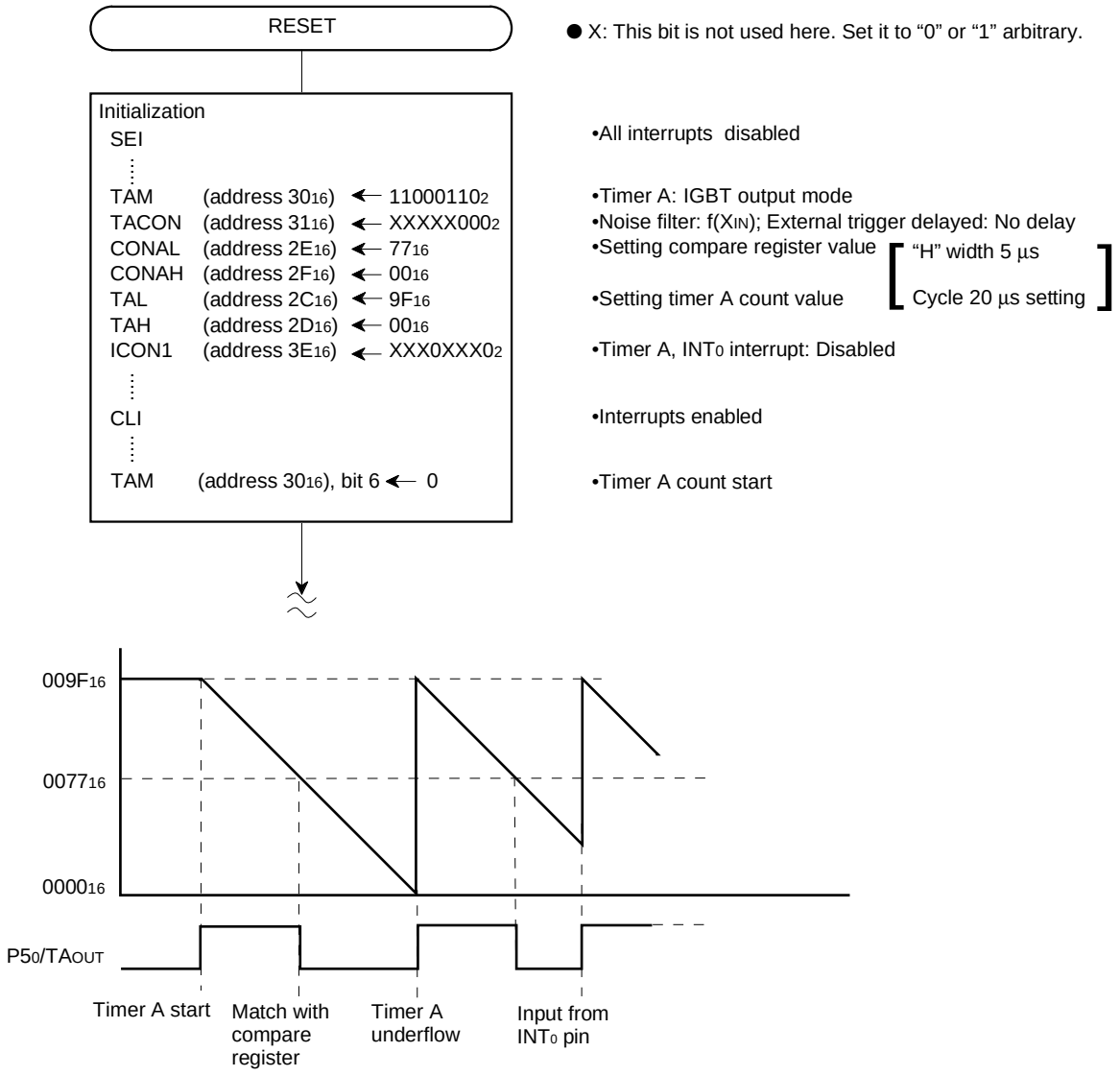


Fig. 2.2.28 Control procedure

2.2.4 Notes on timer A (PWM mode and IGBT output mode)

(1) When timer starts first or last value of compare register is "0000_{16}"

After "L" level (timer A output active edge switch bit is "0"; when starting from "L" output) is output during 2 cycles (until timer underflows two times), PWM output or IGBT output starts.

Reason: When data is written to timer A and compare register, value of timer A and value of compare register are renewed at timer underflow. In case of this, compare register value and timer value are compared before renewal, so that they are judged to be equal, and TA_{OUT} output becomes "L". (Timer A output active edge switch bit = "0": when starting from "L" output)

Timer A underflow should cause "H" output, but the match have the priority. (see **"Figure 2.2.29"**)

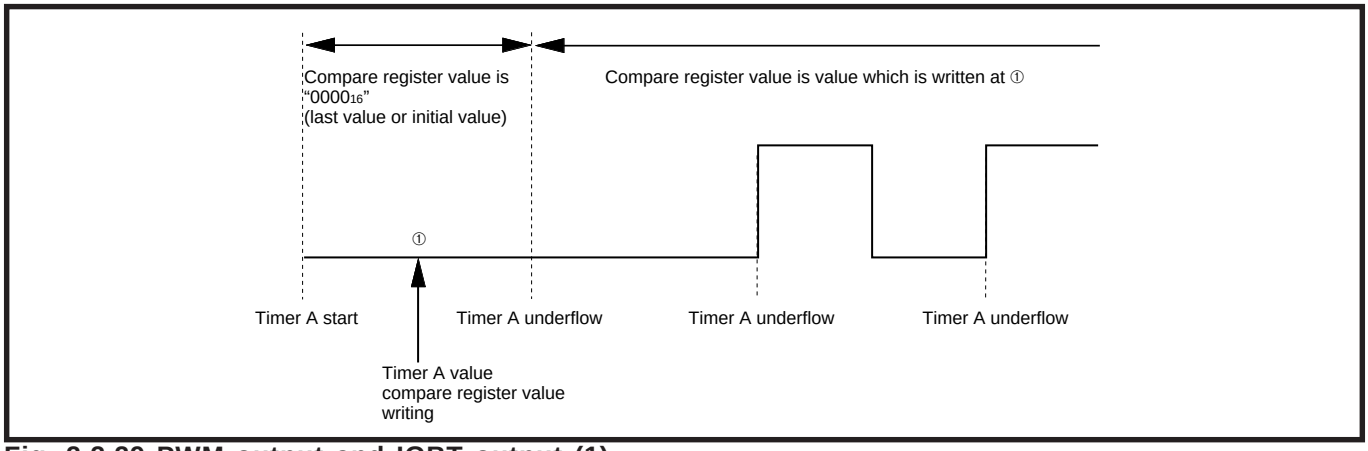


Fig. 2.2.29 PWM output and IGBT output (1)

(2) When compare register is set to "0000_{16}" (last value is except "0000_{16}")

Next 1 cycle of the cycle in which data is written to timer A and compare register is output "H", and "L" is output from the next cycle. (timer A output active edge switch bit = "0": when starting from "L" output)

(see **"Figure 2.2.30"**)

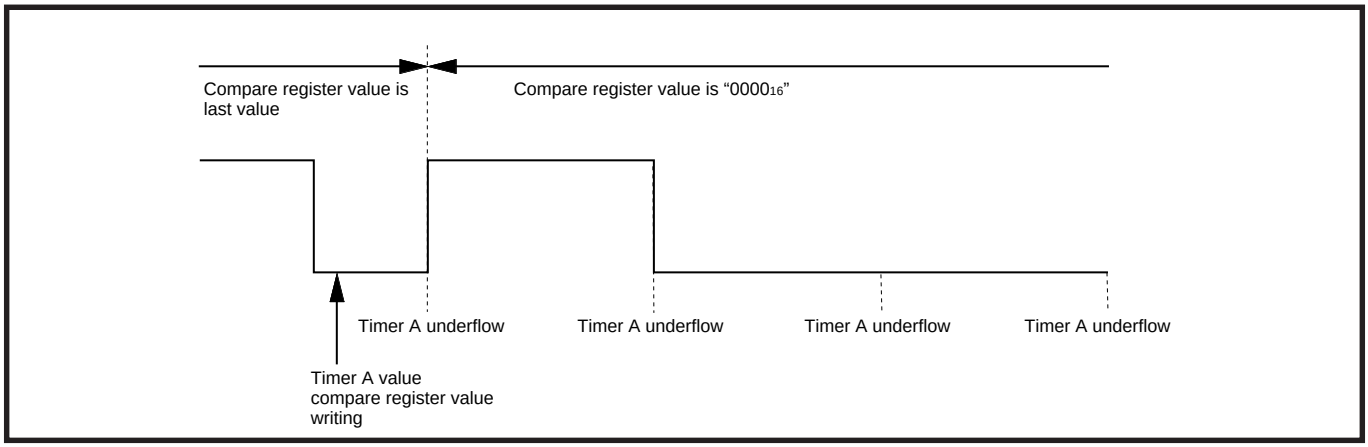


Fig. 2.2.30 PWM output and IGBT output (2)

(3) When timer A and compare register have same value

TA_{OUT} output becomes “H” with underflow immediately after data is written to timer A and compare register. TA_{OUT} output becomes “L” when timer A is reloaded and the value matches with compare register. This “H” output width becomes 1 count of timer A count source. (timer A output active edge switch bit =“0”: when starting from “L” output) (see **“Figure 2.2.31”**)

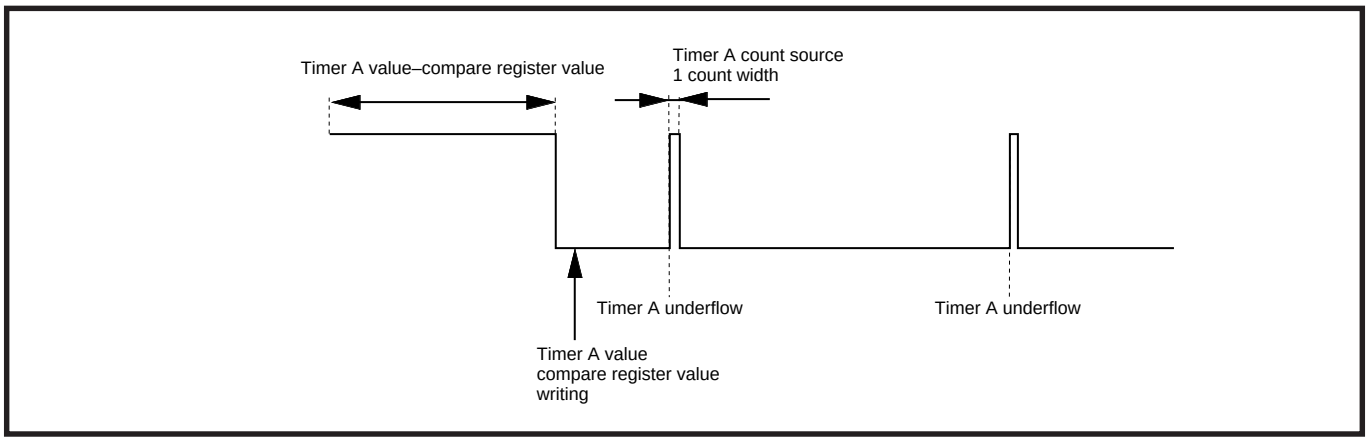


Fig. 2.2.31 PWM output and IGBT output (3)

2.3 Serial I/O

This paragraph explains the registers setting method and the notes relevant to the serial I/O.

2.3.1 Memory map

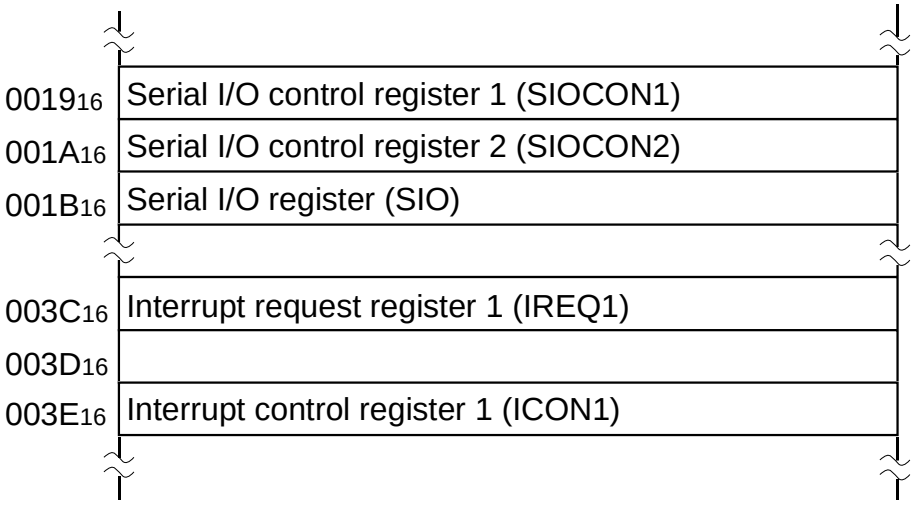


Fig. 2.3.1 Memory map of registers relevant to Serial I/O

2.3.2 Relevant registers

Serial I/O control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O control register 1
(SIOCON1: address 19₁₆)

b	Name	Functions	At reset	R	W
0	Internal synchronous clock selection bits	b2b1b0 0 0 0: f(XIN)/8 or f(XCIN)/8 0 0 1: f(XIN)/16 or f(XCIN)/16 0 1 0: f(XIN)/32 or f(XCIN)/32 0 1 1: f(XIN)/64 or f(XCIN)/64 1 1 0: f(XIN)/128 or f(XCIN)/128 1 1 1: f(XIN)/256 or f(XCIN)/256	0	○	○
1			0	○	○
2			0	○	○
3	Serial I/O port selection bit (P4₀, P4₅, P4₆)	0: I/O port 1: SOUT, SCLK1, SCLK2 signal pin	0	○	○
4	$\overline{S_{RDY}}$ output selection bit (P4₇)	0: I/O port 1: $\overline{S_{RDY}}$ signal pin	0	○	○
5	Transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
6	Synchronous clock selection bit	0: External clock 1: Internal clock	0	○	○
7	P-channel output disable bit (P4₀, P4₅, P4₆)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○

Fig. 2.3.2 Structure of Serial I/O control register 1

Serial I/O control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O control register 2
(SIOCON2: address $1A_{16}$)

b	Name	Functions	At reset	R	W
0	Synchronous clock output pin selection bit	0: S_{CLK1} 1: S_{CLK2}	0	○	○
1	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
2			0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.3.3 Structure of Serial I/O control register 2

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

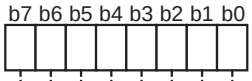
Interrupt request register 1
(IREQ1 : address $3C_{16}$)

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	INT_2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer A interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer 1 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. 2.3.4 Structure of Interrupt request register 1

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	INT_2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer A interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 1 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	○	○

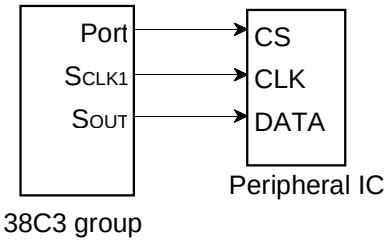
Fig. 2.3.5 Structure of Interrupt control register 1

2.3.3 Serial I/O connection examples

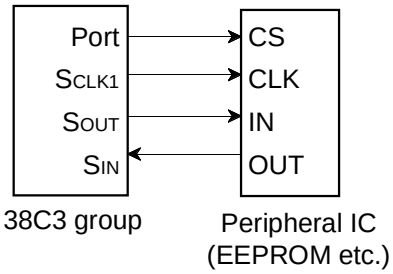
(1) Control of peripheral IC equipped with CS pin

Figure 2.3.6 shows connection examples with peripheral ICs equipped with the CS pin.

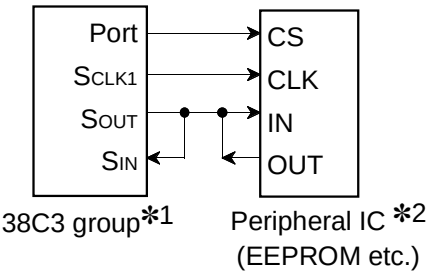
(1) Only transmission
(Using S_{IN} pin as I/O port)



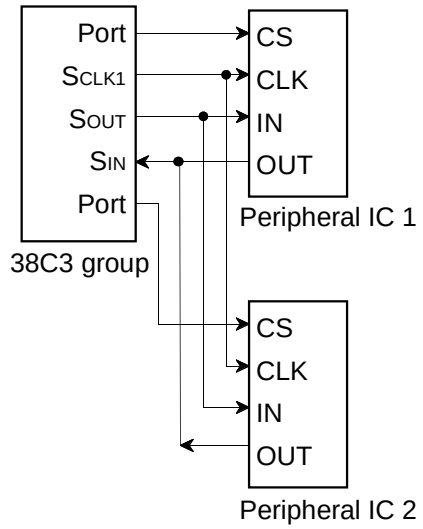
(2) Transmission and reception



(3) Transmission and reception
(When connecting S_{IN1} with S_{OUT1})
(When connecting IN with OUT in peripheral IC)



(4) Connection of plural IC



*1: Select an N-channel open-drain output for S_{OUT} pin output control.

*2: Use the OUT pin of peripheral IC which is an N-channel open-drain output and becomes high impedance during receiving data.

Note: "Port" means an output port controlled by software.

Fig. 2.3.6 Serial I/O connection examples (1)

(2) Connection with microcomputer

Figure 2.3.7 shows connection examples with another microcomputer.

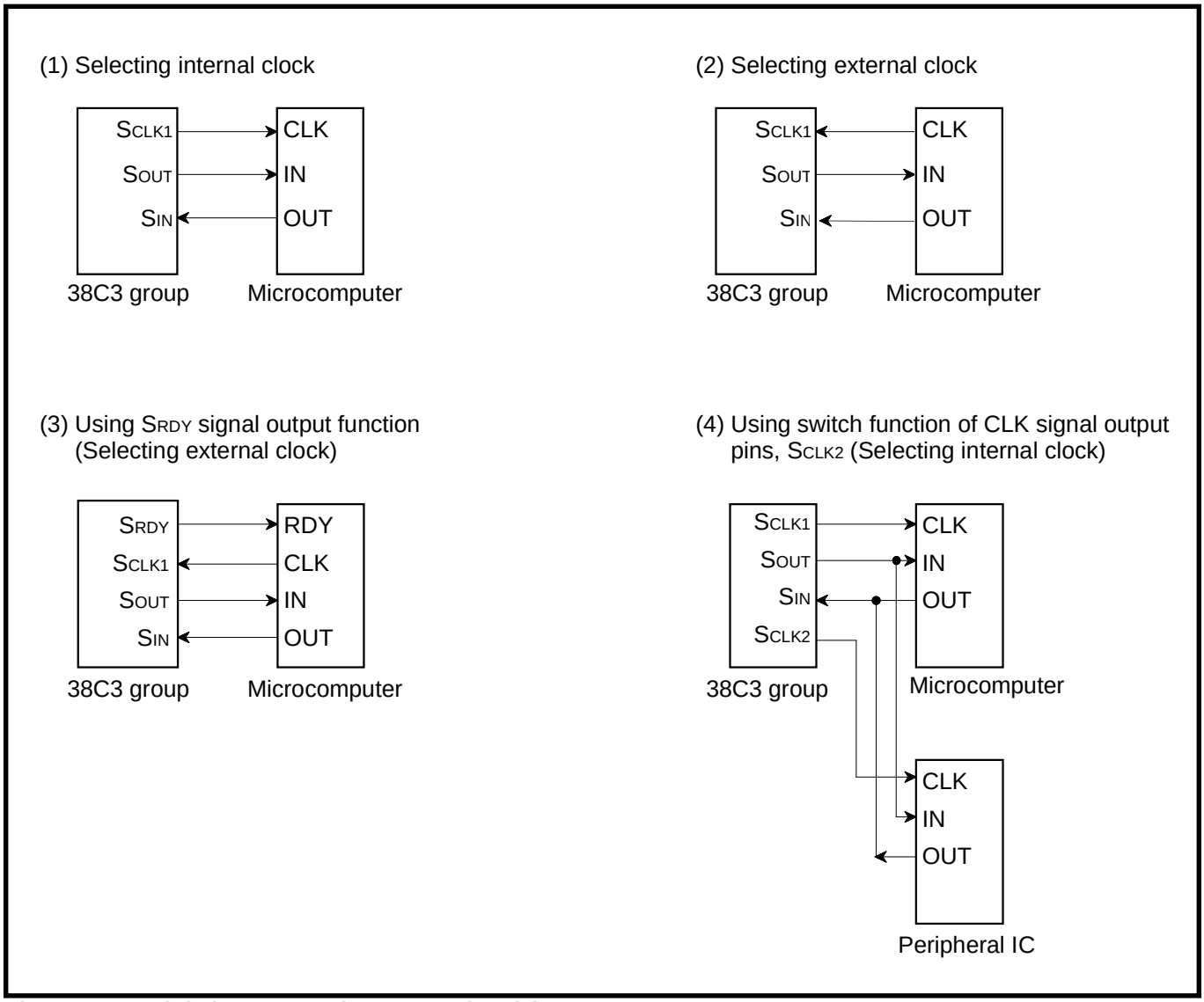


Fig. 2.3.7 Serial I/O connection examples (2)

2.3.4 Serial I/O's modes

38C3 Group can use clock synchronous serial I/O.
Figure 2.3.8 shows the serial I/O's modes.

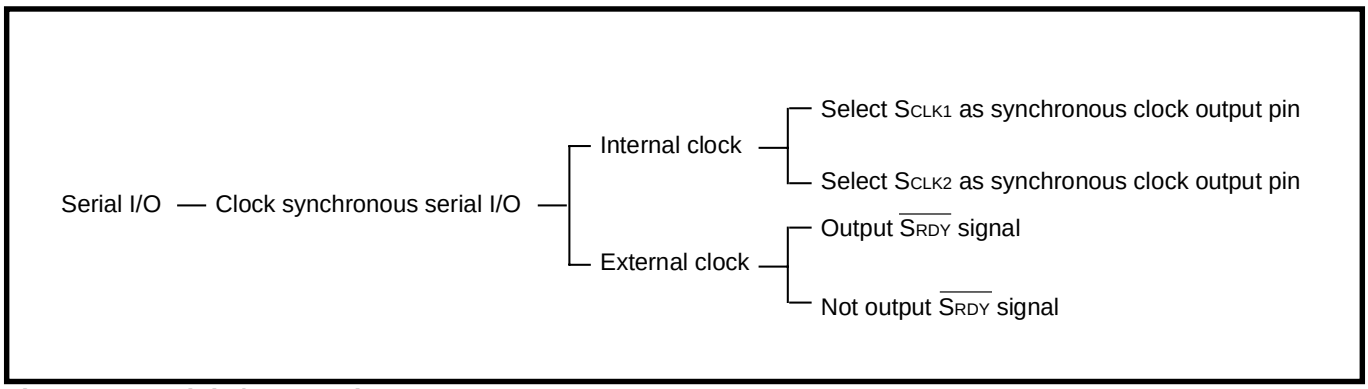


Fig. 2.3.8 Serial I/O's modes

2.3.5 Serial I/O application examples

(1) Communication (transmission/reception) using clock synchronous serial I/O

Outline : 2-byte data is transmitted and received, using the clock synchronous serial I/O. The SRDY signal is used for communication control.

Figure 2.3.9 shows a connection diagram, and Figure 2.3.10 shows a timing chart.

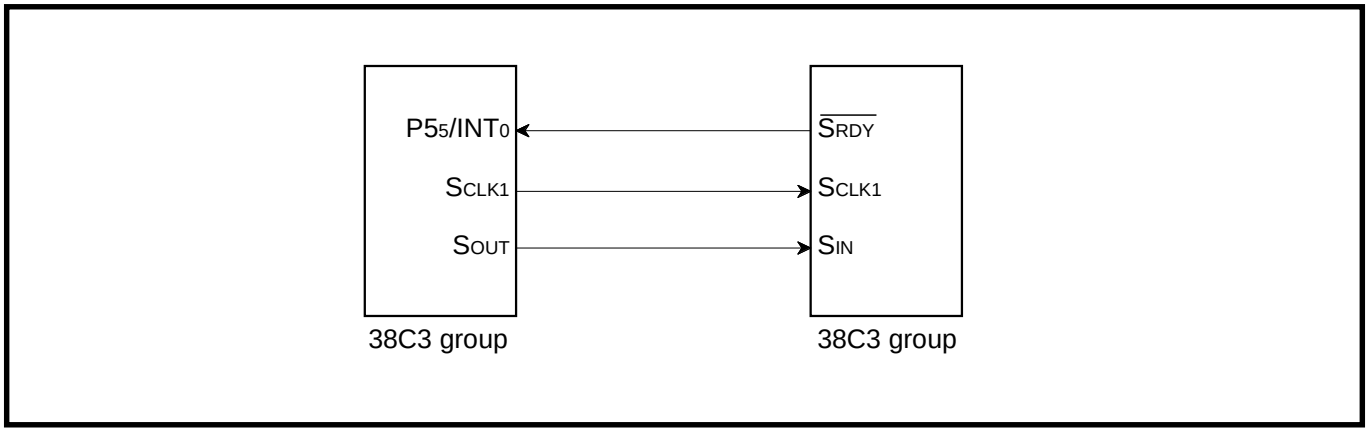


Fig. 2.3.9 Connection diagram

Specifications :
- Use of serial I/O in clock synchronous serial I/O
- Synchronous clock frequency : 125 kHz (f(XIN) = 4 MHz is divided by 32)
- Use of SRDY (receivable signal)
- The reception side outputs the SRDY signal at intervals of 2 ms (generated by the timer), and 2-byte data is transferred from the transmission side to the reception side.

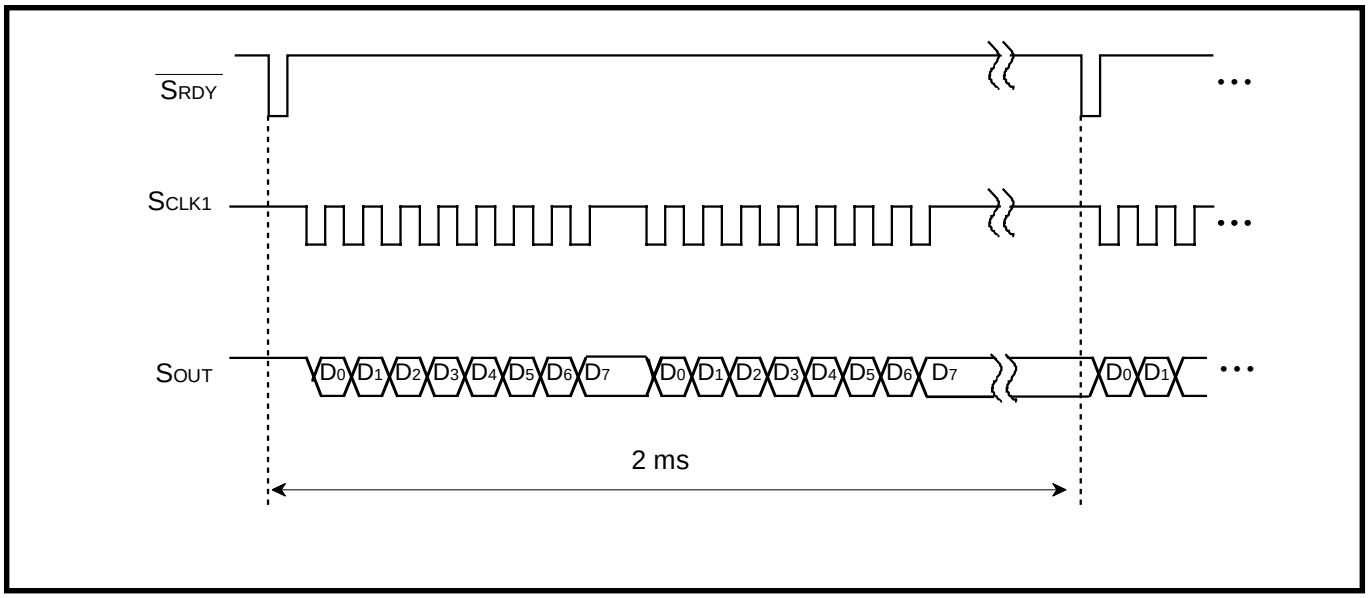


Fig. 2.3.10 Timing chart

Figure 2.3.11 shows the registers setting relevant to the transmission side, and Figure 2.3.12 shows the registers setting relevant to the reception side.

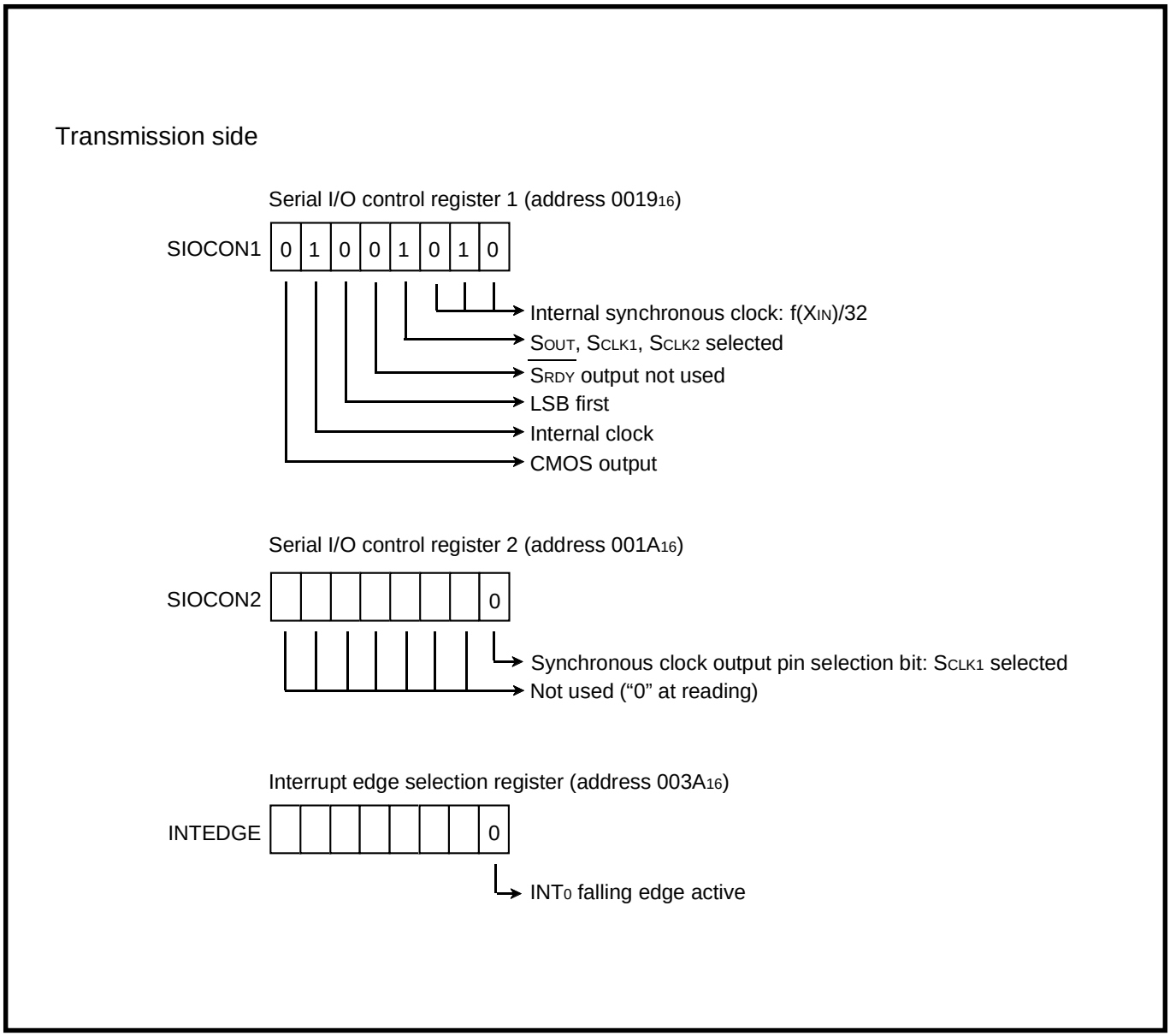


Fig. 2.3.11 Registers setting relevant to transmission side

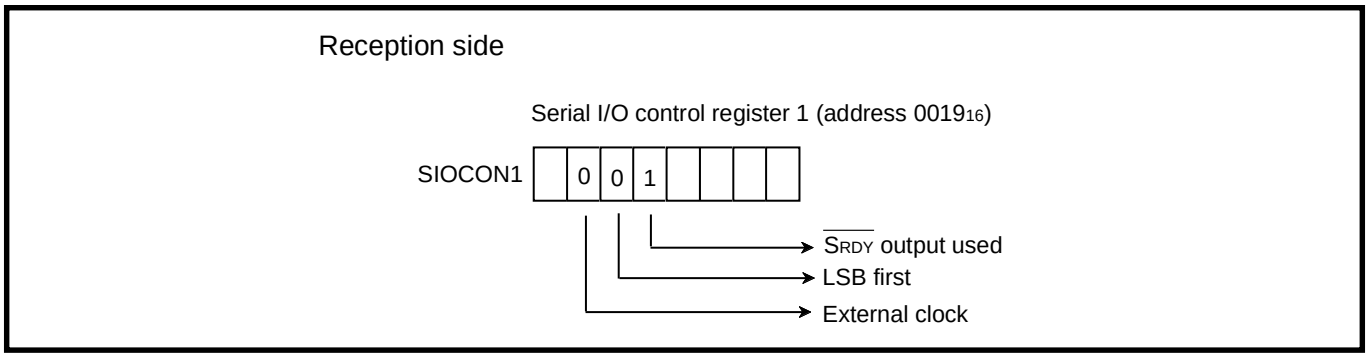


Fig. 2.3.12 Registers setting relevant to reception side

Figure 2.3.13 shows a control procedure of the transmission side, and Figure 2.3.14 shows a control procedure of the reception side.

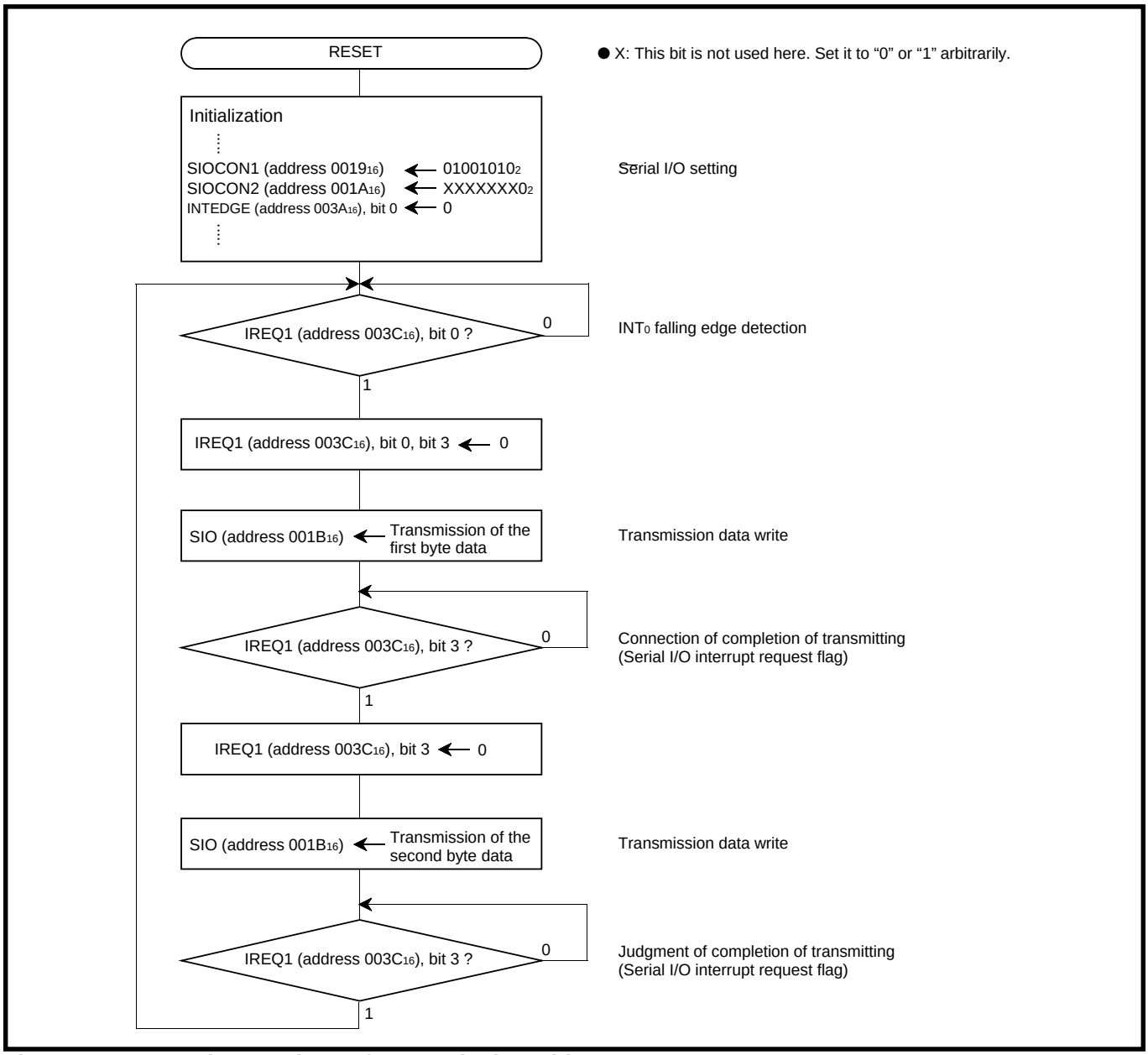


Fig. 2.3.13 Control procedure of transmission side

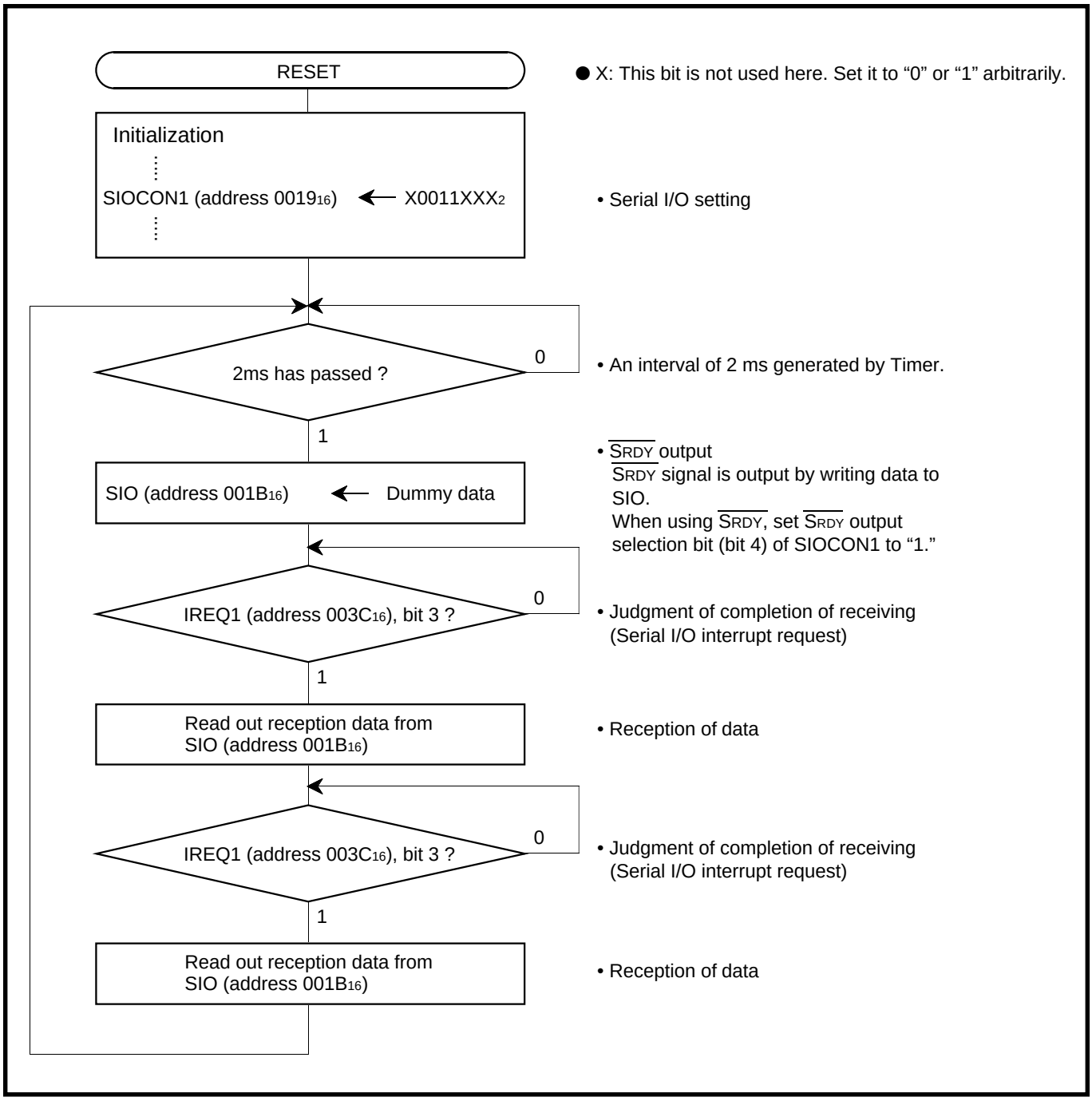


Fig. 2.3.14 Control procedure of reception side

(2) Output of serial data (control of peripheral IC)

Outline : Serial communication is performed, connecting port $P5_7$ with the $\overline{CS}$ pin of a peripheral IC.

Figure 2.3.15 shows a connection diagram, and Figure 2.3.16 shows a timing chart.

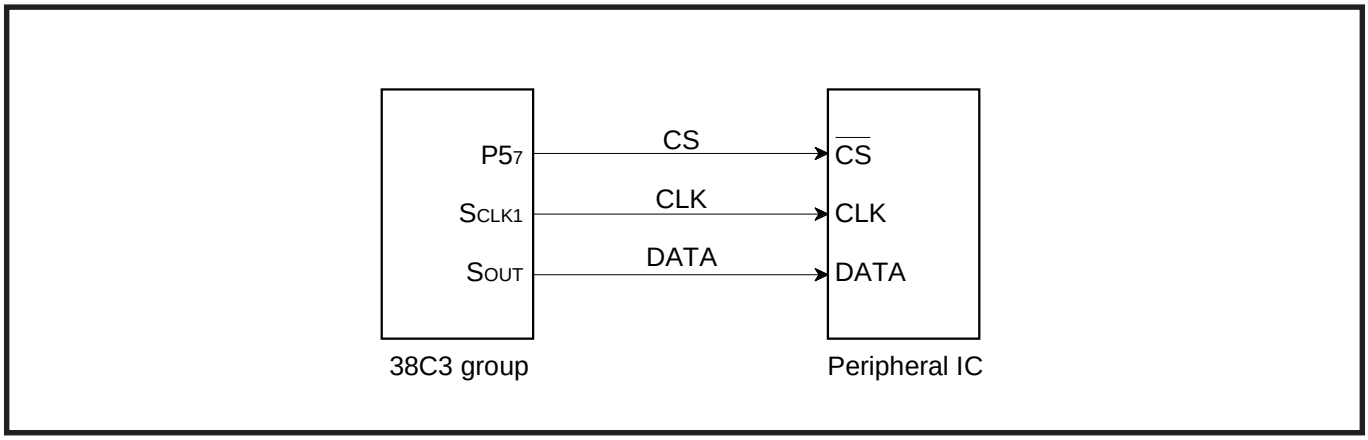


Fig. 2.3.15 Connection diagram

Specifications :
- Use of serial I/O in clock synchronous serial I/O
- Synchronous clock frequency : 125 kHz ($f(X_{IN})$ = 4 MHz is divided by 32)
- Transfer direction : LSB first
- Not use of serial I/O interrupt
- Port $P5_7$ is connected with the $\overline{CS}$ pin ("L" active) of the peripheral IC for transmission control; the output level of port $P5_7$ is controlled by software.

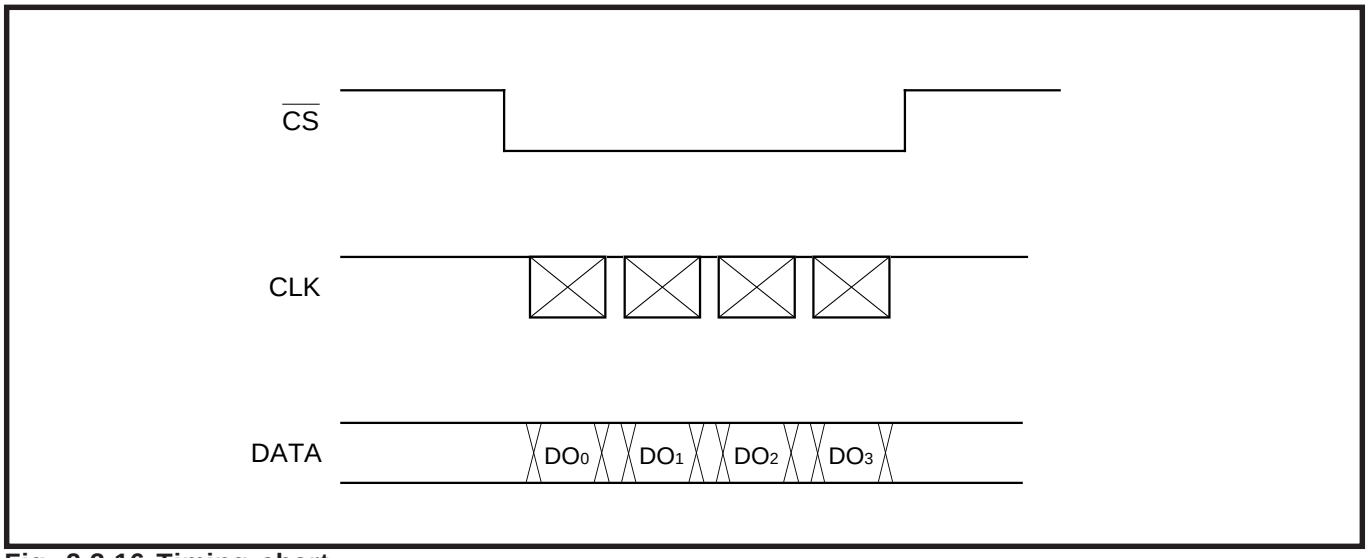


Fig. 2.3.16 Timing chart

2.3 Serial I/O

Figure 2.3.17 shows the relevant registers setting and Figure 2.3.18 shows the setting of transmission data.

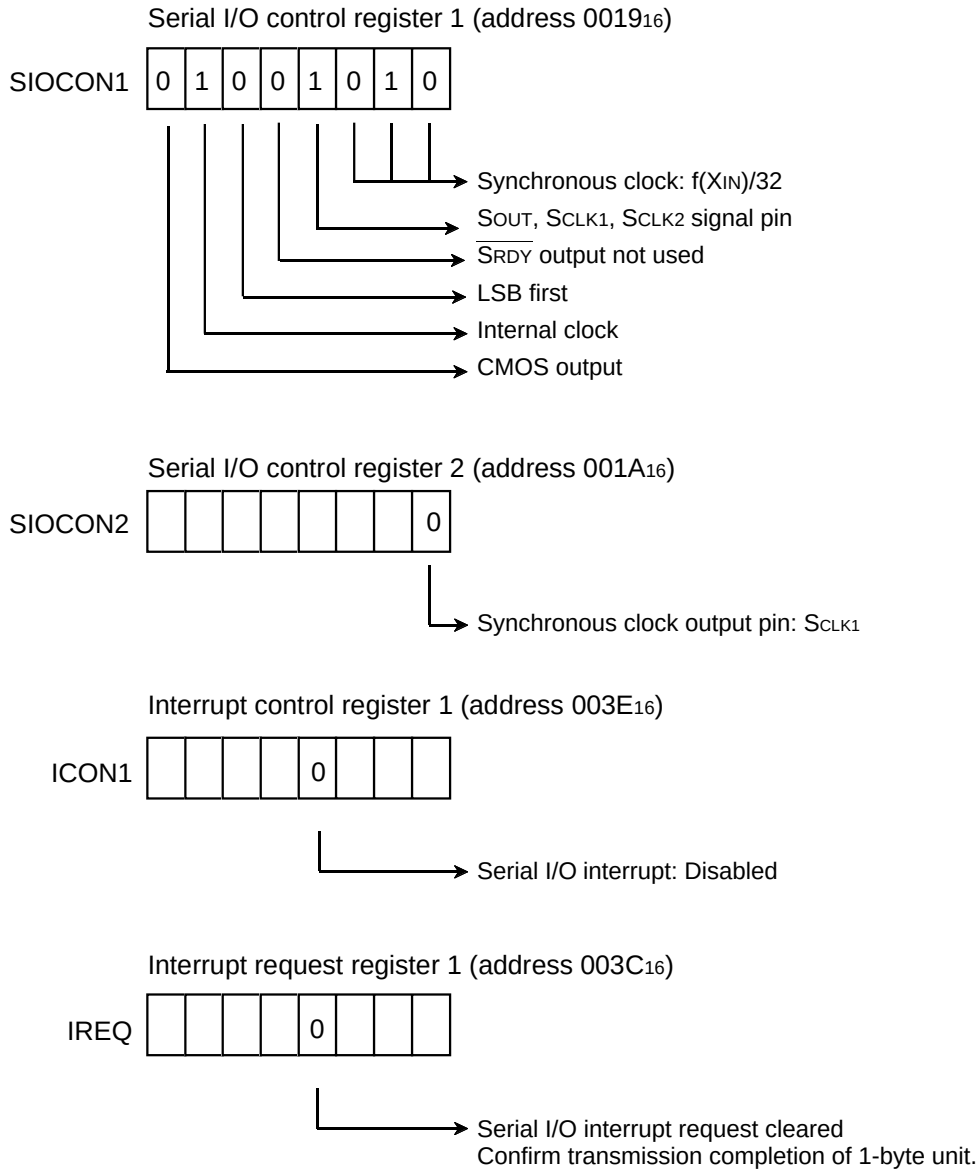


Fig. 2.3.17 Relevant registers setting

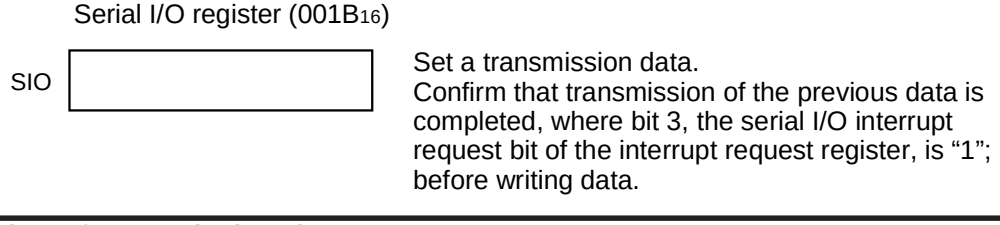


Fig. 2.3.18 Setting of transmission data

Figure 2.3.19 shows a control procedure.

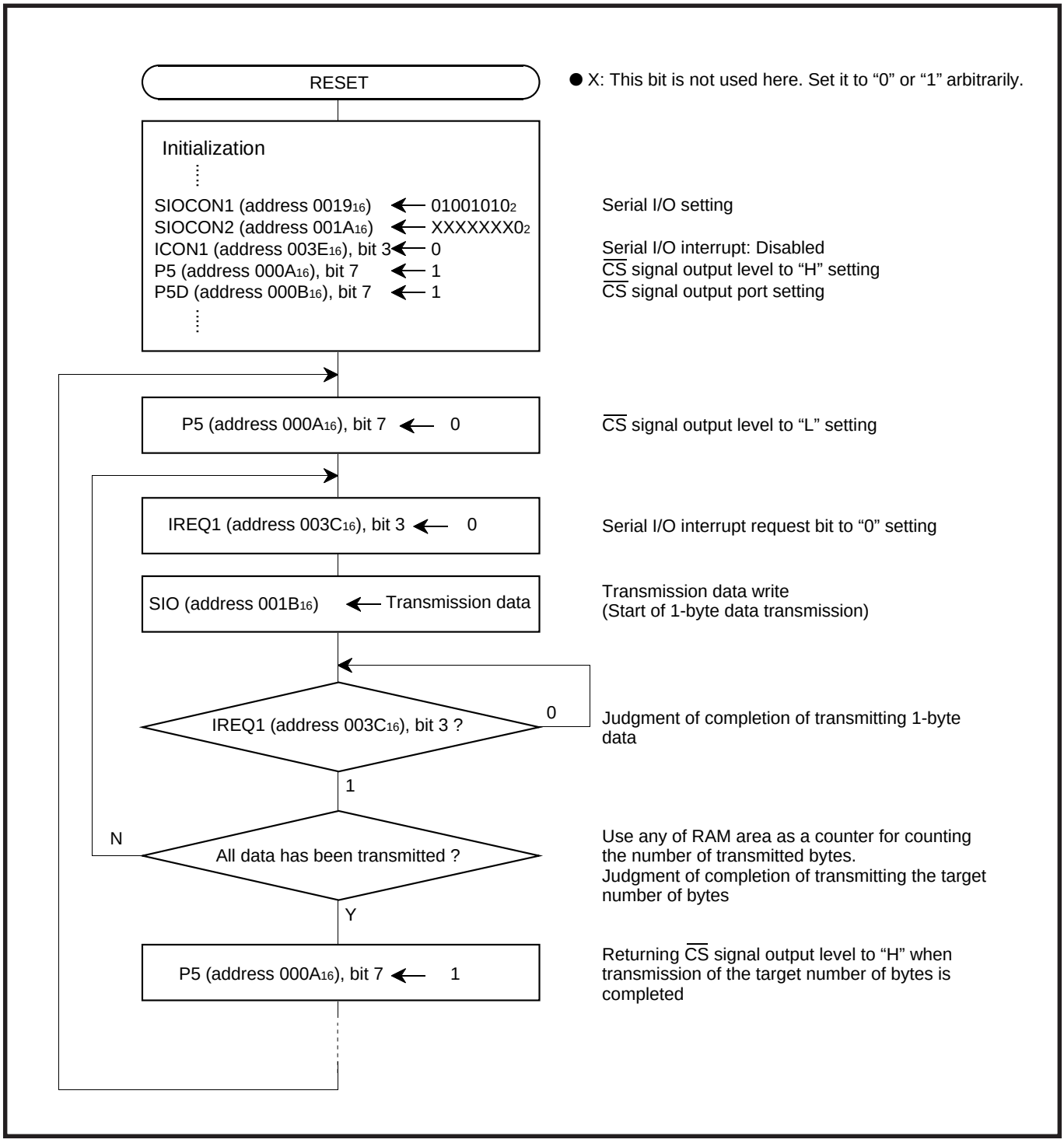


Fig. 2.3.19 Control procedure

2.3 Serial I/O

(3) Cyclic transmission or reception of block data (data of specified number of bytes) between two microcomputers

Outline : When the clock synchronous serial I/O is used for communication, synchronization of the clock and the data between the transmitting and receiving sides may be lost because of noise included in the synchronous clock. It is necessary to correct that constantly, using "heading adjustment".

This "heading adjustment" is carried out by using the interval between blocks in this example.

Figure 2.3.20 shows a connection diagram.

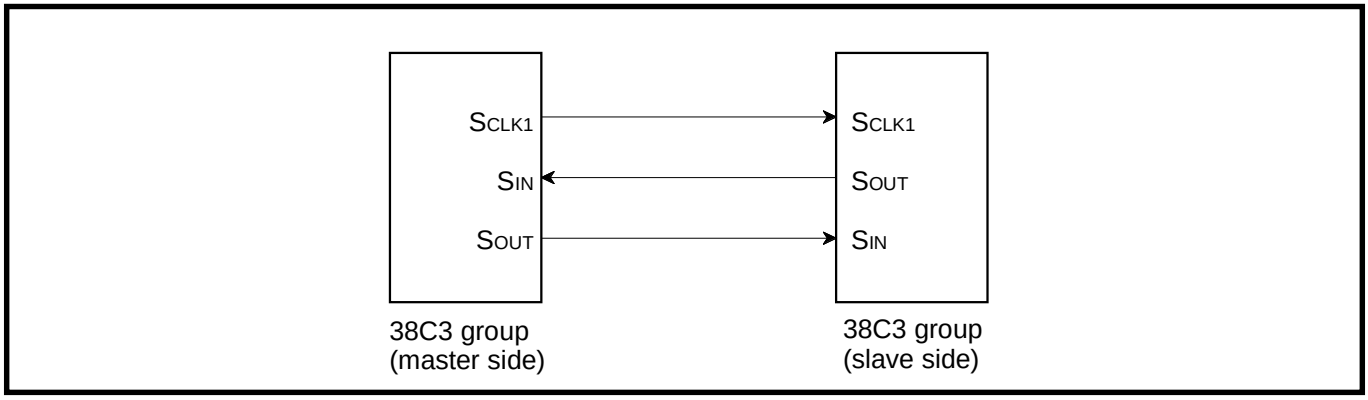


Fig. 2.3.20 Connection diagram

Specifications:
- Synchronous clock frequency : 131 kHz (f(X_{IN}) = 4.19 MHz is divided by 32.)
- Byte cycle: 488 μs
- Number of bytes for transmission or reception : 8 bytes/block each
- Block transfer cycle : 16 ms
- Block transfer term : 3.5 ms
- Interval between blocks : 12.5 ms
- Heading adjustment time : 8 ms
- Transfer direction : LSB first

Limitations of the specifications:
- Reading of the reception data and setting of the next transmission data must be completed within the time obtained from "byte cycle – time for transferring 1-byte data" (in this example, the time taken from generating of the serial I/O interrupt request to input of the next synchronous clock is 431 μs).
- "Heading adjustment time < interval between blocks" must be satisfied.

The communication is performed according to the timing shown in Figure 2.3.21. In the slave unit, when a synchronous clock is not input within a certain time (heading adjusment time), the next clock input is processed as the beginning (heading) of a block.
When a clock is input again after one block (8 bytes) is received, the clock is ignored.
Figure 2.3.22 shows the relevant registers setting in the master unit and Figure 2.3.23 shows the relevant registers setting in the slave unit.

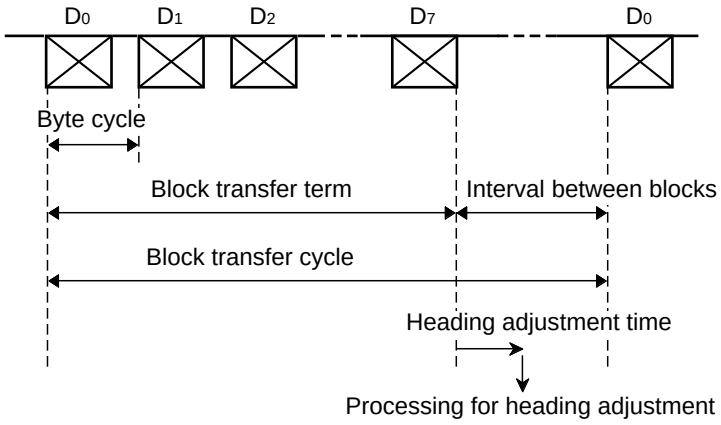


Fig. 2.3.21 Timing chart

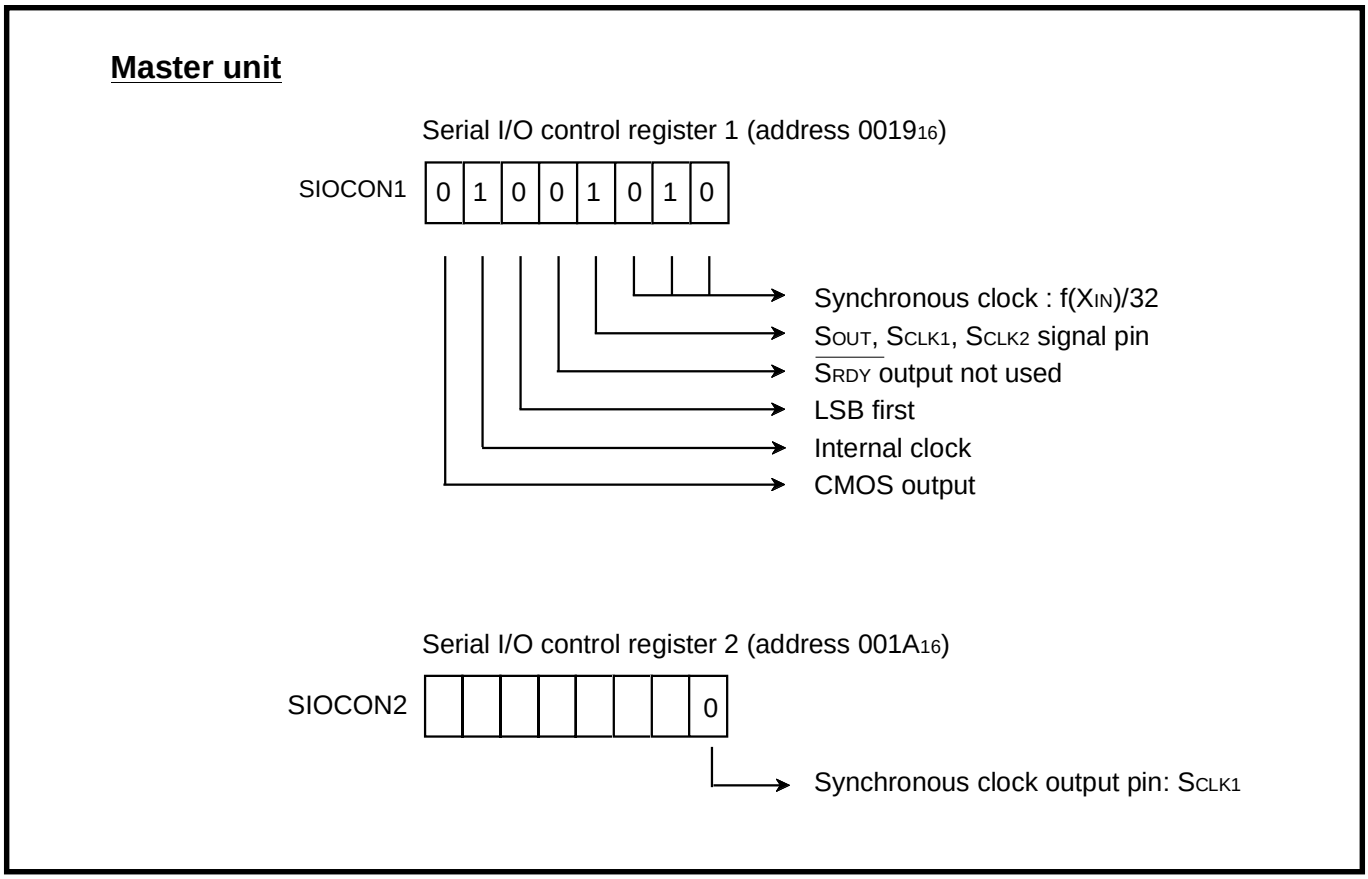


Fig. 2.3.22 Relevant registers setting in master unit

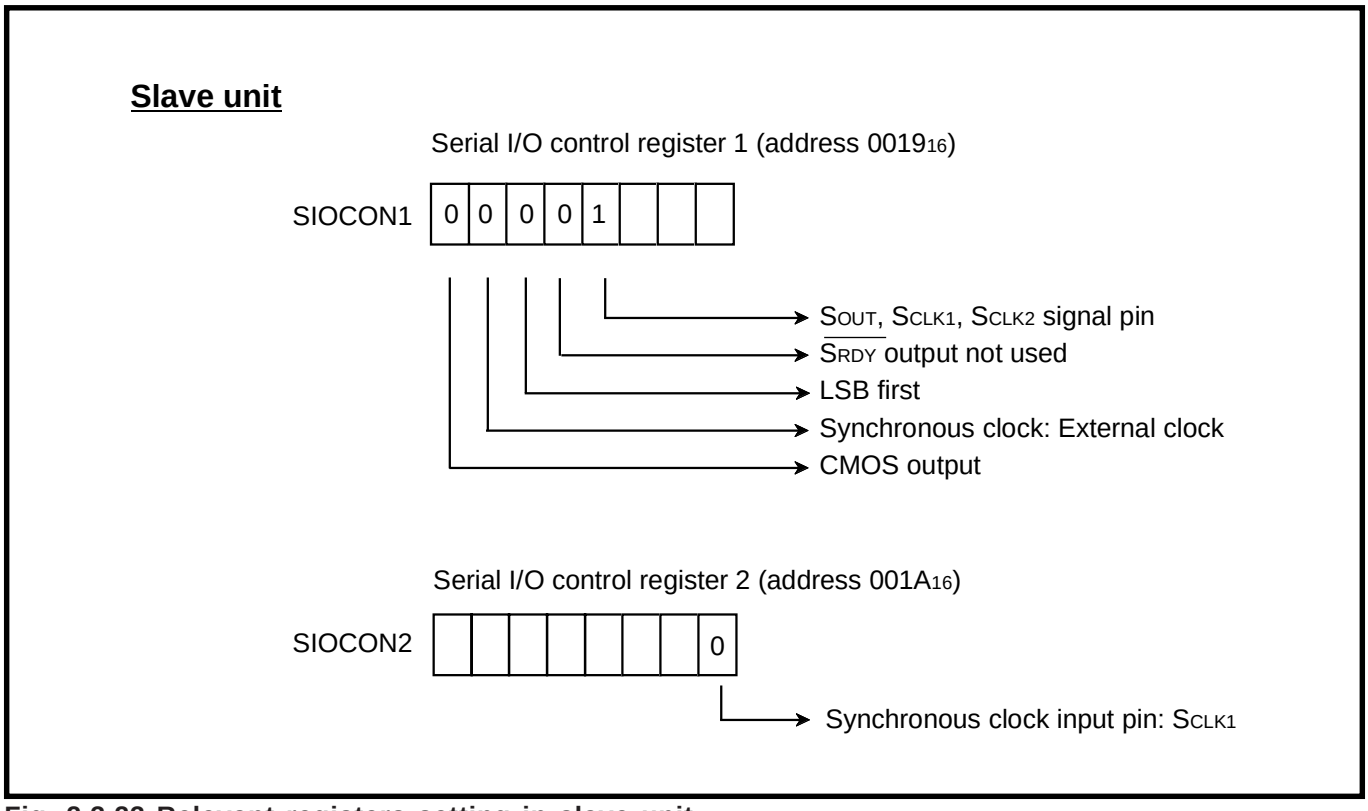


Fig. 2.3.23 Relevant registers setting in slave unit

Control procedure by software:

● Control in the master unit

After setting the relevant registers shown in Figure 2.3.22, the master unit starts transmission or reception of 1-byte data by writing transmission data to the serial I/O register.

To perform the communication in the timing shown in Figure 2.3.21, take the timing into account and write transmission data. Additionally, read out the reception data when the serial I/O interrupt request bit is set to "1," or before the next transmission data is written to the serial I/O register. Figure 2.3.24 shows a control procedure of the master unit using timer interrupts.

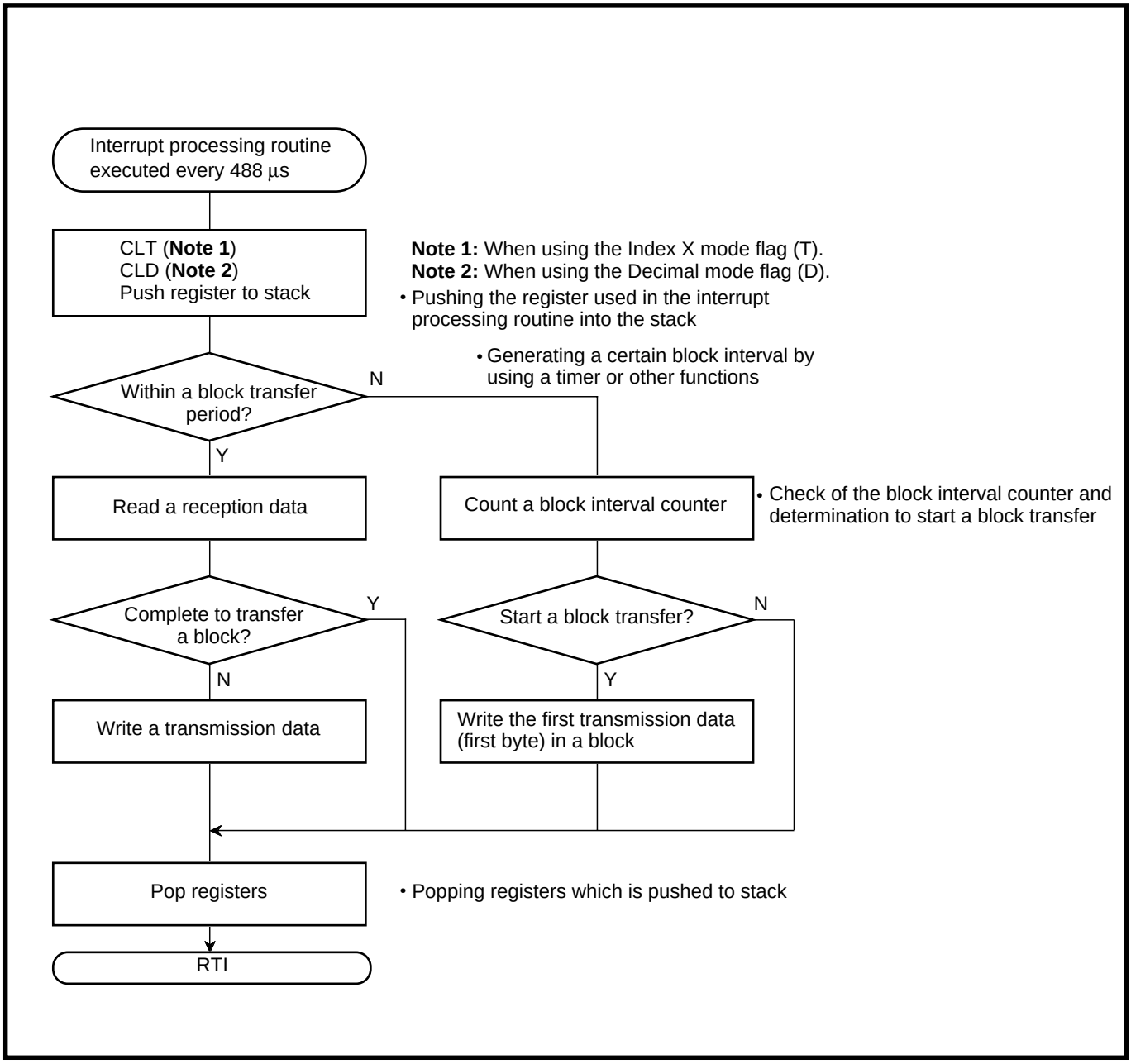


Fig. 2.3.24 Control procedure of master unit

● Control in the slave unit

After setting the relevant registers as shown in Figure 2.3.23, the slave unit becomes the state where a synchronous clock can be received at any time, and the serial I/O interrupt occurs each time an 8-bit synchronous clock is received.

In the serial I/O interrupt processing routine, the data to be transmitted next is written to the serial I/O register after the received data is read out.

However, if no serial I/O interrupt occurs for a certain time (heading adjustment time or more), the following processing will be performed.

1. The first 1-byte data of the transmission data in the block is written into the serial I/O register.
2. The data to be received next is processed as the first 1 byte of the received data in the block.

Figure 2.3.25 shows a control procedure of the slave unit using the serial I/O interrupt and any timer interrupt (for heading adjustment).

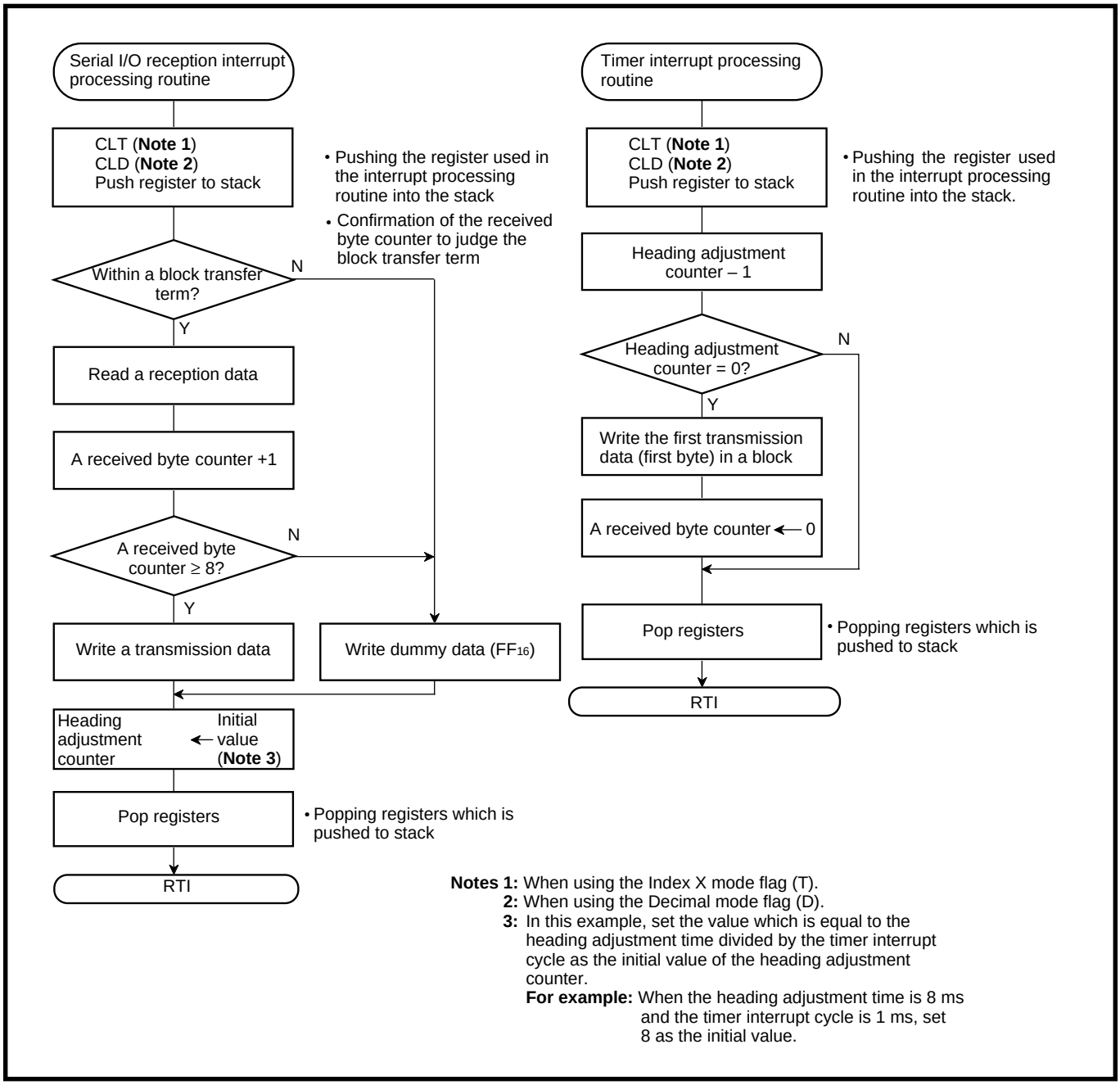


Fig. 2.3.25 Control procedure of slave unit

2.3.6 Notes on serial I/O

(1) Selecting external synchronous clock

When an external synchronous clock is selected, the contents of serial I/O register are being shifted continually while the transfer clock is input to the serial I/O1 clock pin. In this case, control the clock externally.

(2) Transmission data wiritng

When an external clock is used as the synchronous clock, write the transmit data to the serial I/O shift register at “H” level of transfer clock input.

2.4 LCD controller

This paragraph explains the registers setting method and the notes relevant to the LCD controller.

2.4.1 Memory map

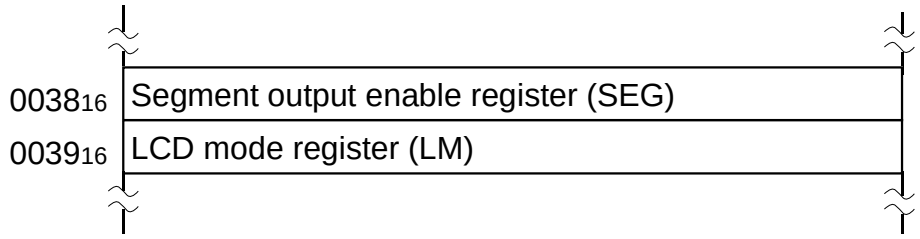


Fig. 2.4.1 Memory map of registers relevant to LCD controller

2.4.2 Relevant registers

Segment output enable register

b7 b6 b5 b4 b3 b2 b1 b0

Segment output enable register
(SEG: address 38_{16})

b	Name	Functions	At reset	R	W
0	Segment output enable bit 0	0: I/O ports $P2_0$–$P2_3$ 1: Segment output SEG_0–SEG_3	0	○	○
1	Segment output enable bit 1	0: I/O ports $P2_4$–$P2_7$ 1: Segment output SEG_4–SEG_7	0	○	○
2	Segment output enable bit 2	0: I/O ports $P0_0$–$P0_3$ 1: Segment output SEG_8–SEG_{11}	0	○	○
3	Segment output enable bit 3	0: I/O ports $P0_4$–$P0_7$ 1: Segment output SEG_{12}–SEG_{15}	0	○	○
4	Segment output enable bit 4	0: I/O ports $P1_0$–$P1_3$ 1: Segment output SEG_{16}–SEG_{19}	0	○	○
5	Segment output enable bit 5	0: I/O ports $P1_4$–$P1_7$ 1: Segment output SEG_{20}–SEG_{23}	0	○	○
6	Segment output enable bit 6	0: Output ports $P3_0$–$P3_3$ 1: Segment output SEG_{24}–SEG_{27}	0	○	○
7	Segment output enable bit 7	0: Output ports $P3_4$–$P3_7$ 1: Segment output SEG_{28}–SEG_{31}	0	○	○

Fig. 2.4.2 Structure of Segment output enable register

LCD mode register

b7 b6 b5 b4 b3 b2 b1 b0
0

LCD mode register
(LM: address 39_{16})

b	Name	Functions	At reset	R	W
0	Duty ratio selection bits	b1b0 0 0: 1 (use COM_0) 0 1: 2 (use COM_0, COM_1) 1 0: 3 (use COM_0–COM_2) 1 1: 4 (use COM_0–COM_3)	0	○	○
1			0	○	○
2	Bias control bit	0: 1/3 bias 1: 1/2 bias	0	○	○
3	LCD enable bit	0: LCD OFF 1: LCD ON	0	○	○
4	Fix “0” to this bit.		0	○	○
5	LCD circuit divider division ratio selection bits	b6b5 0 0: Clock input 0 1: 2 division of clock input 1 0: 4 division of clock input 1 1: 8 division of clock input	0	○	○
6			0	○	○
7	LCDCK count source selection bit **(Note)**	0: f(X_{CIN})/32 1: f(X_{IN})/8192 (f(X_{CIN})/8192 in low-speed mode)	0	○	○

Note: LCDCK is a clock for a LCD timing controller.

Fig. 2.4.3 Structure of LCD mode register

2.4 LCD controller

2.4.3 LCD controller application examples

Outline: A LCD panel display data by using the LCD controller.

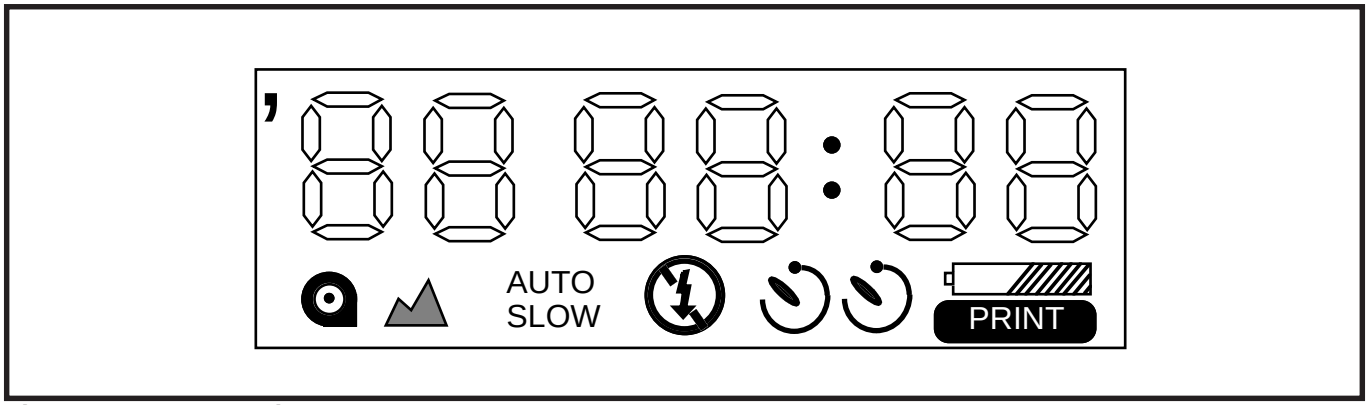


Fig. 2.4.4 LCD panel

Specifications: •Use of segment output SEG_0 to SEG_{13}
•Setting of port P1 to I/O port, setting of port P3 to output
•Frame frequency = 61 Hz
•Duty ratio number = 4, Bias value = 1/3

Figure 2.4.5 shows the segment allocation example.

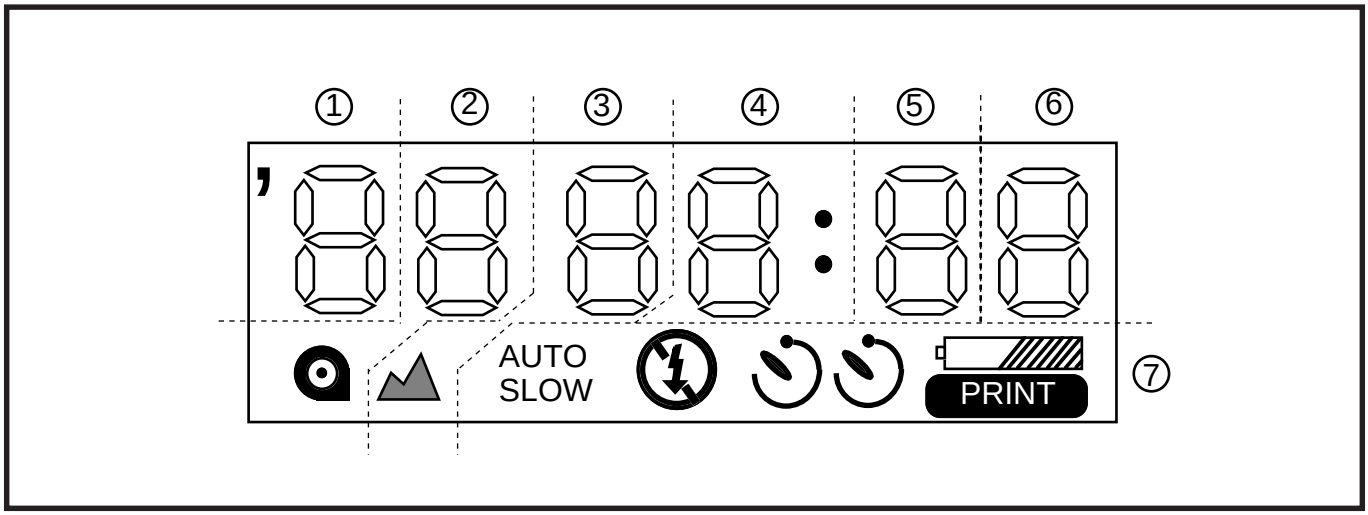


Fig. 2.4.5 Segment allocation example

Setting of LCD display RAM:

Bit / Address	7	6	5	4	3	2	1	0
	COM_3	COM_2	COM_1	COM_0	COM_3	COM_2	COM_1	COM_0
0040_{16}	SEG_1				SEG_0			
0041_{16}	SEG_3				SEG_2			
0042_{16}	SEG_5				SEG_4			
0043_{16}	SEG_7				SEG_6			
0044_{16}	SEG_9				SEG_8			
0045_{16}	SEG_{11}				SEG_{10}			
0046_{16}	SEG_{13}				SEG_{12}			
0047_{16}	SEG_{15}				SEG_{14}			
0048_{16}	SEG_{17}				SEG_{16}			
0049_{16}	SEG_{19}				SEG_{18}			
$004A_{16}$	SEG_{21}				SEG_{20}			
$004B_{16}$	SEG_{23}				SEG_{22}			
$004C_{16}$	SEG_{25}				SEG_{24}			
$004D_{16}$	SEG_{27}				SEG_{26}			
$004E_{16}$	SEG_{29}				SEG_{28}			
$004F_{16}$	SEG_{31}				SEG_{30}			

Fig. 2.4.6 LCD display RAM map

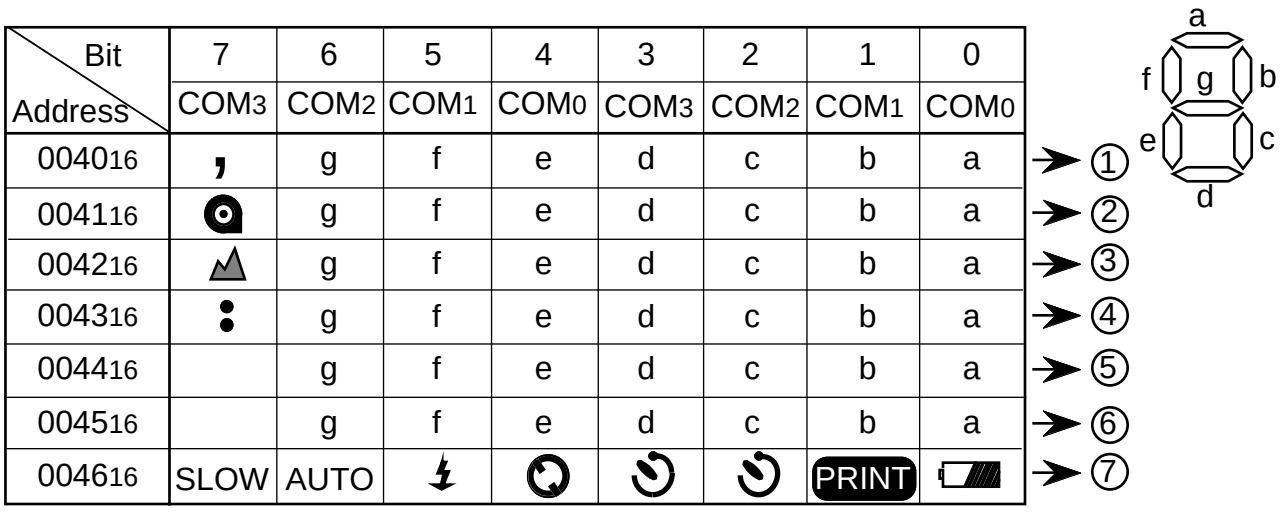
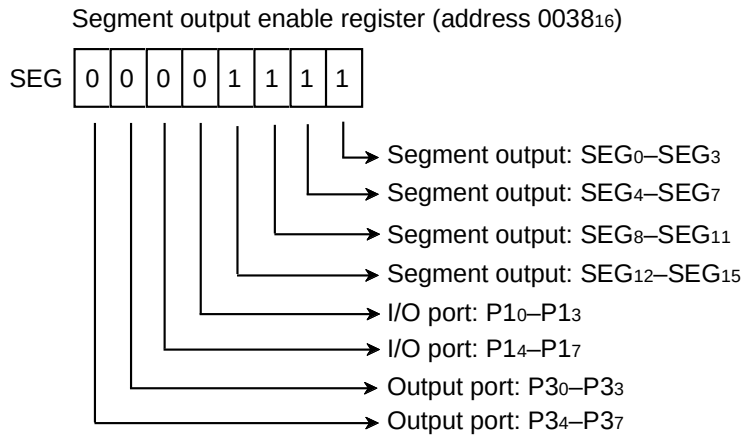
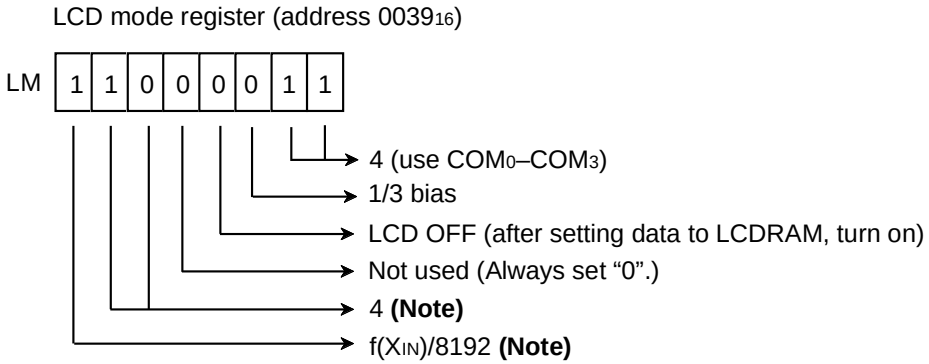


Fig. 2.4.7 LCD display RAM setting

2.4 LCD controller

Figure 2.4.8 shows the relevant registers setting.





Note: Frame frequency = f(LCDCK)/division ratio
f(LCDCK) = count source frequency for LCDCK/LCD circuit division ratio
From the above, the frame frequency at f(X_{IN}) = 8 MHz is as follows:

Frame frequency = $\{(8 \times 10^6/8192)/4\}/4 \cong 61.035$ Hz

Fig. 2.4.8 Relevant registers setting

Control procedure: Figure 2.4.9 shows the control procedure of relevant registers to turn on all the LCD display in Figure 2.4.4.

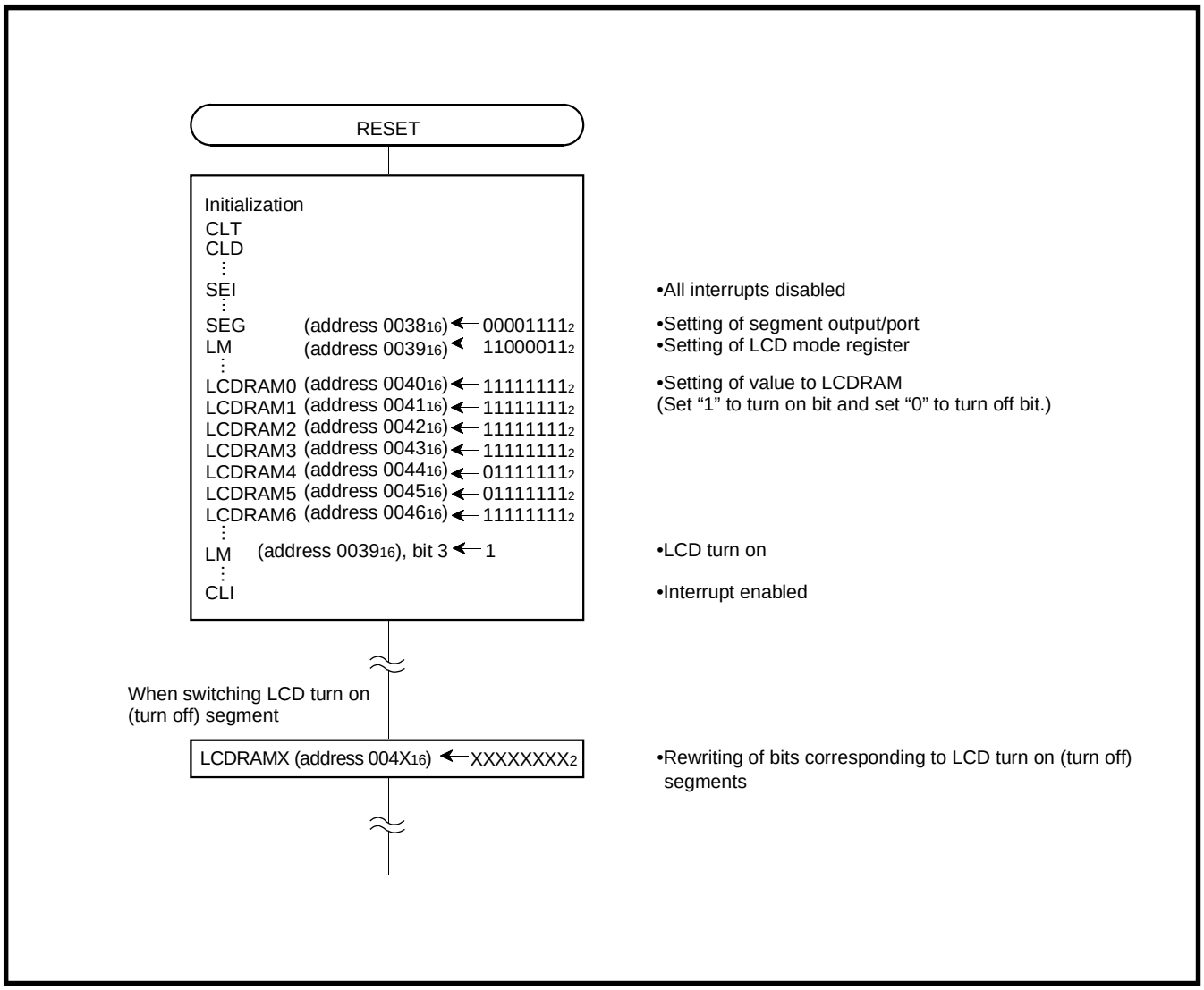


Fig. 2.4.9 Control procedure

2.4 LCD controller

2.4.4 Notes on LCD controller

●When switching from the high-speed or middle-speed mode to the low-speed mode, switch the mode in the following order:

(1) 32 kHz oscillation selected (bit 4 of CPU mode register (address $003B_{16}$) = "1")
(2) Count source for LCDCK = f(X_{CIN})/32 (bit 7 of LCD mode register (address 0039_{16}) = "0")
(3) Internal system clock: X_{CIN}-X_{COUT} selected (bit 7 of CPU mode register (address $003B_{16}$) = "1")
(4) Main clock X_{IN}–X_{OUT} stopped (bit 5 of CPU mode register (address $003B_{16}$) = "1")
Execute the setting (2) after the oscillation at 32 kHz (setting (1)) becomes completely stable.

●If the STP instruction is executed while the LCD is turned on by setting bit 3 of the LCD mode register (address 0039_{16}) to "1", a DC voltage is applied to the LCD. For this reason, do not execute the STP instruction while the LCD is lighting.

●When the LCD is not used, open the segment and the common pins.
Connect V_{L1}–V_{L3} to V_{SS}.

●For the following products, if the LCD enable bit of the LCD mode register (bit 3 of address 0039_{16}) is set to "0", all LCDs cannot be turned off. To turn off all LCDs, set "0" (turn off) to all corresponding LCD display RAM.

Corresponding products: M38C34M6AXXXFP, M38C34M6MXXXFP, M38C37ECAXXXFP, M38C37ECMXXXFP, M38C37ECAFP, M38C37ECMFP, M38C37ECAFS, M38C3ECMFS, M38C37RFS, M38C37RMFS

2.5 A-D converter

This paragraph describes the setting method of A-D converter relevant registers, notes etc.

2.5.1 Memory map

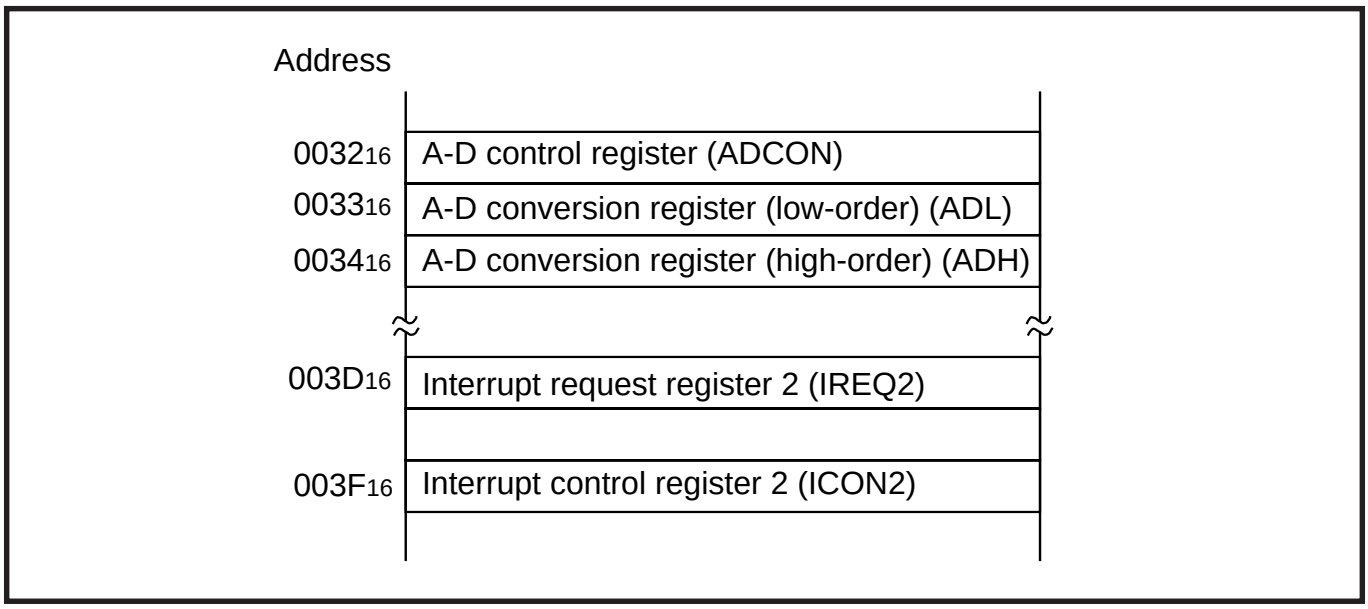


Fig. 2.5.1 Memory map of A-D converter relevant registers

2.5.2 Relevant registers

A-D control register

b7 b6 b5 b4 b3 b2 b1 b0

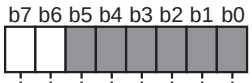
A-D control register
(ADCON: address 32_{16})

b	Name	Functions	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0: P6₀/AN₀ 0 0 1: P6₁/AN₁ 0 1 0: P6₂/AN₂ 0 1 1: P6₃/AN₃ 1 0 0: P6₄/AN₄ 1 0 1: P6₅/AN₅ 1 1 0: P6₆/AN₆ 1 1 1: P6₇/AN₇	0	○	○
1			0	○	○
2			0	○	○
3	Nothing is arranged for this bit. This is write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
4	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
6			0	○	×
7			0	○	×

Fig. 2.5.2 Structure of A-D control register

2.5 A-D converter

A-D conversion register (low-order)



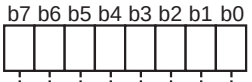
A-D conversion register (low-order)
(ADL: address 33_{16})

b	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”.	Undefined	○	×
1		Undefined	○	×
2		Undefined	○	×
3		Undefined	○	×
4		Undefined	○	×
5		Undefined	○	×
6	These are A-D conversion result (low-order 2 bits) stored bits. This is read exclusive register.	Undefined	○	×
7		Undefined	○	×

Note: Do not read this register during A-D conversion.

Fig. 2.5.3 Structure of A-D conversion register (low-order)

A-D conversion register (high-order)



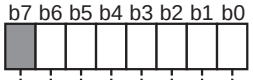
A-D conversion register (high-order)
(ADH: address 34_{16})

b	Functions	At reset	R	W
0	This is A-D conversion result (high-order 8 bits) stored bits. This is read exclusive register.	Undefined	○	×
1		Undefined	○	×
2		Undefined	○	×
3		Undefined	○	×
4		Undefined	○	×
5		Undefined	○	×
6		Undefined	○	×
7		Undefined	○	×

Note: Do not read this register during A-D conversion.

Fig. 2.5.4 Structure of A-D conversion register (high-order)

Interrupt request register 2



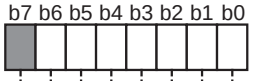
Interrupt request register 2
(IREQ2 : address $3D_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Timer 5 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Timer 6 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	CNTR1 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	AD conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

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

Fig. 2.5.5 Structure of Interrupt request register 2

Interrupt control register 2



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

b	Name	Functions	At reset	R	W
0	Timer 4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Timer 5 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
2	Timer 6 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
3	$CNTR_0$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
4	$CNTR_1$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key input interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD conversion interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Fig. 2.5.6 Structure of Interrupt control register 2

2.5.3 A-D converter application examples

(1) Read-in of analog signal

Outline: The analog input voltage from a sensor is converted to digital values.

Figure 2.5.7 shows a connection diagram, and Figure 2.5.8 shows the setting of relevant registers.

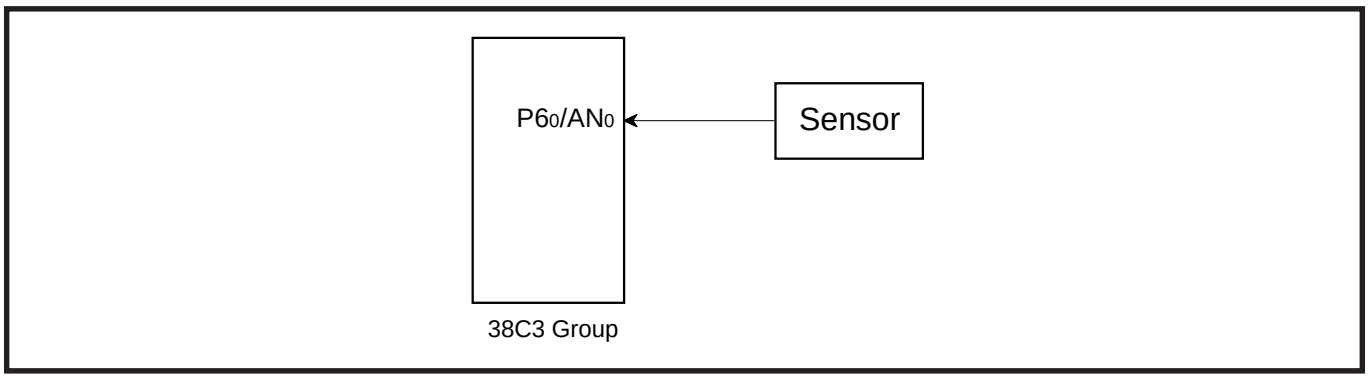
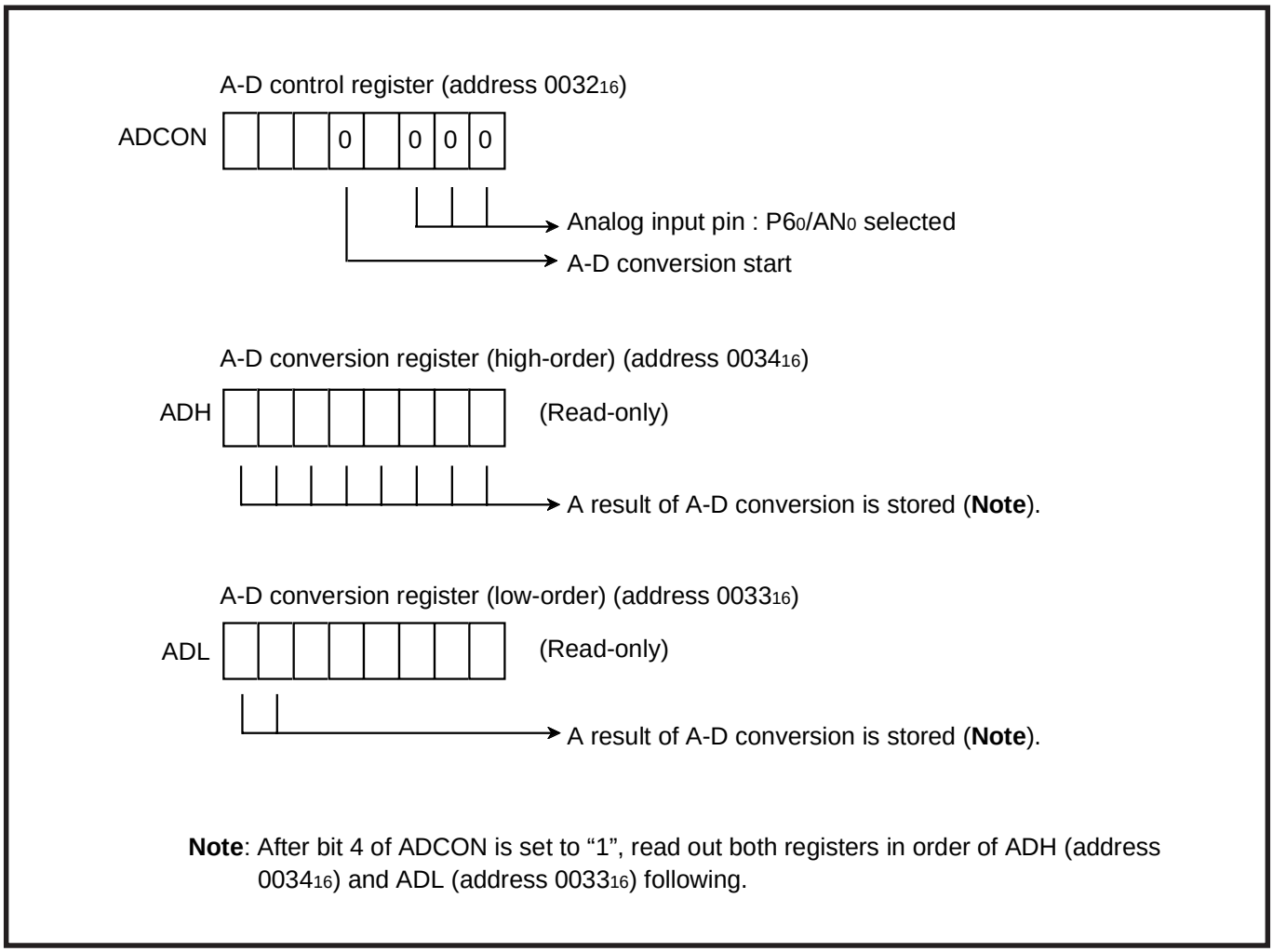


Fig. 2.5.7 Connection diagram

Specifications: •Conversion of analog input voltage input from sensor to digital values
•Use of $P6_0/AN_0$ pin as analog input pin



Note: After bit 4 of ADCON is set to "1", read out both registers in order of ADH (address 0034_{16}) and ADL (address 0033_{16}) following.

Fig. 2.5.8 Setting of relevant registers

Control procedure: A-D converter is started by performing register setting shown Figure 2.5.8. Figure 2.5.9 shows the control procedure.

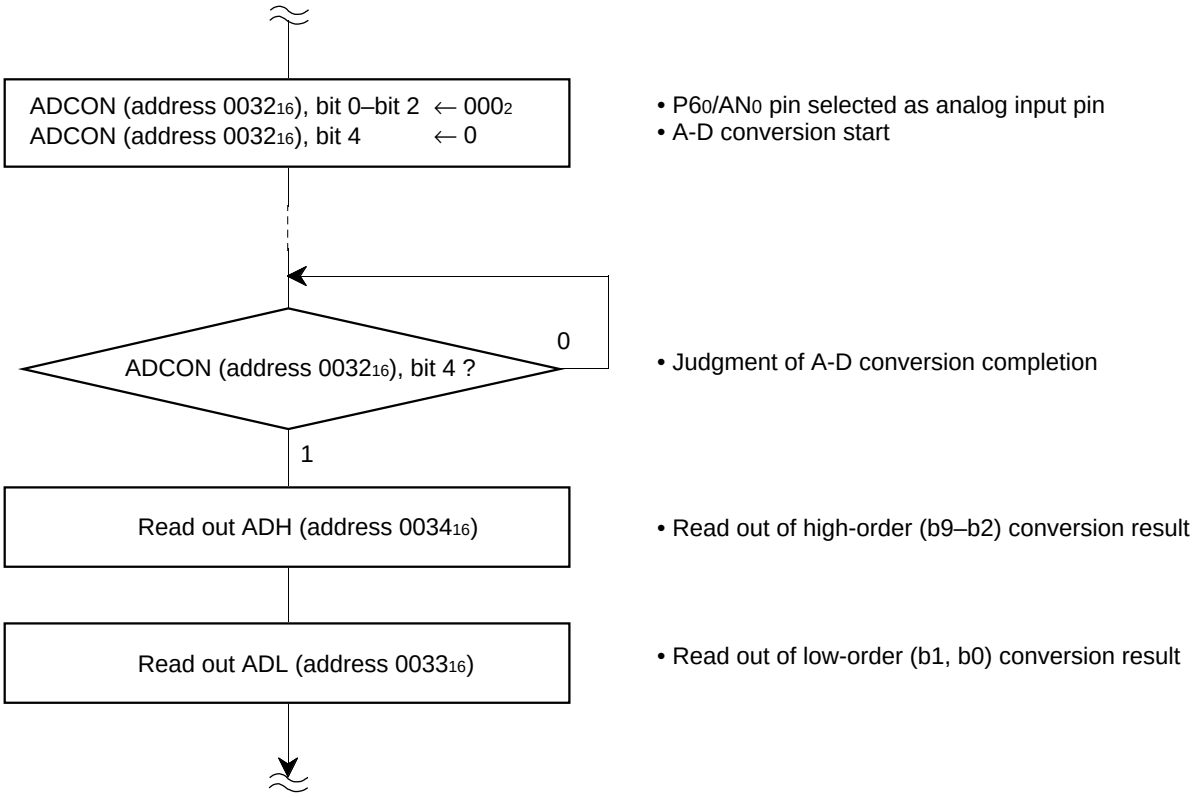


Fig. 2.5.9 Control procedure

2.5 A-D converter

2.5.4 Notes on A-D converter

(1) Analog input pin

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

● **Reason**

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

(2) A-D converter power source pin

The AVSS pin is A-D converter power source pin. Regardless of using the A-D conversion function or not, connect it as following :

• AVSS : Connect to the VSS line.

● **Reason**

If the AVSS pin is opened, the microcomputer may have a failure because of noise or others.

(3) Clock frequency during A-D conversion

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

• f(XIN) is 500 kHz or more.

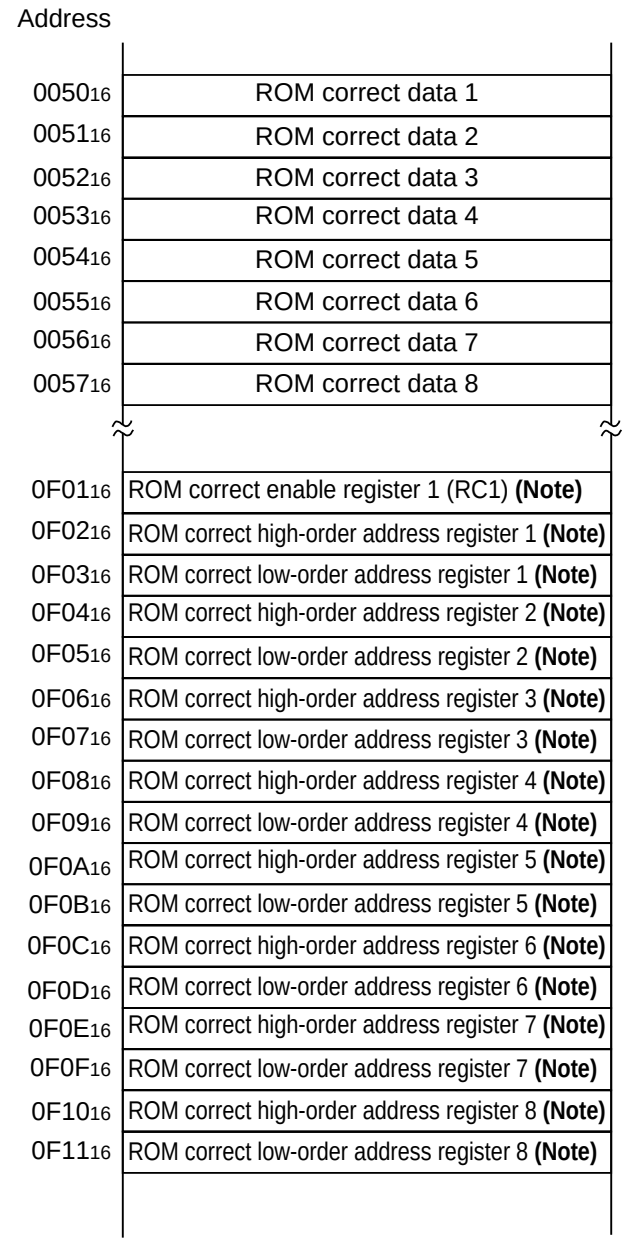
• Use clock divided by main clock (f(XIN)) as internal system clock.

• Do not execute the **STP** instruction and **WIT** instruction.

2.6 ROM correct function

This paragraph describes the setting method of ROM correct function relevant registers, notes etc.

2.6.1 Memory map



Note: This register is valid only in mask ROM version.

Fig. 2.6.1 Memory map of ROM correct function relevant registers

2.6 ROM correct function

2.6.2 Relevant registers

ROM correct enable register 1

b7 b6 b5 b4 b3 b2 b1 b0

ROM correct enable register 1
(RC1: address 0F01$_{16}$)

b	Name	Functions	At reset	R	W
0	ROM correct address 1 enable bit	0: Disabled 1: Enabled	0	○	○
1	ROM correct address 2 enable bit	0: Disabled 1: Enabled	0	○	○
2	ROM correct address 3 enable bit	0: Disabled 1: Enabled	0	○	○
3	ROM correct address 4 enable bit	0: Disabled 1: Enabled	0	○	○
4	ROM correct address 5 enable bit	0: Disabled 1: Enabled	0	○	○
5	ROM correct address 6 enable bit	0: Disabled 1: Enabled	0	○	○
6	ROM correct address 7 enable bit	0: Disabled 1: Enabled	0	○	○
7	ROM correct address 8 enable bit	0: Disabled 1: Enabled	0	○	○

Fig. 2.6.2 Structure of ROM correct enable register 1

2.6.3 ROM correct function application examples

Outline: When the contents of ROM would be corrected, the contents of ROM can be changed artificially by connecting E²PROM to the externals and storing the contents (correct address, correct data) to the ROM correct function relevant registers.

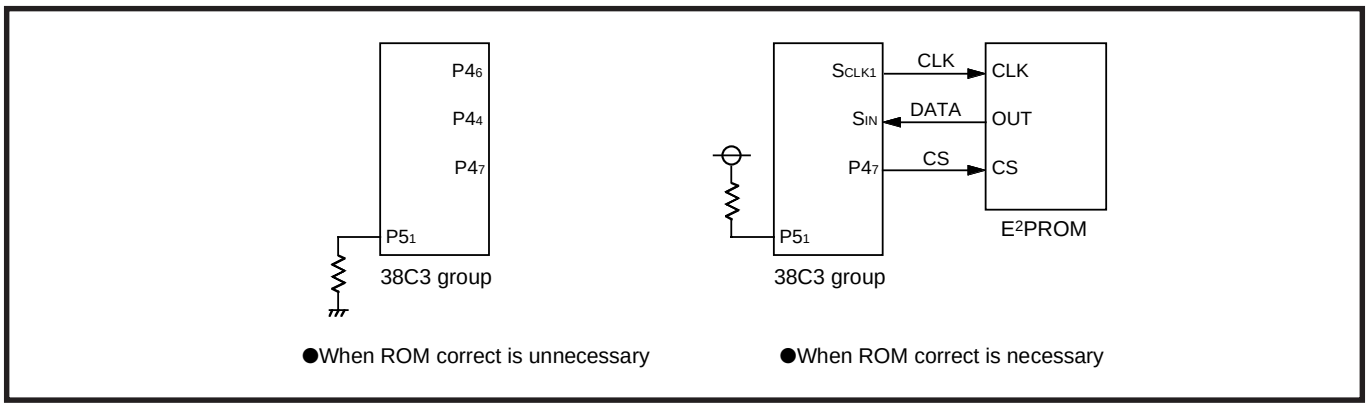


Fig. 2.6.3 Connection diagram

Specifications: ●If ROM correct is necessary, make P5₁ pull-up. If ROM correct is unnecessary, make P5₁ pull-down.
●Use of serial I/O as communication with E²PROM
●Connection of clock and SCLK1, connection of CS pin and P47

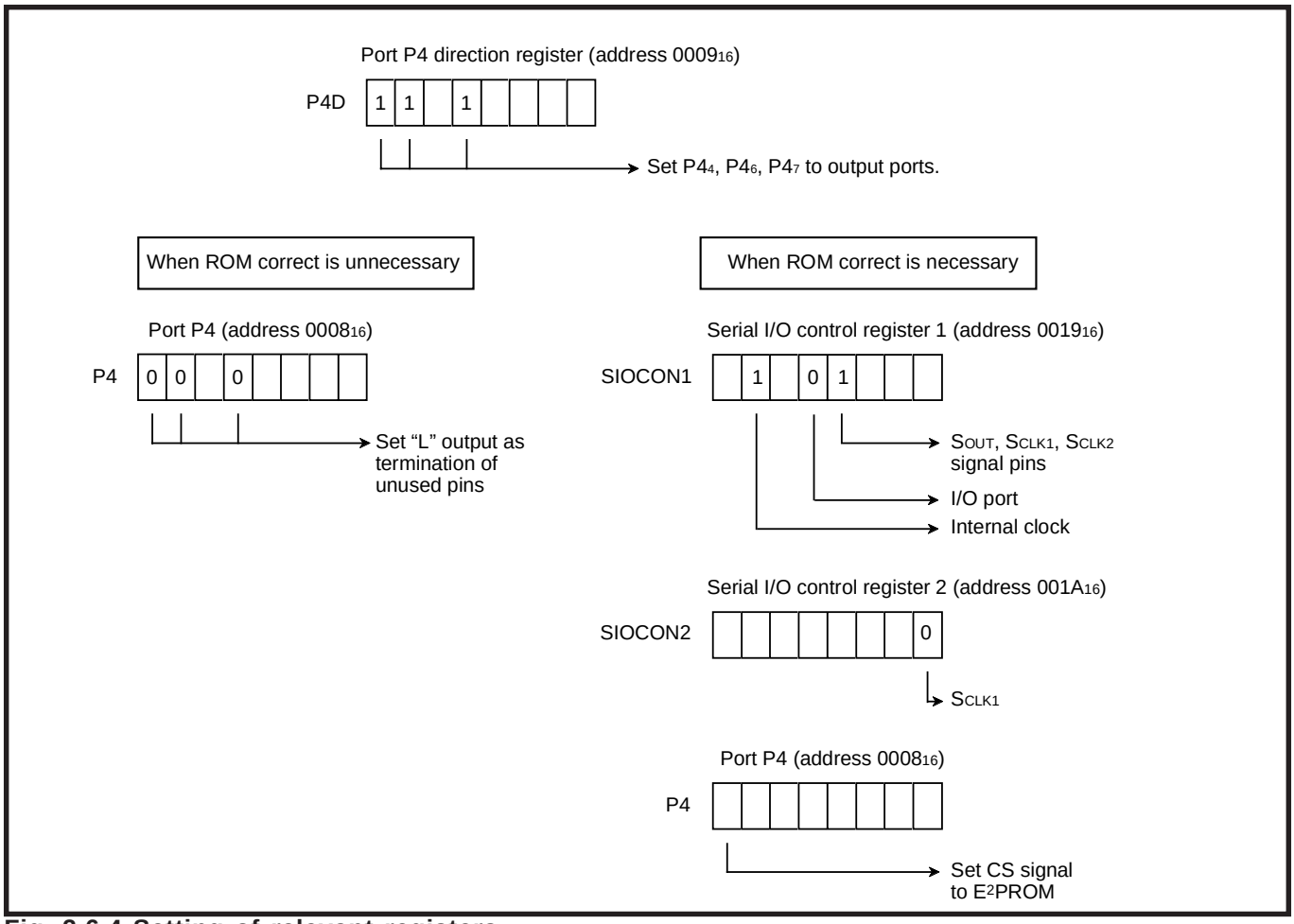


Fig. 2.6.4 Setting of relevant registers

2.6 ROM correct function

Control procedure:

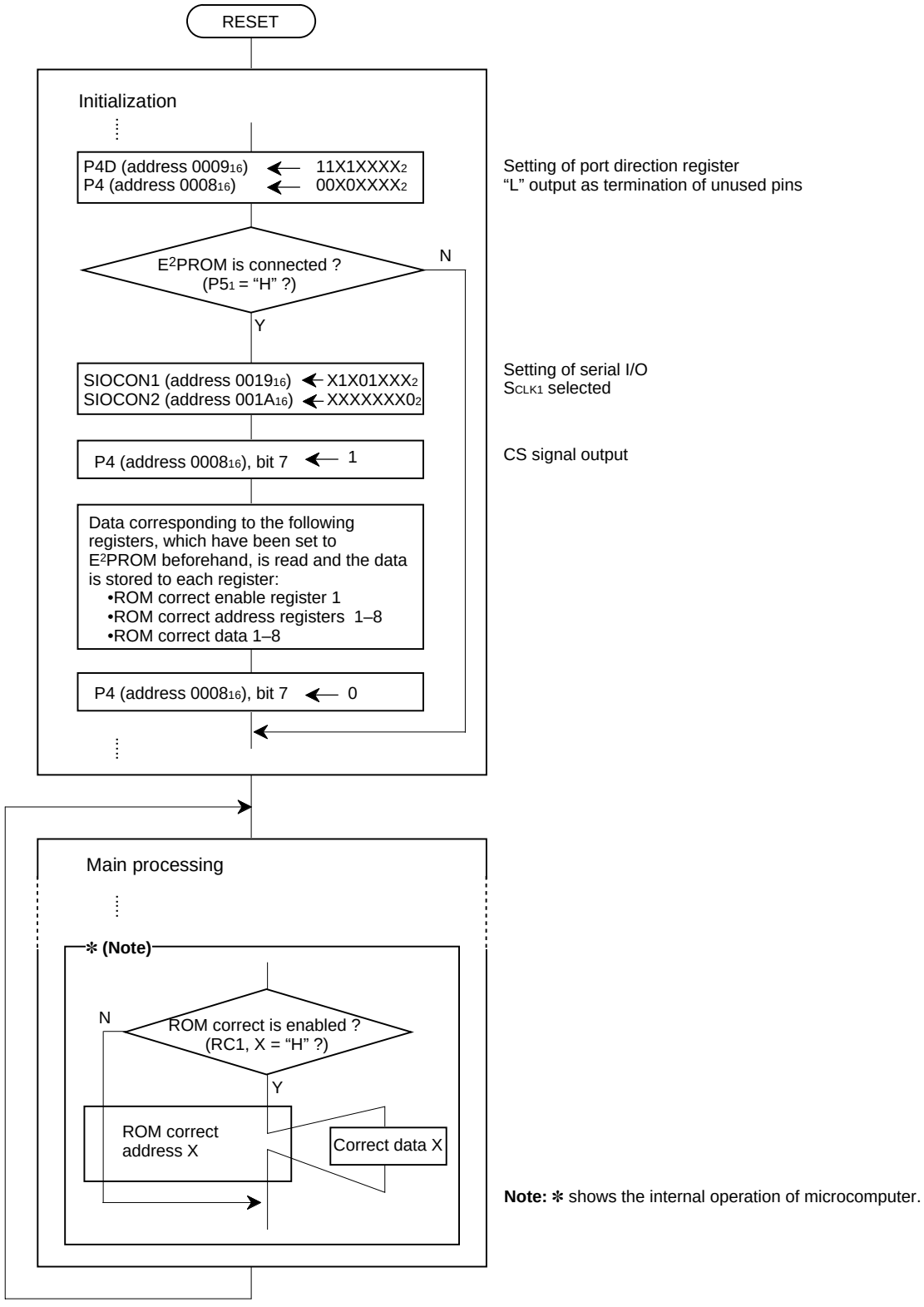


Fig. 2.6.5 Control procedure

2.7 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.

2.7.1 Connection example of reset IC

Figure 2.7.1 shows the example of power-on reset circuit. Figure 2.7.2 shows the system example which switches to the RAM backup mode by detecting a drop of the system power source voltage with the INT interrupt.

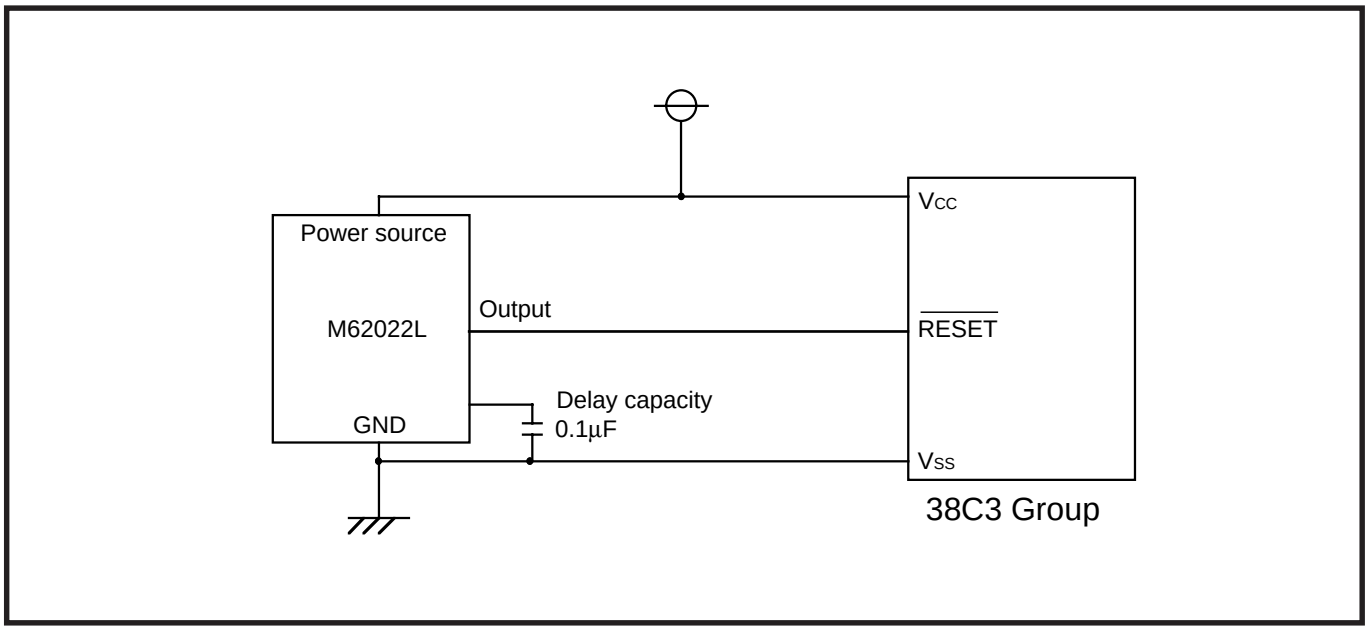


Fig. 2.7.1 Example of power-on reset circuit

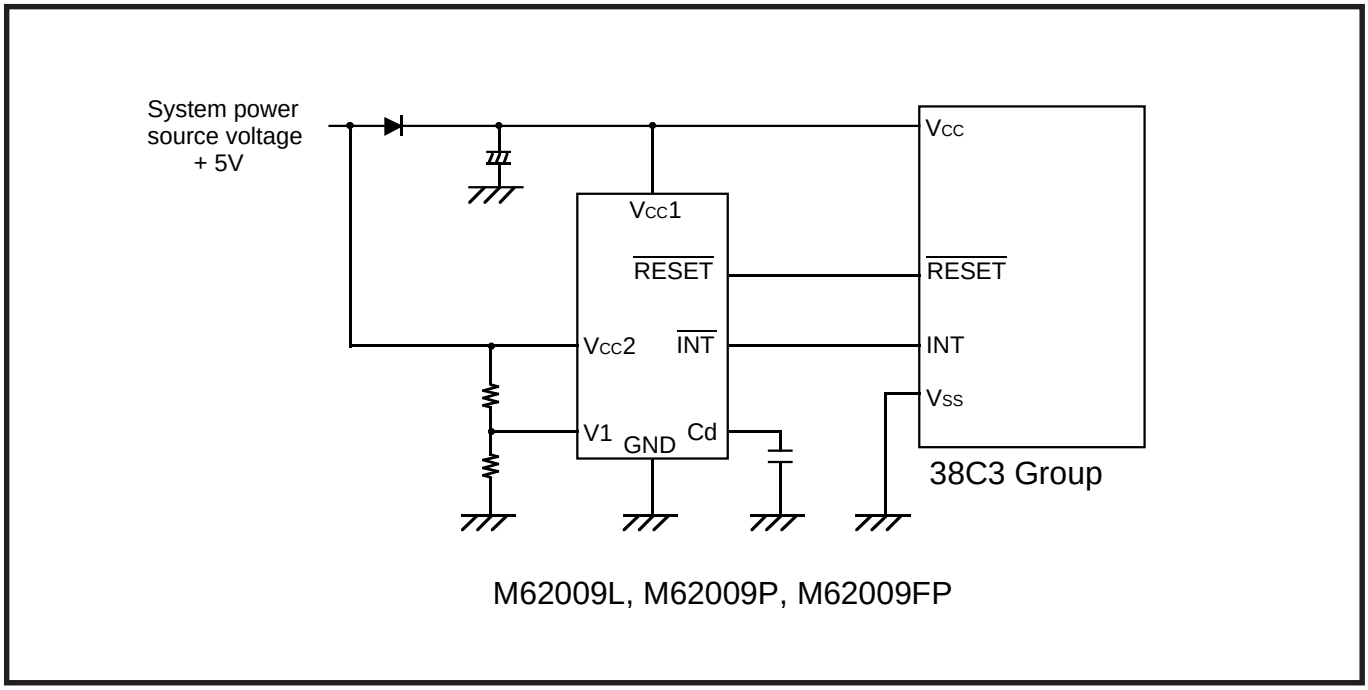


Fig. 2.7.2 RAM backup system example

2.7 Reset circuit

2.7.2 Notes on reset circuit

(1) Reset input voltage control

Make sure that the reset input voltage is 0.5 V or less for Vcc of 2.5 V **(Note)**.
Perform switch to the high-speed mode when power source voltage is within 4.0 to 5.5 V.
Note: M version of mask ROM version is 2.2 V.

(2) Countermeasure when $\overline{\text{RESET}}$ signal rise time is long

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.

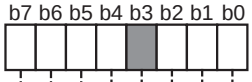
2.8 Clock generating circuit

This paragraph describes the setting method of clock generating circuit relevant registers, application examples etc.

2.8.1 Relevant register

Figure 2.8.1 shows the structure of the CPU mode register.

CPU mode register



CPU mode register
(CPUM, CM: address $3B_{16}$)

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : 10 : 11 : Not available	0	○	○
1			0	○	○
2	Stack page selection bit	0 : Page 0 1 : Page 1	0	○	○
3	Nothing is arranged for this bit. When this bit is read out, the contents are “1”. Do not write “0” to this bit.		1	○	○
4	Port Xc switch bit	0: I/O port function 1: XCIN-XCOUT oscillation function	0	○	○
5	Main clock (XIN-XOUT) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bit	0: f(XIN)/2 (high-speed mode) 1: f(XIN)/8 (middle-speed mode)	1	○	○
7	Internal system clock selection bit	0: XIN–XOUT selection (middle-/high-speed mode) 1: XCIN–XCOUT selection (low-speed mode)	0	○	○

Fig. 2.8.1 Structure of CPU mode register

2.8 Clock generating circuit

2.8.2 Clock generating circuit application examples

(1) Status transition during power failure

Outline: The clock is counted up every one second by using the timer interrupt during a power failure.

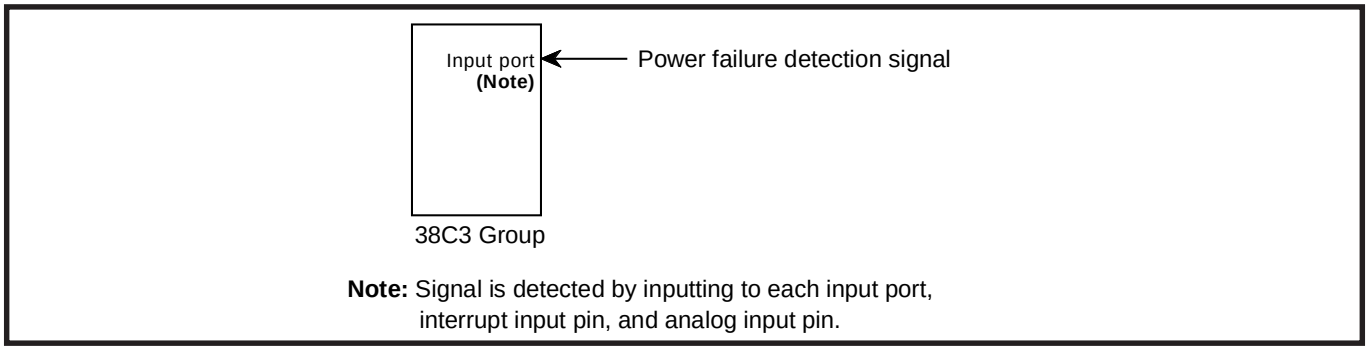


Fig. 2.8.2 Connection diagram

Specifications: •Reducing power dissipation as low as possible while maintaining clock function
•Clock: $f(X_{IN})$ = 8 MHz, $f(X_{CIN})$ = 32.768 kHz
•Port processing
Input port: Fixed to "H" or "L" level on the external
Output port: Fixed to output level that does not cause current flow to the external
(Example) When a circuit turns on LED at "L" output level, fix the output level to "H".
I/O port: Input port → Fixed to "H" or "L" level on the external
Output port → Output of data that does not consume current
V_{REF}: Stop to supply to reference voltage input pin by external circuit

Figure 2.8.3 shows the status transition diagram during power failure and Figure 2.8.4 shows the setting of relevant registers.

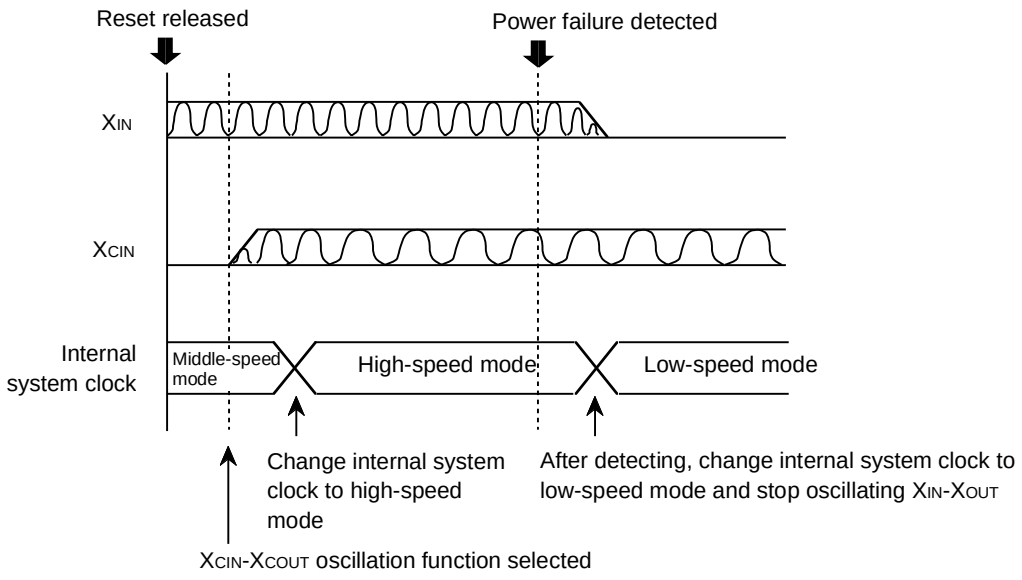
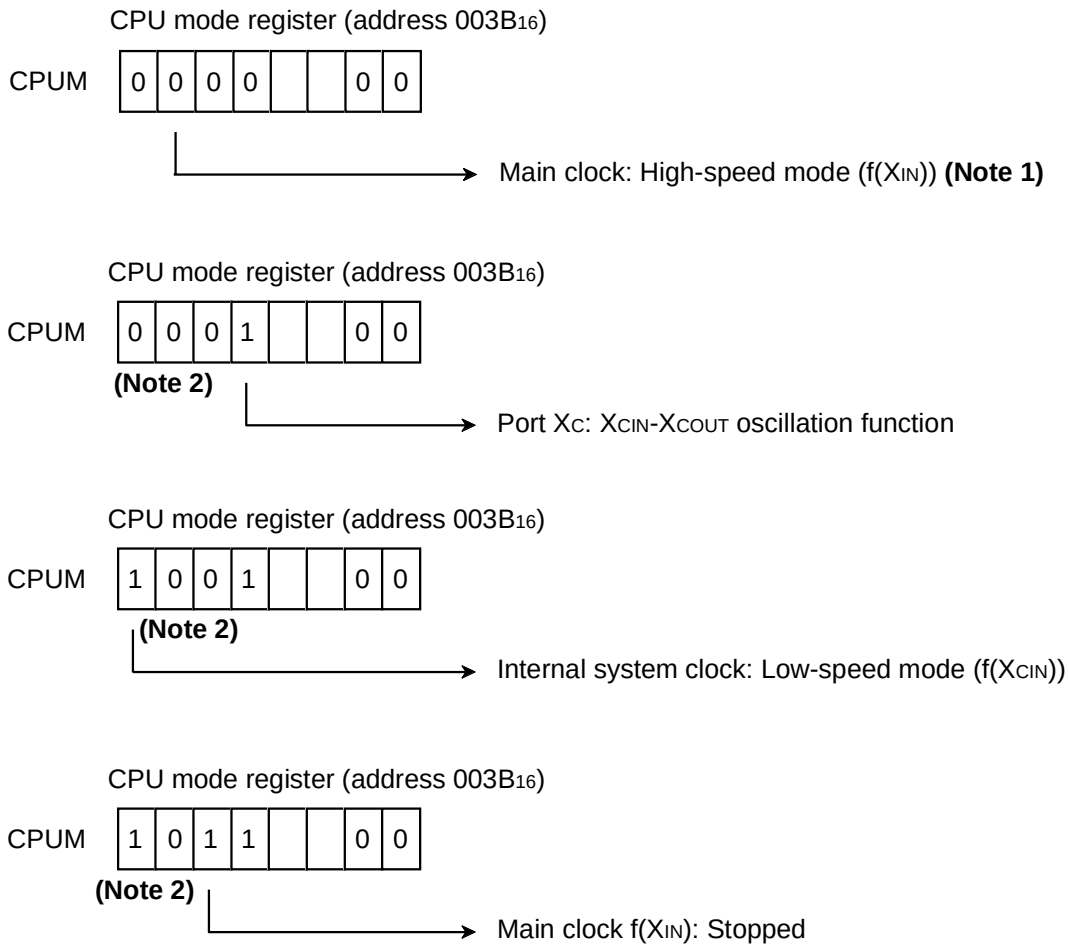


Fig. 2.8.3 Status transition diagram during power failure

2.8 Clock generating circuit



Notes 1: This setting is necessary only when selecting the high-speed mode.
2: When selecting the middle-speed mode, bit 6 is “1”.

Fig. 2.8.4 Setting of relevant registers

Control procedure: Set the relevant registers in the order shown below to prepare for a power failure.

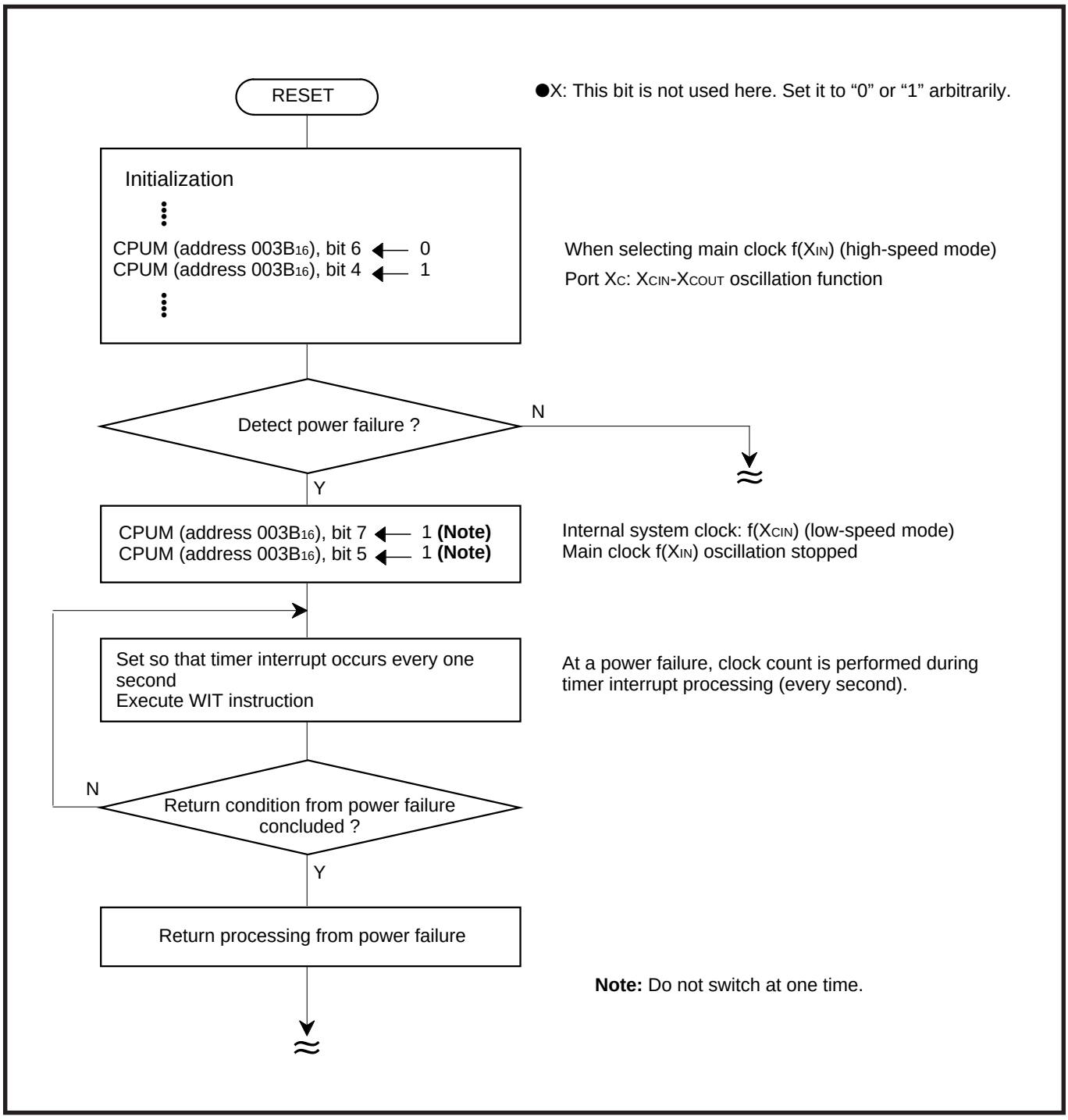


Fig. 2.8.5 Control procedure

(2) Counting without clock error during power failure

Outline: It keeps counting without clock error during a power failure.

Specifications: •Reducing power consumption as low as possible while maintaining clock function
•Clock: f(X_{IN}) = 4.19 MHz
•Sub clock: f(X_{CIN}) = 32.768 kHz
•Use of Timer 3 interrupt

For the peripheral circuit and the status transition during a power failure, refer to **"Figures 2.8.2 and 2.8.3"**.

Figure 2.8.6 shows the structure of clock counter, Figures 2.8.7 and 2.8.8 show the setting of relevant registers.

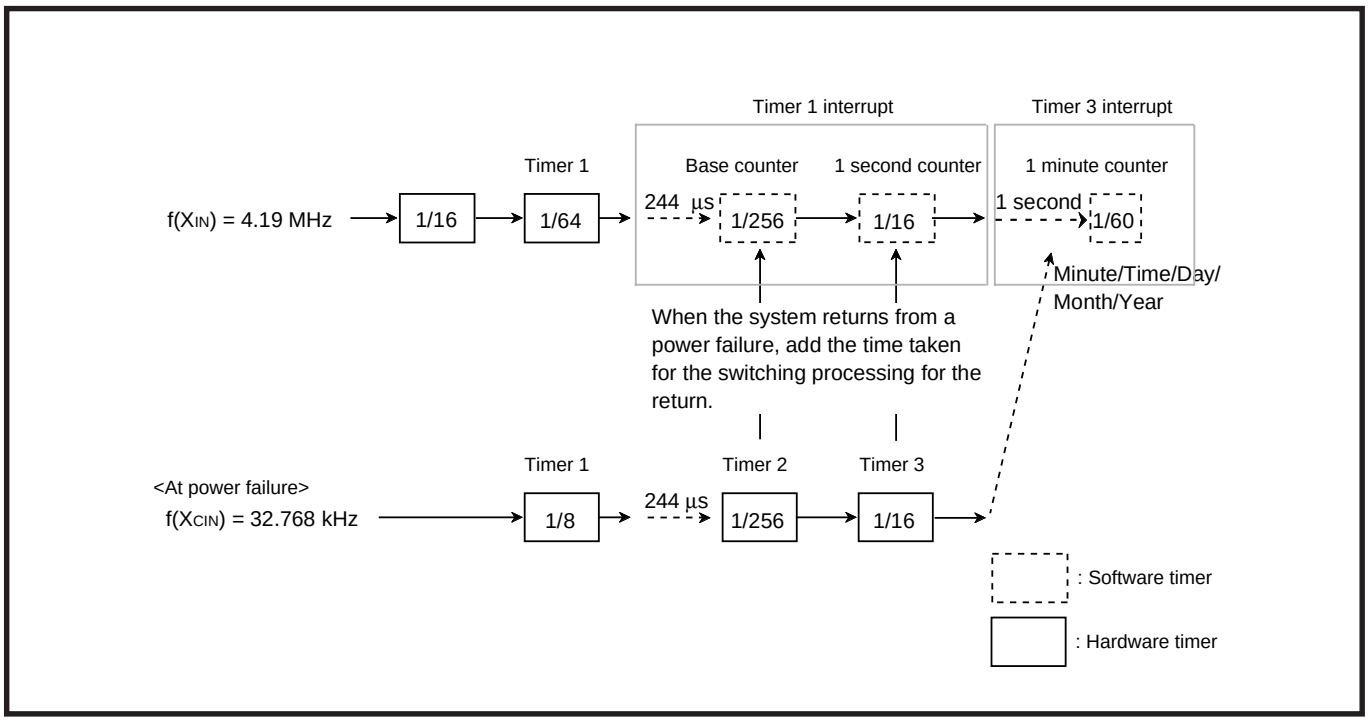


Fig. 2.8.6 Structure of clock counter

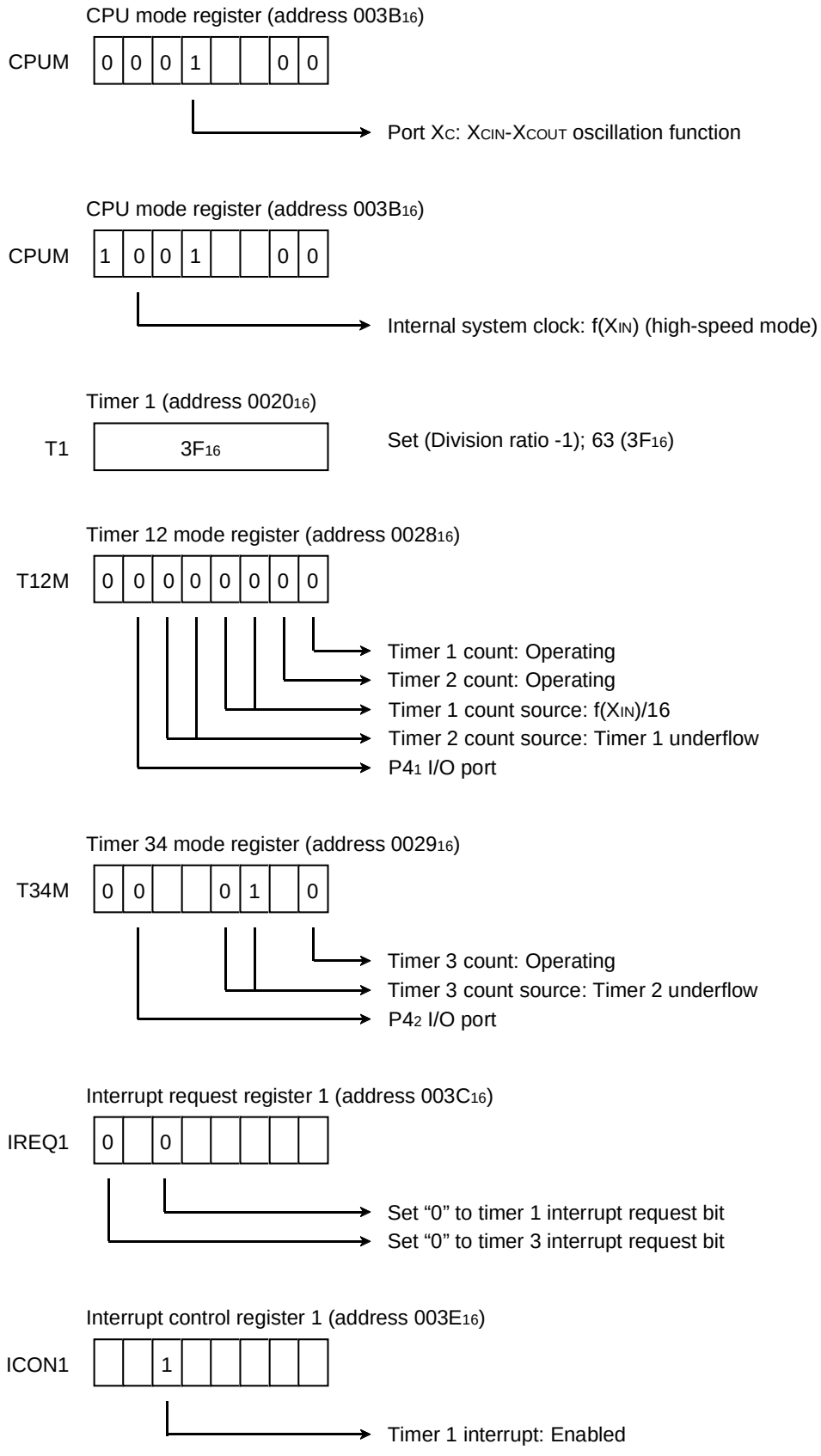


Fig. 2.8.7 Initial setting of relevant registers

2.8 Clock generating circuit

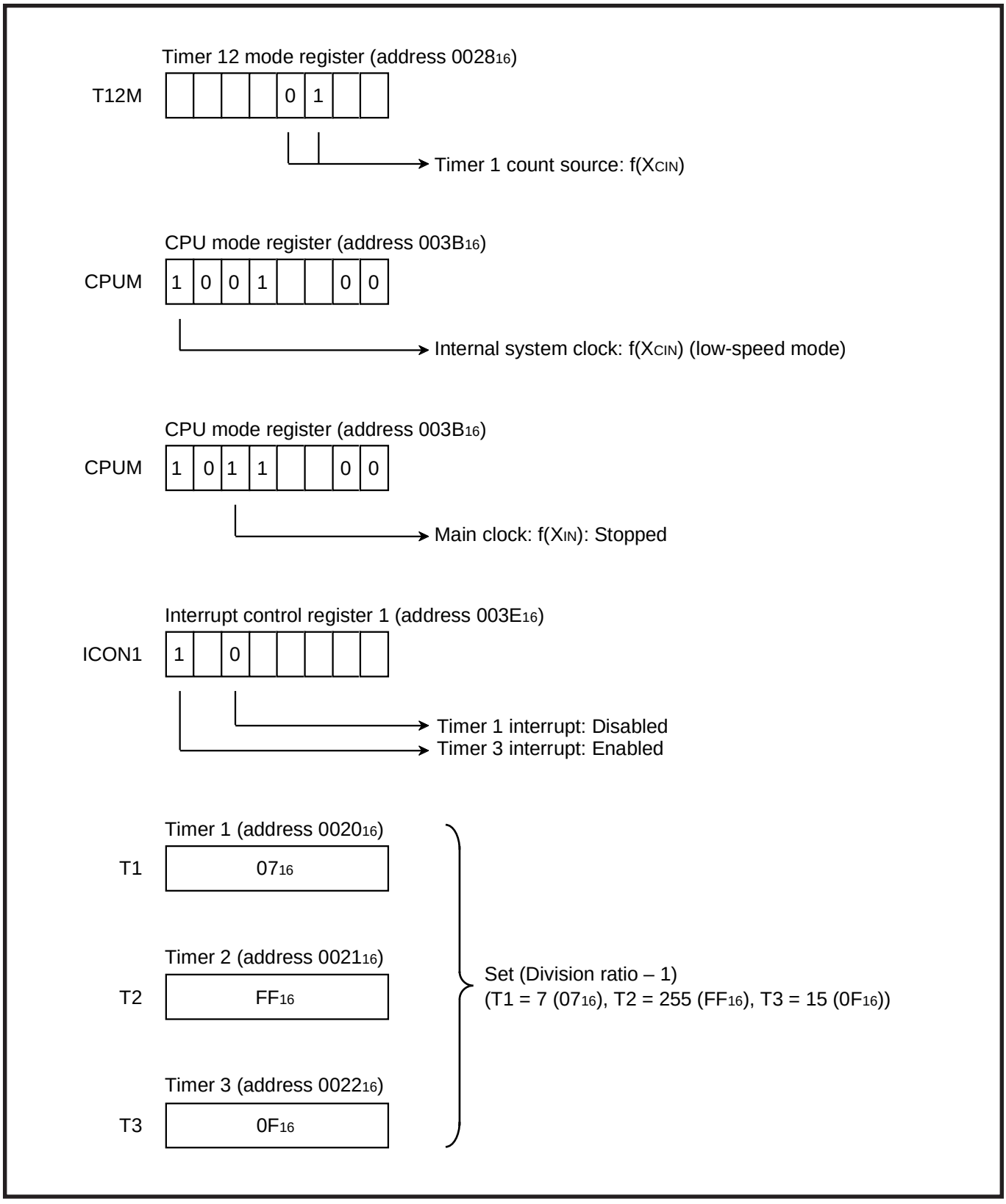


Fig. 2.8.8 Setting of relevant registers after detecting power failure

Control procedure: Set the relevant registers in the order shown below to prepare for a power failure.

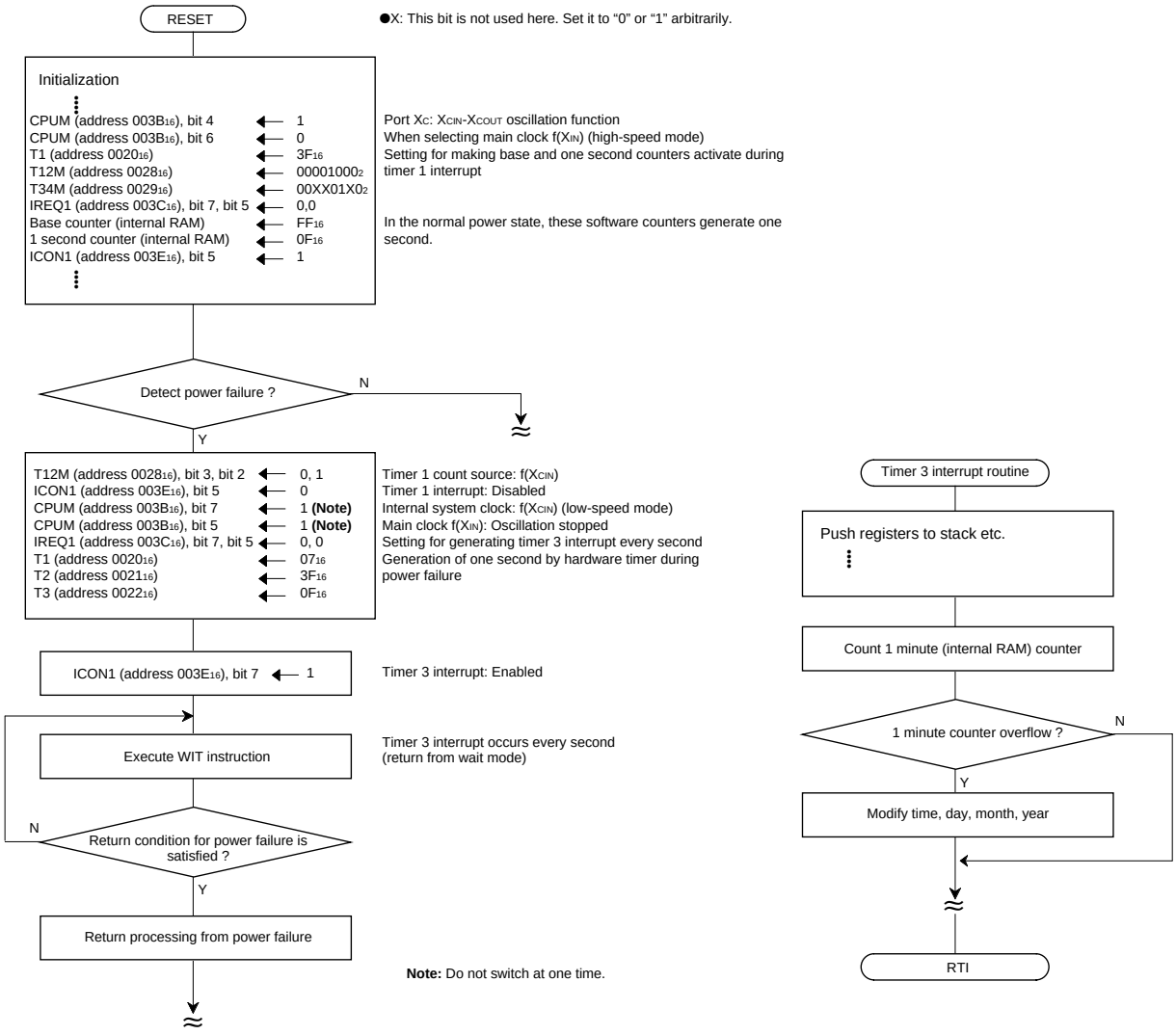


Fig. 2.8.9 Control procedure

MEMORANDUM

CHAPTER 3

APPENDIX

3.1 Electrical characteristics

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

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 $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, $P8_0$–$P8_7$		–0.3 to V_{CC}+0.3	V
V_I	Input voltage V_{L1}		–0.3 to V_{L2}	V
V_I	Input voltage V_{L2}		V_{L1} to V_{L3}	V
V_I	Input voltage V_{L3}		V_{L2} to V_{CC}+0.3	V
V_I	Input voltage $\overline{RESET}$, X_{IN}		–0.3 to V_{CC}+0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$	At output port	–0.3 to V_{CC}+0.3	V
		At segment output	–0.3 to V_{L3}+0.3	V
V_O	Output voltage COM_0–COM_3		–0.3 to V_{L3}+0.3	V
V_O	Output voltage $P4_0$–$P4_7$, $P5_0$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, $P8_0$–$P8_7$		–0.3 to V_{CC}+0.3	V
V_O	Output voltage X_{OUT}		–0.3 to V_{CC}+0.3	V
P_d	Power dissipation	Ta = 25°C	300	mW
T_{opr}	Operating temperature		–20 to 85	°C
T_{stg}	Storage temperature		–40 to 125	°C

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions

(Vcc = 2.5 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage	High-speed mode f(X_{IN}) = 8 MHz	4.0	5.0	5.5	V
		Middle-speed mode f(X_{IN}) = 8 MHz	2.5	5.0	5.5	V
		Low-speed mode	2.5	5.0	5.5	V
V_{SS}	Power source voltage			0		V
V_{REF}	A-D converter reference voltage		2.0		V_{CC}	V
AV_{SS}	Analog power source voltage			0		V
V_{IA}	Analog input voltage AN_0–AN_7		AV_{SS}		V_{CC}	V
V_{IH}	“H” input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$		0.7V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (CM4 = 0)		0.8V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage $P8_0$–$P8_7$		0.4V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage $\overline{RESET}$		0.8V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage X_{IN}		0.8V_{CC}		V_{CC}	V
V_{IL}	“L” input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$		0		0.3V_{CC}	V
V_{IL}	“L” input voltage $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (CM4 = 0)		0		0.2V_{CC}	V
V_{IL}	“L” input voltage $P8_0$–$P8_7$		0		0.16V_{CC}	V
V_{IL}	“L” input voltage $\overline{RESET}$		0		0.2V_{CC}	V
V_{IL}	“L” input voltage X_{IN}		0		0.2V_{CC}	V

Table 3.1.3 Recommended operating conditions

(Vcc = 2.5 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$\Sigma I_{OH(peak)}$	"H" total peak output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ $P8_0$–$P8_7$, $P5_0$			–60	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current **(Note 1)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			–30	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			40	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current **(Note 1)** $P8_0$–$P8_7$, $P5_0$			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current **(Note 1)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			40	mA
$\Sigma I_{OH(avg)}$	"H" total average output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ $P8_0$–$P8_7$, $P5_0$			–30	mA
$\Sigma I_{OH(avg)}$	"H" total average output current **(Note 1)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			–15	mA
$\Sigma I_{OL(avg)}$	"L" total average output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			20	mA
$\Sigma I_{OL(avg)}$	"L" total average output current **(Note 1)** $P8_0$–$P8_7$, $P5_0$			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current **(Note 1)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			20	mA
$I_{OH(peak)}$	"H" peak output current **(Note 2)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			–4.0	mA
$I_{OH(peak)}$	"H" peak output current **(Note 2)** $P4_0$–$P4_7$, $P5_0$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ $P8_0$–$P8_7$			–10	mA
$I_{OL(peak)}$	"L" peak output current **(Note 2)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			5.0	mA
$I_{OL(peak)}$	"L" peak output current **(Note 2)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			10	mA
$I_{OL(peak)}$	"L" peak output current **(Note 2)** $P8_0$–$P8_7$, $P5_0$			30	mA
$I_{OH(avg)}$	"H" average output current **(Note 3)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			–2.0	mA
$I_{OH(avg)}$	"H" average output current **(Note 3)** $P4_0$–$P4_7$, $P5_0$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ $P8_0$–$P8_7$			–5.0	mA
$I_{OL(avg)}$	"L" average output current **(Note 3)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			2.5	mA
$I_{OL(avg)}$	"L" average output current **(Note 3)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			5.0	mA
$I_{OL(avg)}$	"L" average output current **(Note 3)** $P8_0$–$P8_7$, $P5_0$			15	mA

Notes 1: The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100 ms. The total peak current is the peak value of all the currents.

2: The peak output current is the peak current flowing in each port.

3: The average output current is average value measured over 100 ms.

APPENDIX

3.1 Electrical characteristics

Table 3.1.4 Recommended operating conditions

(Vcc = 2.5 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
f(CNTR0)	Input frequency (duty cycle 50%)	(4.0 V ≤ Vcc ≤ 5.5 V)			4.0	MHz
f(CNTR1)		(Vcc ≤ 4.0 V)			(2×Vcc)–4	MHz
f(XIN)	Main clock input oscillation frequency **(Note 4)**	High-speed mode (4.0 V ≤ Vcc ≤ 5.5 V)			8.0	MHz
		High-speed mode (Vcc ≤ 4.0 V)			(4×Vcc)–8	MHz
		Middle-speed mode			8.0	MHz
f(XCIN)	Sub-clock input oscillation frequency **(Notes 4, 5)**			32.768	50	kHz

Notes 4: When the oscillation frequency has a duty cycle of 50%.

5: When using the microcomputer in low-speed mode, set the sub-clock input oscillation frequency on condition that f(XCIN) < f(XIN)/3.

3.1.3 Electrical characteristics

Table 3.1.5 Electrical characteristics

(V_{CC} = 4.0 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	"H" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$	I_{OH} = –2.0 mA	V_{CC}–2.0			V
		I_{OH} = –0.6 mA V_{CC} = 2.5 V	V_{CC}–1.0			V
V_{OH}	"H" output voltage $P4_0$–$P4_7$, $P5_0$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, **(Note)** $P8_0$–$P8_7$	I_{OH} = –5 mA	V_{CC}–2.0			V
		I_{OH} = –1.25 mA	V_{CC}–0.5			V
		I_{OH} = –1.25 mA V_{CC} = 2.5 V	V_{CC}–1.0			V
V_{OL}	"L" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$	I_{OL} = 2.5 mA			2.0	V
		I_{OL} = 1.25 mA			0.5	V
		I_{OL} = 1.25 mA V_{CC} = 2.5 V			1.0	V
V_{OL}	"L" output voltage $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ **(Note)**	I_{OL} = 5.0 mA			2.0	V
		I_{OL} = 2.5 mA			0.5	V
		I_{OL} = 2.5 mA V_{CC} = 2.5 V			1.0	V
V_{OL}	"L" output voltage $P8_0$–$P8_7$, $P5_0$	I_{OL} = 15 mA			2.0	V
V_{T+}–V_{T-}	Hysteresis INT_0–INT_2, $CNTR_0$, $CNTR_1$, $P8_0$–$P8_7$			0.5		V
V_{T+}–V_{T-}	Hysteresis $SCLK_1$, S_{IN}			0.5		V
V_{T+}–V_{T-}	Hysteresis $\overline{RESET}$	$\overline{RESET}$: V_{CC} = 2.5 V – 5.5 V		0.5		V
I_{IH}	"H" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$	V_I = V_{CC} Pull-down "off"			5.0	µA
		V_{CC} = 5.0 V, V_I = V_{CC} Pull-down "on"	30	70	140	µA
		V_{CC} = 3.0 V, V_I = V_{CC} Pull-down "on"	6.0	25	45	µA
I_{IH}	"H" input current $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, $P8_0$–$P8_7$	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 $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_1$				–5.0	µA
I_{IL}	"L" input current $P4_0$–$P4_7$, $P5_0$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, $P8_0$–$P8_7$	V_I = V_{SS} Pull-up "off"			–5.0	µA
		V_{CC} = 5.0 V, V_I = V_{SS} Pull-up "on"	–30	–70	–140	µA
		V_{CC} = 3.0 V, V_I = V_{SS} Pull-up "on"	–6	–25	–45	µA
I_{IL}	"L" input current $\overline{RESET}$	V_I = V_{SS}			–5	µA
I_{IL}	"L" input current X_{IN}	V_I = V_{SS}		–4		µA

Note: When "1" is set to the port X_C switch bit (bit 4 of address $003B_{16}$) of the CPU mode register, the drive ability of Port $P7_1$ is different from the value above mentioned.

APPENDIX

3.1 Electrical characteristics

Table 3.1.6 Electrical characteristics

(Vcc = 2.5 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{RAM}	RAM hold voltage	When clock is stopped		2.0		5.5	V
I_{CC}	Power source current	High-speed mode, Vcc = 5 V f(X_{IN}) = 8 MHz f(X_{CIN}) = 32.768 kHz Output transistors “off”, A-D converter in operating			6.4	13	mA
		High-speed mode, Vcc = 5 V f(X_{IN}) = 8 MHz (in WIT state) f(X_{CIN}) = 32.768 kHz Output transistors “off”, A-D converter stopped			1.6	3.2	mA
		Low-speed mode, V_{CC} = 3 V, Ta ≤ 55 °C f(X_{IN}) = stopped f(X_{CIN}) = 32.768 kHz Output transistors “off”			15	22	µA
		Low-speed mode, V_{CC} = 3 V, Ta = 25 °C f(X_{IN}) = stopped f(X_{CIN}) = 32.768 kHz (in WIT state) Output transistors “off”			4.5	9.0	µA
		All oscillation stopped (in STP state)	Ta = 25 °C		0.1	1.0	µA
		Output transistors “off”	Ta = 85 °C			10	µA

3.1.4 A-D converter characteristics

Table 3.1.7 A-D converter characteristics

(Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –20 to 85°C, 4 MHz ≤ f(XIN) ≤ 8 MHz, in middle-speed/high-speed mode)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				10	Bits
—	Absolute accuracy (excluding quantization error)	VCC = VREF = 5.12 V		±1	±2.5	LSB
Tconv	Conversion time		61		62	tc(ϕ)
IVREF	Reference input current	VREF = 5 V	50	150	200	μA
IIA	Analog port input current			0.5	5.0	μA
RLADDER	Ladder resistor			35		kΩ

APPENDIX

3.1 Electrical characteristics

3.1.5 Timing requirements and switching characteristics

Table 3.1.8 Timing requirements 1

(Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{w(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{c(XIN)}$	Main clock input cycle time (XIN input)	125			ns
$t_{wH(XIN)}$	Main clock input “H” pulse width	45			ns
$t_{wL(XIN)}$	Main clock input “L” pulse width	40			ns
$t_{c(CNTR)}$	CNTR0, CNTR1 input cycle time	250			ns
$t_{wH(CNTR)}$	CNTR0, CNTR1 input “H” pulse width	105			ns
$t_{wL(CNTR)}$	CNTR0, CNTR1 input “L” pulse width	105			ns
$t_{wH(INT)}$	INT0–INT2 input “H” pulse width	80			ns
$t_{wL(INT)}$	INT0–INT2 input “L” pulse width	80			ns
$t_{c(SCLK)}$	Serial I/O clock input cycle time	800			ns
$t_{wH(SCLK)}$	Serial I/O clock input “H” pulse width	370			ns
$t_{wL(SCLK)}$	Serial I/O clock input “L” pulse width	370			ns
$t_{su(SIN-SCLK)}$	Serial I/O input setup time	220			ns
$t_{h(SCLK-SIN)}$	Serial I/O input hold time	100			ns

Table 3.1.9 Timing requirements 2

(Vcc = 2.5 to 4.0 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{w(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{c(XIN)}$	Main clock input cycle time (XIN input)	125			ns
$t_{wH(XIN)}$	Main clock input “H” pulse width	45			ns
$t_{wL(XIN)}$	Main clock input “L” pulse width	40			ns
$t_{c(CNTR)}$	CNTR0, CNTR1 input cycle time	500/(VCC–2)			ns
$t_{wH(CNTR)}$	CNTR0, CNTR1 input “H” pulse width	250/(VCC–2)–20			ns
$t_{wL(CNTR)}$	CNTR0, CNTR1 input “L” pulse width	250/(VCC–2)–20			ns
$t_{wH(INT)}$	INT0–INT2 input “H” pulse width	230			ns
$t_{wL(INT)}$	INT0–INT2 input “L” pulse width	230			ns
$t_{c(SCLK)}$	Serial I/O clock input cycle time	2000			ns
$t_{wH(SCLK)}$	Serial I/O clock input “H” pulse width	950			ns
$t_{wL(SCLK)}$	Serial I/O clock input “L” pulse width	950			ns
$t_{su(SIN-SCLK)}$	Serial I/O input setup time	400			ns
$t_{h(SCLK-SIN)}$	Serial I/O input hold time	200			ns

Table 3.1.10 Switching characteristics 1

(Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{wH}(SCLK)	Serial I/O clock output "H" pulse width	t_C(SCLK)/2–30			ns
t_{wL}(SCLK)	Serial I/O clock output "L" pulse width	t_C(SCLK)/2–30			ns
t_d(SCLK-SOUT)	Serial I/O output delay time **(Note 1)**			140	ns
t_v(SCLK-SOUT)	Serial I/O output valid time **(Note 1)**	–30			ns
t_r(SCLK)	Serial I/O clock output rising time			30	ns
t_f(SCLK)	Serial I/O clock output falling time			30	ns
t_r(CMOS)	CMOS output rising time **(Note 2)**		10	30	ns
t_f(CMOS)	CMOS output falling time **(Note 2)**		10	30	ns

Notes 1: When the P-channel output disable bit (bit 7 of address 0019_{16}) is "0."
2: The XOUT, XCOUT pins are excluded.

Table 3.1.11 Switching characteristics 2

(Vcc = 2.5 to 4.0 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{wH}(SCLK)	Serial I/O clock output "H" pulse width	t_C(SCLK)/2–50			ns
t_{wL}(SCLK)	Serial I/O clock output "L" pulse width	t_C(SCLK)/2–50			ns
t_d(SCLK-SOUT)	Serial I/O output delay time **(Note 1)**			350	ns
t_v(SCLK-SOUT)	Serial I/O output valid time **(Note 1)**	–30			ns
t_r(SCLK)	Serial I/O clock output rising time			50	ns
t_f(SCLK)	Serial I/O clock output falling time			50	ns
t_r(CMOS)	CMOS output rising time **(Note 2)**		20	50	ns
t_f(CMOS)	CMOS output falling time **(Note 2)**		20	50	ns

Notes 1: When the P-channel output disable bit (bit 7 of address 0019_{16}) is "0."
2: The XOUT, XCOUT pins are excluded.

APPENDIX

3.1 Electrical characteristics

3.1.6 Absolute maximum ratings (M version)

Table 3.1.12 Absolute maximum ratings (M 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 $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, $P8_0$–$P8_7$		–0.3 to V_{CC}+0.3	V
V_I	Input voltage V_{L1}		–0.3 to V_{L2}	V
V_I	Input voltage V_{L2}		V_{L1} to V_{L3}	V
V_I	Input voltage V_{L3}		V_{L2} to V_{CC}+0.3	V
V_I	Input voltage $\overline{RESET}$, X_{IN}		–0.3 to V_{CC}+0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$	At output port	–0.3 to V_{CC}+0.3	V
		At segment output	–0.3 to V_{L3}+0.3	V
V_O	Output voltage COM_0–COM_3		–0.3 to V_{L3}+0.3	V
V_O	Output voltage $P4_0$–$P4_7$, $P5_0$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, $P8_0$–$P8_7$		–0.3 to V_{CC}+0.3	V
V_O	Output voltage X_{OUT}		–0.3 to V_{CC}+0.3	V
P_d	Power dissipation	Ta = 25°C	300	mW
T_{opr}	Operating temperature		–20 to 85	°C
T_{stg}	Storage temperature		–40 to 125	°C

3.1.7 Recommended operating conditions (M version)

Table 3.1.13 Recommended operating conditions (M version)

(Vcc = 2.2 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage	High-speed mode f(X_{IN}) = 8 MHz	4.0	5.0	5.5	V
		Middle-speed mode f(X_{IN}) = 8 MHz	2.2	5.0	5.5	V
		Low-speed mode	2.2	5.0	5.5	V
V_{SS}	Power source voltage			0		V
V_{REF}	A-D converter reference voltage		2.0		V_{CC}	V
AV_{SS}	Analog power source voltage			0		V
V_{IA}	Analog input voltage AN_0–AN_7		AV_{SS}		V_{CC}	V

Table 3.1.14 Recommended operating conditions (M version)

(Vcc = 2.5 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{IH}	"H" input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$	0.7Vcc		Vcc	V
V_{IH}	"H" input voltage	$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (CM4 = 0)	0.8Vcc		Vcc	V
V_{IH}	"H" input voltage	$P8_0$–$P8_7$	0.4Vcc		Vcc	V
V_{IH}	"H" input voltage	$\overline{RESET}$	0.8Vcc		Vcc	V
V_{IH}	"H" input voltage	X_{IN}	0.8Vcc		Vcc	V
V_{IL}	"L" input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$	0		0.3Vcc	V
V_{IL}	"L" input voltage	$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (CM4 = 0)	0		0.2Vcc	V
V_{IL}	"L" input voltage	$P8_0$–$P8_7$	0		0.16Vcc	V
V_{IL}	"L" input voltage	$\overline{RESET}$	0		0.2Vcc	V
V_{IL}	"L" input voltage	X_{IN}	0		0.2Vcc	V

Table 3.1.15 Recommended operating conditions (M version)

(Vcc = 2.2 to 2.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{IH}	"H" input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$	0.8Vcc		Vcc	V
V_{IH}	"H" input voltage	$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (CM4 = 0)	0.95Vcc		Vcc	V
V_{IH}	"H" input voltage	$P8_0$–$P8_7$	0.5Vcc		Vcc	V
V_{IH}	"H" input voltage	$\overline{RESET}$	0.95Vcc		Vcc	V
V_{IH}	"H" input voltage	X_{IN}	0.95Vcc		Vcc	V
V_{IL}	"L" input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$	0		0.2Vcc	V
V_{IL}	"L" input voltage	$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (CM4 = 0)	0		0.05Vcc	V
V_{IL}	"L" input voltage	$P8_0$–$P8_7$	0		0.1Vcc	V
V_{IL}	"L" input voltage	$\overline{RESET}$	0		0.05Vcc	V
V_{IL}	"L" input voltage	X_{IN}	0		0.05Vcc	V

APPENDIX

3.1 Electrical characteristics

Table 3.1.16 Recommended operating conditions (M version)

(Vcc = 2.2 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$\Sigma I_{OH(peak)}$	"H" total peak output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ $P8_0$–$P8_7$, $P5_0$			–60	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current **(Note 1)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			–30	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			40	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current **(Note 1)** $P8_0$–$P8_7$, $P5_0$			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current **(Note 1)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			40	mA
$\Sigma I_{OH(avg)}$	"H" total average output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ $P8_0$–$P8_7$, $P5_0$			–30	mA
$\Sigma I_{OH(avg)}$	"H" total average output current **(Note 1)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			–15	mA
$\Sigma I_{OL(avg)}$	"L" total average output current **(Note 1)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			20	mA
$\Sigma I_{OL(avg)}$	"L" total average output current **(Note 1)** $P8_0$–$P8_7$, $P5_0$			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current **(Note 1)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			20	mA
$I_{OH(peak)}$	"H" peak output current **(Note 2)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			–4.0	mA
$I_{OH(peak)}$	"H" peak output current **(Note 2)** $P4_0$–$P4_7$, $P5_0$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ $P8_0$–$P8_7$			–10	mA
$I_{OL(peak)}$	"L" peak output current **(Note 2)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			5.0	mA
$I_{OL(peak)}$	"L" peak output current **(Note 2)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			10	mA
$I_{OL(peak)}$	"L" peak output current **(Note 2)** $P8_0$–$P8_7$, $P5_0$			30	mA
$I_{OH(avg)}$	"H" average output current **(Note 3)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			–2.0	mA
$I_{OH(avg)}$	"H" average output current **(Note 3)** $P4_0$–$P4_7$, $P5_0$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ $P8_0$–$P8_7$			–5.0	mA
$I_{OL(avg)}$	"L" average output current **(Note 3)** $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$			2.5	mA
$I_{OL(avg)}$	"L" average output current **(Note 3)** $P4_0$–$P4_7$, $P5_2$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$			5.0	mA
$I_{OL(avg)}$	"L" average output current **(Note 3)** $P8_0$–$P8_7$, $P5_0$			15	mA

Notes 1: The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100 ms. The total peak current is the peak value of all the currents.

2: The peak output current is the peak current flowing in each port.

3: The average output current is average value measured over 100 ms.

Table 3.1.17 Recommended operating conditions (M version)

(Vcc = 2.2 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
f(CNTR0)	Input frequency (duty cycle 50%)	(4.0 V ≤ Vcc ≤ 5.5 V)			4.0	MHz
f(CNTR1)		(2.2 V ≤ Vcc ≤ 4.0 V)			(2×Vcc)–4	MHz
f(XIN)	Main clock input oscillation frequency **(Note 4)**	High-speed mode (4.0 V ≤ Vcc ≤ 5.5 V)			8.0	MHz
		High-speed mode (2.2 V ≤ Vcc ≤ 4.0 V)			(4×Vcc)–8	MHz
		Middle-speed mode			8.0	MHz
f(XCIN)	Sub-clock input oscillation frequency **(Notes 4, 5)**			32.768	50	kHz

Notes 4: When the oscillation frequency has a duty cycle of 50%.

5: When using the microcomputer in low-speed mode, set the sub-clock input oscillation frequency on condition that f(XCIN) < f(XIN)/3.

APPENDIX

3.1 Electrical characteristics

3.1.8 Electrical characteristics (M version)

Table 3.1.18 Electrical characteristics (M version)

(V_{CC} = 4.0 to 5.5 V, Ta = –20 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$_7$	I_{OH} = –2.0 mA	V_{CC}–2.0			V
		I_{OH} = –0.6 mA V_{CC} = 2.5 V	V_{CC}–1.0			V
V_{OH}	"H" output voltage P4$_0$–P4$_7$, P5$_0$, P5$_2$–P5$_7$, P6$_0$–P6$_7$, P7$_0$, P7$_1$, P8$_0$–P8$_7$ **(Note)**	I_{OH} = –5 mA	V_{CC}–2.0			V
		I_{OH} = –1.25 mA	V_{CC}–0.5			V
		I_{OH} = –1.25 mA V_{CC} = 2.5 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$_7$	I_{OL} = 2.5 mA			2.0	V
		I_{OL} = 1.25 mA			0.5	V
		I_{OL} = 1.25 mA V_{CC} = 2.5 V			1.0	V
V_{OL}	"L" output voltage P4$_0$–P4$_7$, P5$_2$–P5$_7$, P6$_0$–P6$_7$, P7$_0$, P7$_1$ **(Note)**	I_{OL} = 5.0 mA			2.0	V
		I_{OL} = 2.5 mA			0.5	V
		I_{OL} = 2.5 mA V_{CC} = 2.5 V			1.0	V
V_{OL}	"L" output voltage P8$_0$–P8$_7$, P5$_0$	I_{OL} = 15 mA			2.0	V
V_{T+}–V_{T-}	Hysteresis INT$_0$–INT$_2$, CNTR$_0$, CNTR$_1$, P8$_0$–P8$_7$			0.5		V
V_{T+}–V_{T-}	Hysteresis SCLK$_1$, SIN			0.5		V
V_{T+}–V_{T-}	Hysteresis $\overline{\text{RESET}}$	$\overline{\text{RESET}}$: V_{CC} = 2.2 V – 5.5 V		0.5		V
I_{IH}	"H" input current P0$_0$–P0$_7$, P1$_0$–P1$_7$, P2$_0$–P2$_7$	V_I = V_{CC} Pull-down "off"			5.0	µA
		V_{CC} = 5.0 V, V_I = V_{CC} Pull-down "on"	30	70	140	µA
		V_{CC} = 3.0 V, V_I = V_{CC} Pull-down "on"	6.0	25	45	µA
I_{IH}	"H" input current P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$, P7$_1$, P8$_0$–P8$_7$	V_I = V_{CC}			5.0	µA
I_{IH}	"H" input current $\overline{\text{RESET}}$	V_I = V_{CC}			5.0	µA
I_{IH}	"H" input current XIN	V_I = V_{CC}		4.0		µA
I_{IL}	"L" input current P0$_0$–P0$_7$, P1$_0$–P1$_7$, P2$_0$–P2$_7$, P5$_1$				–5.0	µA
I_{IL}	"L" input current P4$_0$–P4$_7$, P5$_0$, P5$_2$–P5$_7$, P6$_0$–P6$_7$, P7$_0$, P7$_1$, P8$_0$–P8$_7$	V_I = V_{SS} Pull-up "off"			–5.0	µA
		V_{CC} = 5.0 V, V_I = V_{SS} Pull-up "on"	–30	–70	–140	µA
		V_{CC} = 3.0 V, V_I = V_{SS} Pull-up "on"	–6	–25	–45	µA
I_{IL}	"L" input current $\overline{\text{RESET}}$	V_I = V_{SS}			–5	µA
I_{IL}	"L" input current XIN	V_I = V_{SS}		–4		µA

Note: When "1" is set to the port XC switch bit (bit 4 of address 003B$_{16}$) of the CPU mode register, the drive ability of Port P7$_1$ is different from the value above mentioned.

Table 3.1.19 Electrical characteristics (M version)

(Vcc = 2.2 to 5.5 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{RAM}	RAM hold voltage	When clock is stopped		2.0		5.5	V
I_{CC}	Power source current	High-speed mode, Vcc = 5 V f(X_{IN}) = 8 MHz f(X_{CIN}) = 32.768 kHz Output transistors “off”, A-D converter in operating			6.4	13	mA
		High-speed mode, Vcc = 5 V f(X_{IN}) = 8 MHz (in WIT state) f(X_{CIN}) = 32.768 kHz Output transistors “off”, A-D converter stopped			1.6	3.2	mA
		Low-speed mode, V_{CC} = 3 V, Ta ≤ 55 °C f(X_{IN}) = stopped f(X_{CIN}) = 32.768 kHz Output transistors “off”			15	22	µA
		Low-speed mode, V_{CC} = 3 V, Ta = 25 °C f(X_{IN}) = stopped f(X_{CIN}) = 32.768 kHz (in WIT state) Output transistors “off”			4.5	9.0	µA
		All oscillation stopped (in STP state)	Ta = 25 °C		0.1	1.0	µA
		Output transistors “off”	Ta = 85 °C			10	µA

APPENDIX

3.1 Electrical characteristics

3.1.9 A-D converter characteristics (M version)

Table 3.1.20 A-D converter characteristics (M version)

(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, Ta = –20 to 85°C, 4 MHz ≤ $f(X_{IN})$ ≤ 8 MHz, in middle-speed/high-speed mode)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				10	Bits
—	Absolute accuracy (excluding quantization error)	V_{CC} = V_{REF} = 5.12 V		±1	±2.5	LSB
T_{conv}	Conversion time		61		62	tc(ϕ)
IV_{REF}	Reference input current	V_{REF} = 5 V	50	150	200	μA
I_{IA}	Analog port input current			0.5	5.0	μA
R_{LADDER}	Ladder resistor			35		kΩ

3.1.10 Timing requirements and switching characteristics (M version)

Table 3.1.21 Timing requirements 1 (M version)

(Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{w(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{c(XIN)}$	Main clock input cycle time (XIN input)	125			ns
$t_{wH(XIN)}$	Main clock input “H” pulse width	45			ns
$t_{wL(XIN)}$	Main clock input “L” pulse width	40			ns
$t_{c(CNTR)}$	CNTR0, CNTR1 input cycle time	250			ns
$t_{wH(CNTR)}$	CNTR0, CNTR1 input “H” pulse width	105			ns
$t_{wL(CNTR)}$	CNTR0, CNTR1 input “L” pulse width	105			ns
$t_{wH(INT)}$	INT0–INT2 input “H” pulse width	80			ns
$t_{wL(INT)}$	INT0–INT2 input “L” pulse width	80			ns
$t_{c(SCLK)}$	Serial I/O clock input cycle time	800			ns
$t_{wH(SCLK)}$	Serial I/O clock input “H” pulse width	370			ns
$t_{wL(SCLK)}$	Serial I/O clock input “L” pulse width	370			ns
$t_{su(SIN-SCLK)}$	Serial I/O input setup time	220			ns
$t_{h(SCLK-SIN)}$	Serial I/O input hold time	100			ns

Table 3.1.22 Timing requirements 2 (M version)

(Vcc = 2.2 to 4.0 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{w(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{c(XIN)}$	Main clock input cycle time (XIN input)	125			ns
$t_{wH(XIN)}$	Main clock input “H” pulse width	45			ns
$t_{wL(XIN)}$	Main clock input “L” pulse width	40			ns
$t_{c(CNTR)}$	CNTR0, CNTR1 input cycle time	500/(VCC–2)			ns
$t_{wH(CNTR)}$	CNTR0, CNTR1 input “H” pulse width	250/(VCC–2)–20			ns
$t_{wL(CNTR)}$	CNTR0, CNTR1 input “L” pulse width	250/(VCC–2)–20			ns
$t_{wH(INT)}$	INT0–INT2 input “H” pulse width	230			ns
$t_{wL(INT)}$	INT0–INT2 input “L” pulse width	230			ns
$t_{c(SCLK)}$	Serial I/O clock input cycle time	2000			ns
$t_{wH(SCLK)}$	Serial I/O clock input “H” pulse width	950			ns
$t_{wL(SCLK)}$	Serial I/O clock input “L” pulse width	950			ns
$t_{su(SIN-SCLK)}$	Serial I/O input setup time	400			ns
$t_{h(SCLK-SIN)}$	Serial I/O input hold time	200			ns

APPENDIX

3.1 Electrical characteristics

Table 3.1.23 Switching characteristics 1 (M version)

(Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
t_{wH}(SCLK)	Serial I/O clock output "H" pulse width		t_C(SCLK)/2–30			ns
t_{wL}(SCLK)	Serial I/O clock output "L" pulse width		t_C(SCLK)/2–30			ns
t_d(SCLK-SOUT)	Serial I/O output delay time	**(Note 1)**			140	ns
t_v(SCLK-SOUT)	Serial I/O output valid time	**(Note 1)**	–30			ns
t_r(SCLK)	Serial I/O clock output rising time				30	ns
t_f(SCLK)	Serial I/O clock output falling time				30	ns
t_r(CMOS)	CMOS output rising time	**(Note 2)**		10	30	ns
t_f(CMOS)	CMOS output falling time	**(Note 2)**		10	30	ns

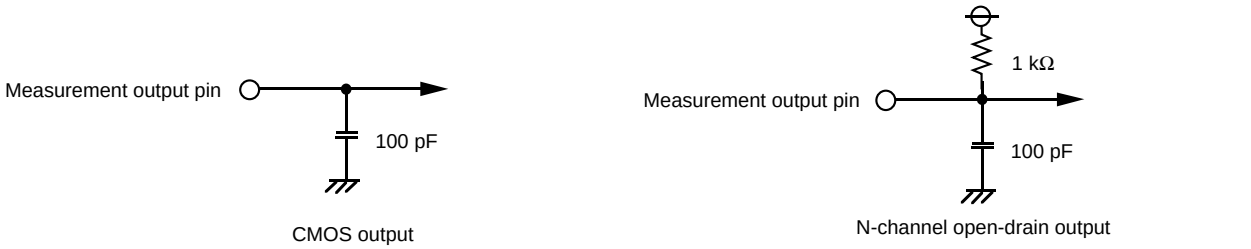
Notes 1: When the P-channel output disable bit (bit 7 of address 0019_{16}) is "0."
2: The XOUT, XCOUT pins are excluded.

Table 3.1.24 Switching characteristics 2 (M version)

(Vcc = 2.2 to 4.0 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
t_{wH}(SCLK)	Serial I/O clock output "H" pulse width		t_C(SCLK)/2–50			ns
t_{wL}(SCLK)	Serial I/O clock output "L" pulse width		t_C(SCLK)/2–50			ns
t_d(SCLK-SOUT)	Serial I/O output delay time	**(Note 1)**			350	ns
t_v(SCLK-SOUT)	Serial I/O output valid time	**(Note 1)**	–30			ns
t_r(SCLK)	Serial I/O clock output rising time				50	ns
t_f(SCLK)	Serial I/O clock output falling time				50	ns
t_r(CMOS)	CMOS output rising time	**(Note 2)**		20	50	ns
t_f(CMOS)	CMOS output falling time	**(Note 2)**		20	50	ns

Notes 1: When the P-channel output disable bit (bit 7 of address 0019_{16}) is "0."
2: The XOUT, XCOUT pins are excluded.



Note: When bit 7 of the serial I/O control register 1 (address 0019_{16}) is " 1."
(N-channel open-drain output mode)

Fig. 3.1.1 Circuit for measuring output switching characteristics

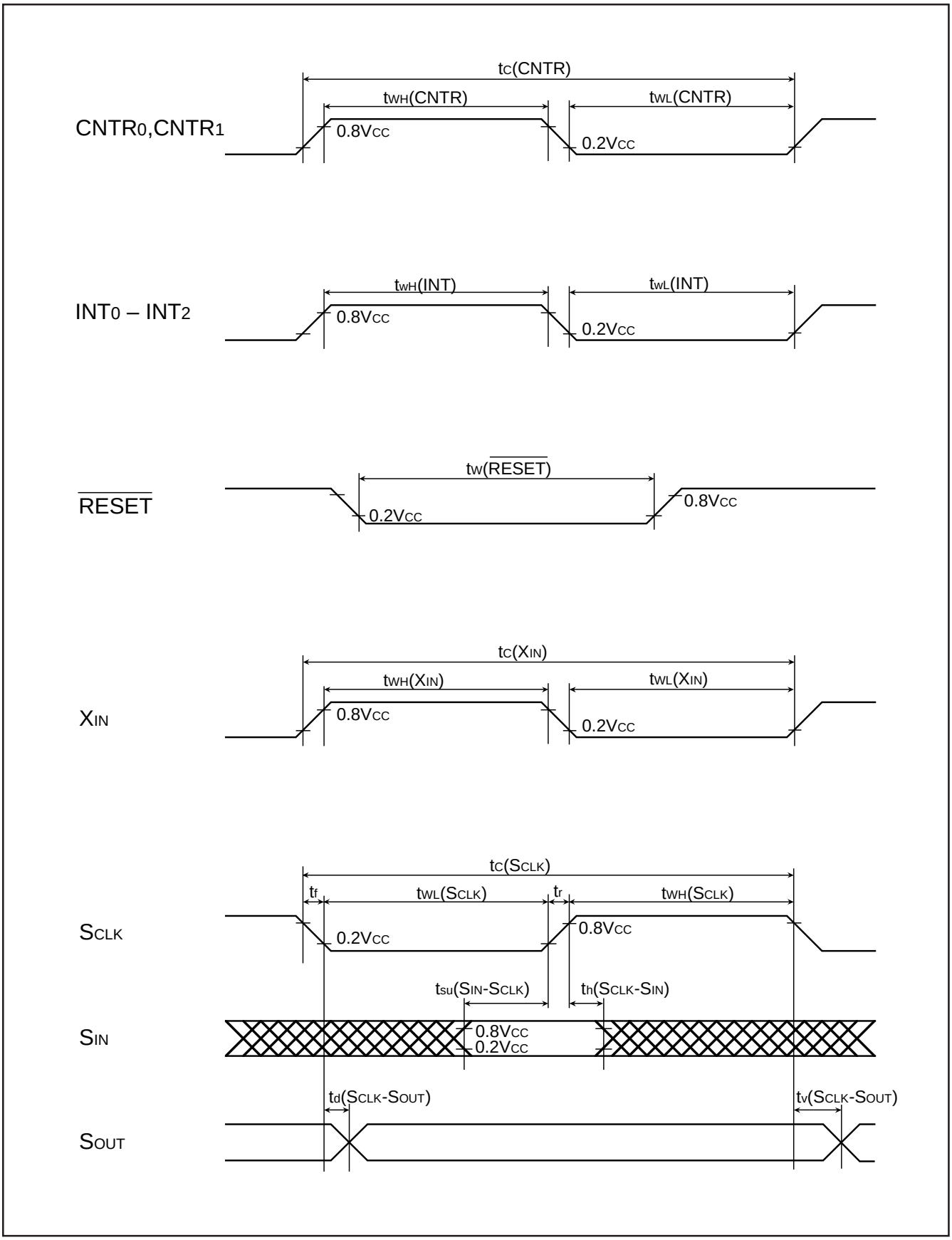


Fig. 3.1.2 Timing chart

3.2 Standard characteristics

3.2.1 Power source current standard characteristics

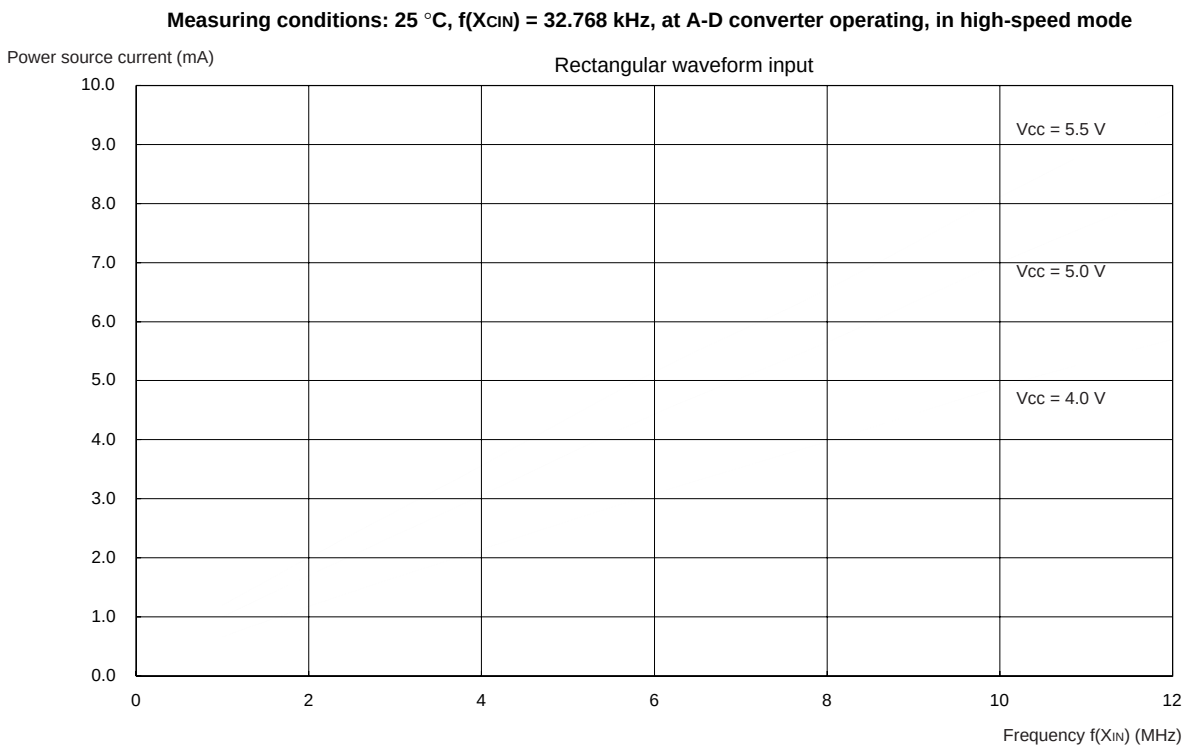


Fig. 3.2.1 Power source current standard characteristics

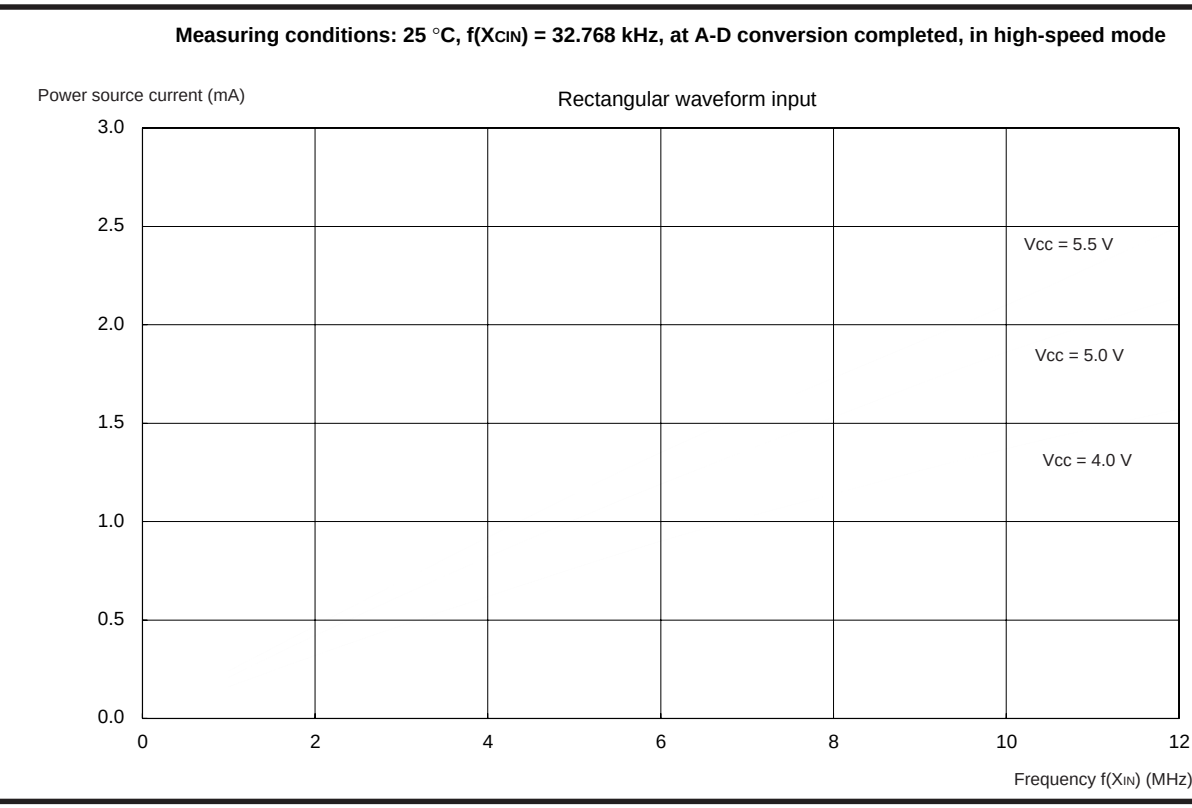


Fig. 3.2.2 Power source current standard characteristics (in wait mode)

3.2.2 Port standard characteristics

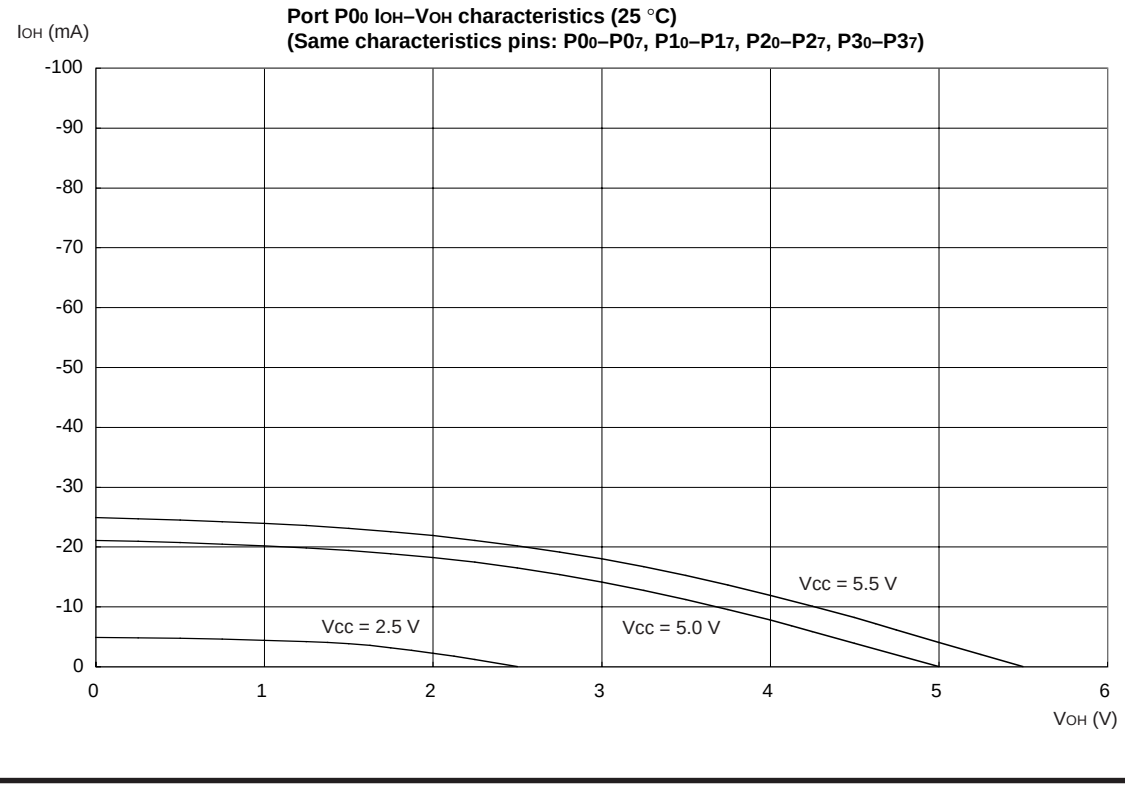


Fig. 3.2.3 CMOS output port (P0, P1, P2, P3) P-channel side characteristics (25 °C)

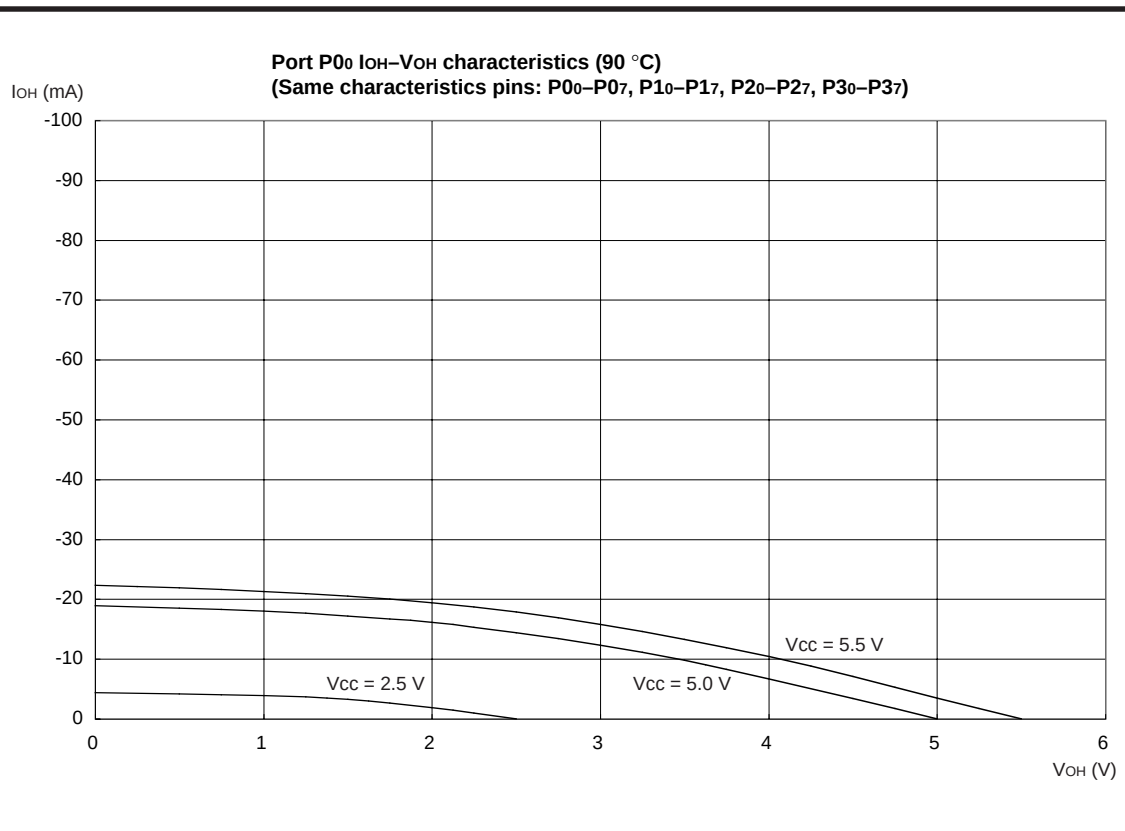


Fig. 3.2.4 CMOS output port (P0, P1, P2, P3) P-channel side characteristics (90 °C)

APPENDIX

3.2 Standard characteristics

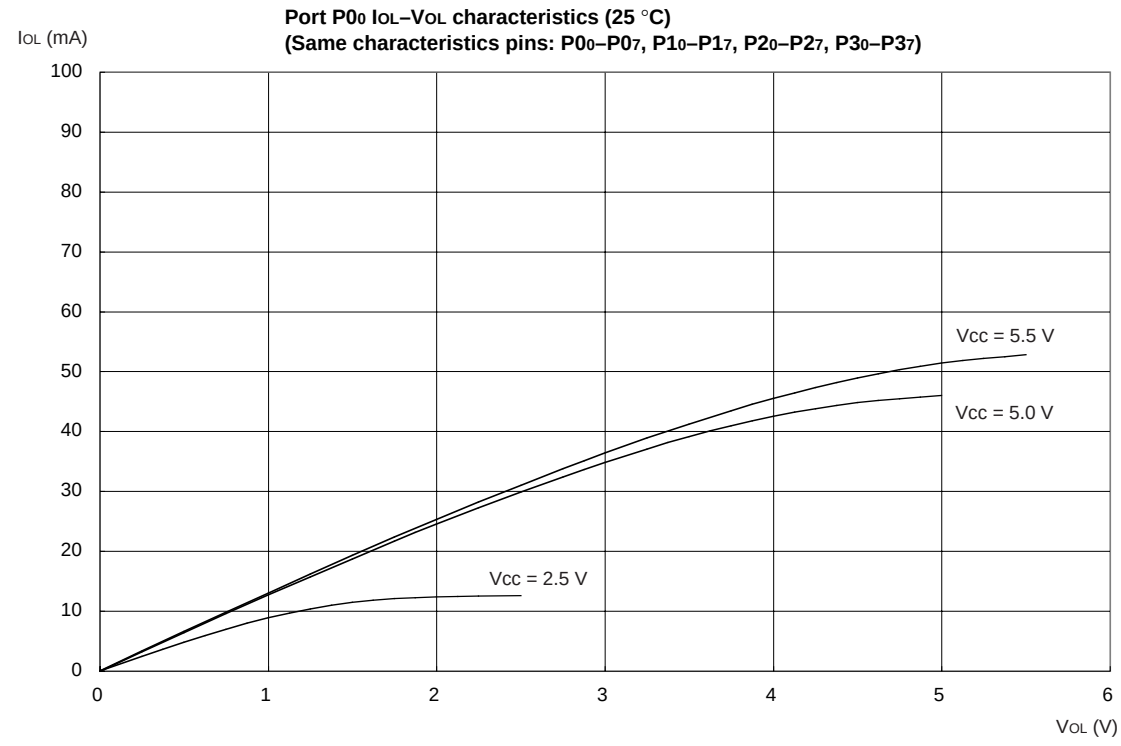


Fig. 3.2.5 CMOS output port (P0, P1, P2, P3) P-channel side characteristics (25 °C)

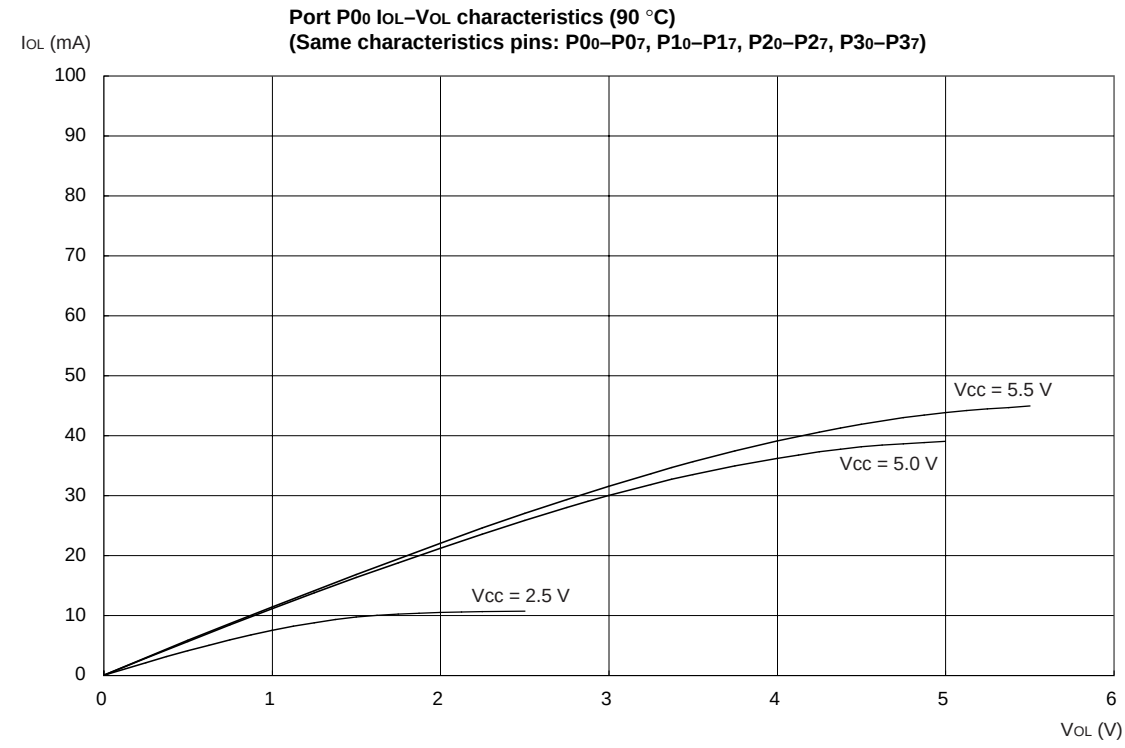


Fig. 3.2.6 CMOS output port (P0, P1, P2, P3) N-channel side characteristics (90 °C)

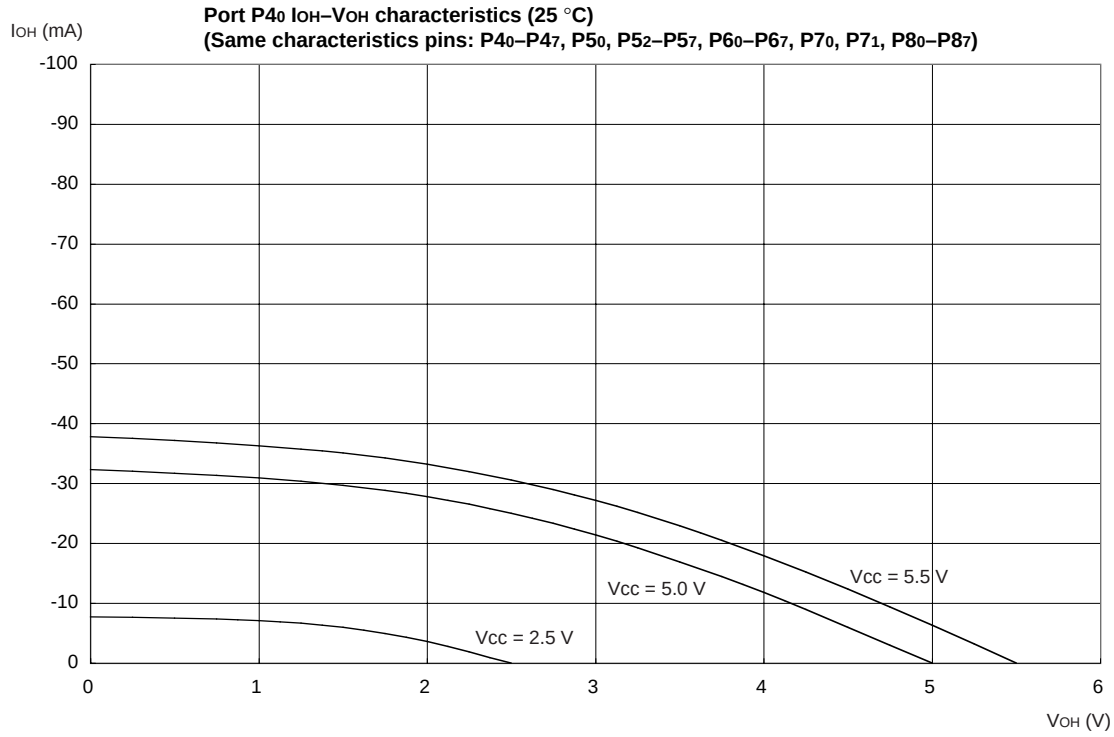


Fig. 3.2.7 CMOS output port (P4, $P5_0$, $P5_2$–$P5_7$, P6, $P7_0$, $P7_1$, P8) P-channel side characteristics (25 °C)

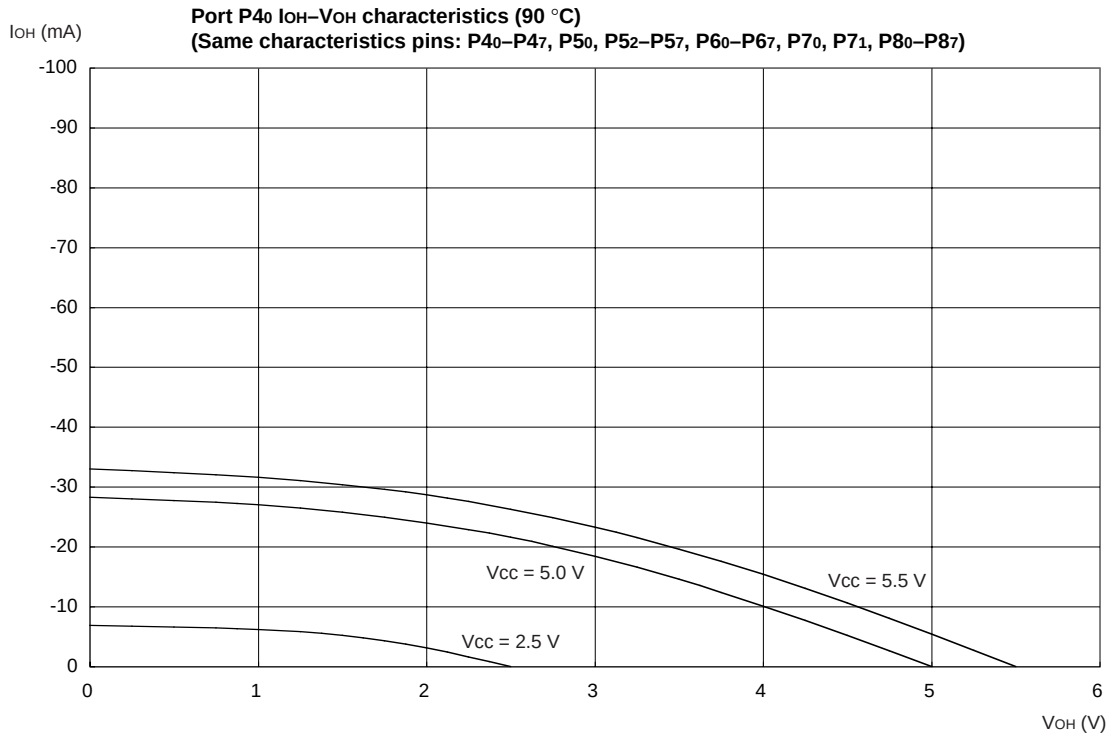


Fig. 3.2.8 CMOS output port (P4, $P5_0$, $P5_2$–$P5_7$, P6, $P7_0$, $P7_1$, P8) P-channel side characteristics (90 °C)

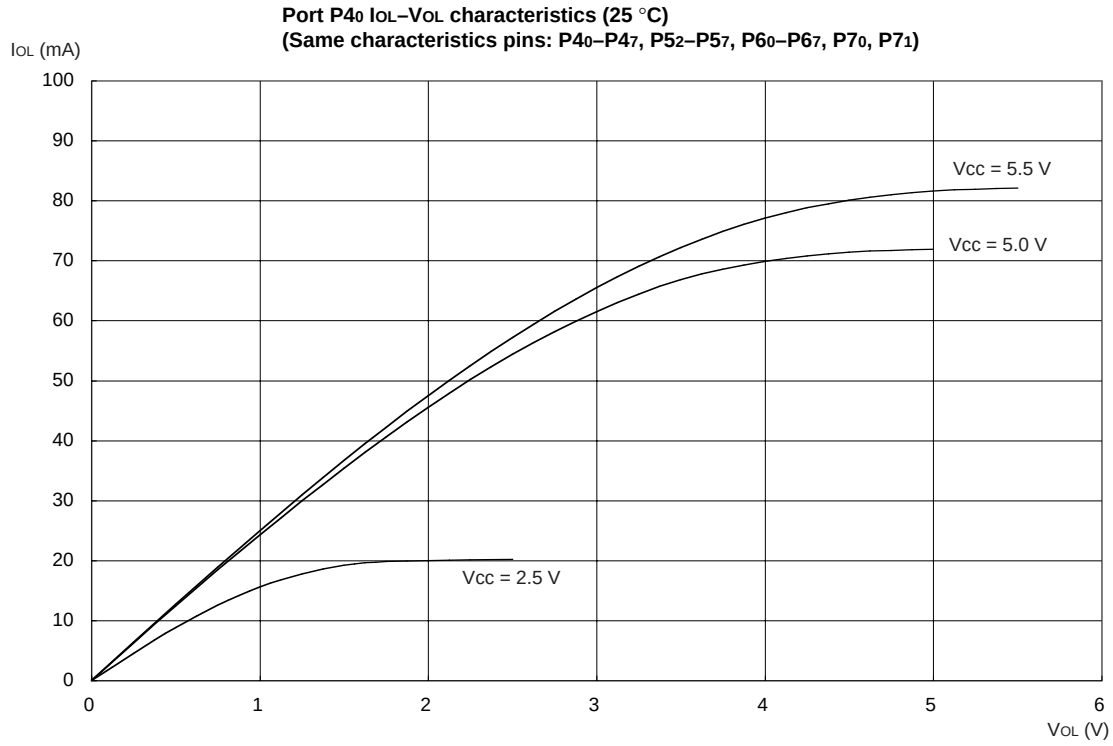


Fig. 3.2.9 CMOS output port (P4, $P5_2$–$P5_7$, P6, $P7_0$, $P7_1$) N-channel side characteristics (25 °C)

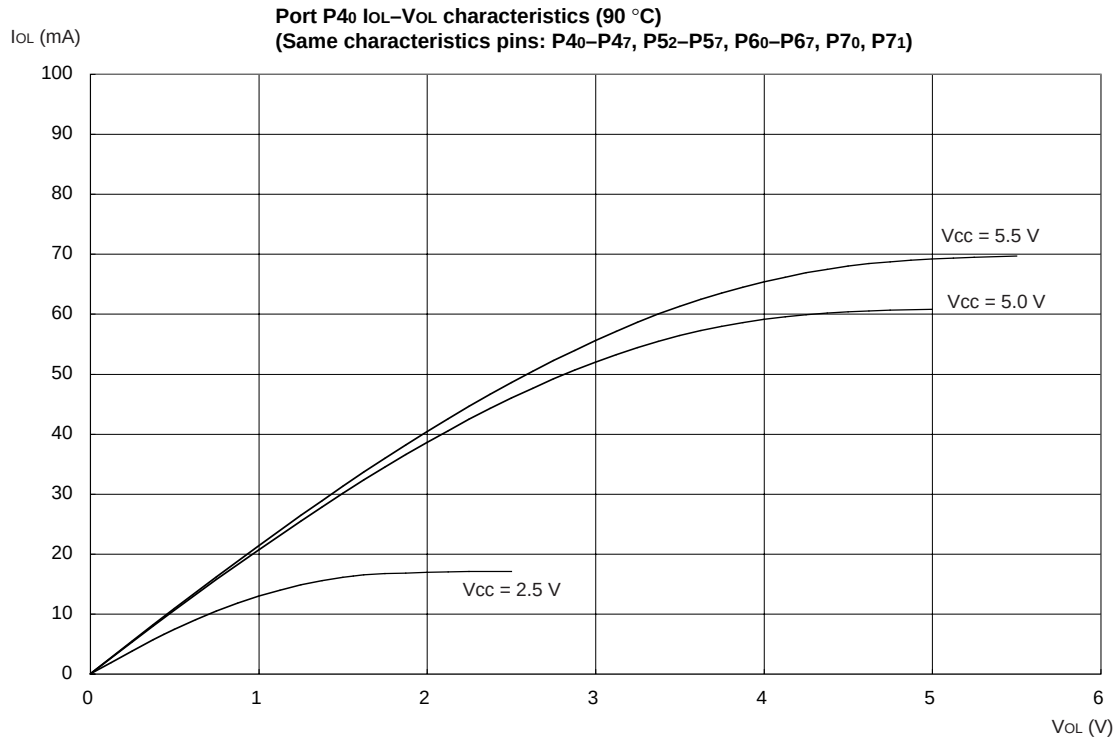


Fig. 3.2.10 CMOS output port (P4, $P5_2$–$P5_7$, P6, $P7_0$, $P7_1$) N-channel side characteristics (90 °C)

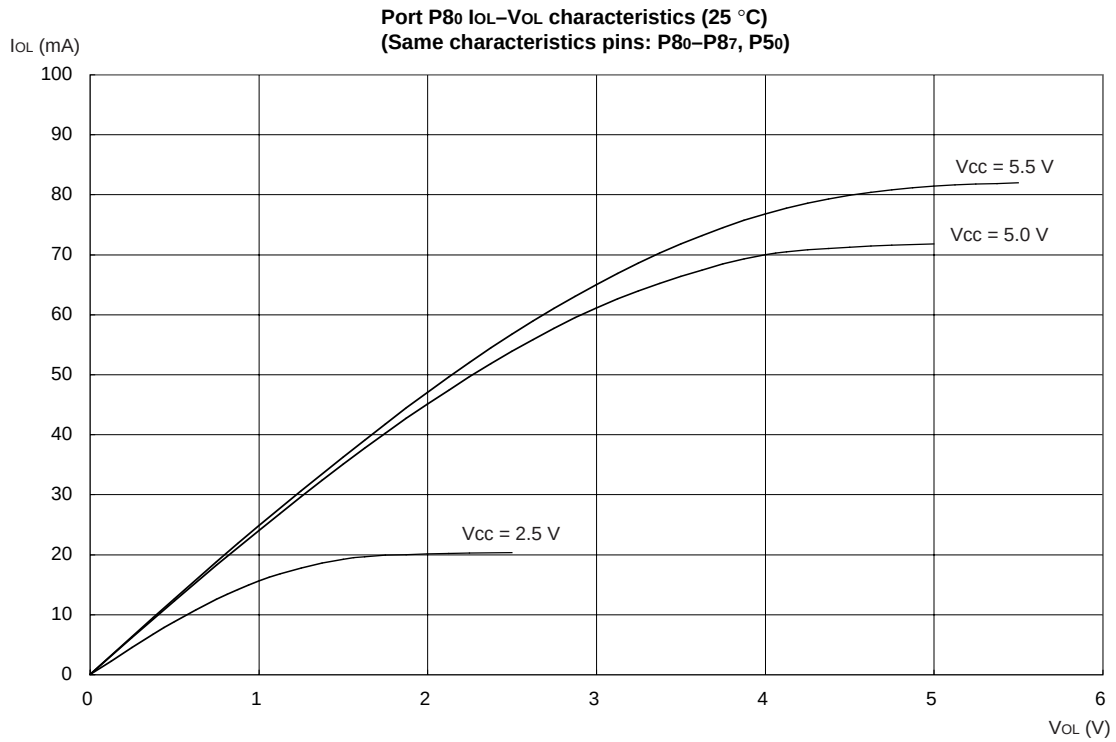


Fig. 3.2.11 CMOS output port ($P5_0$, P8) N-channel side characteristics (25 °C)

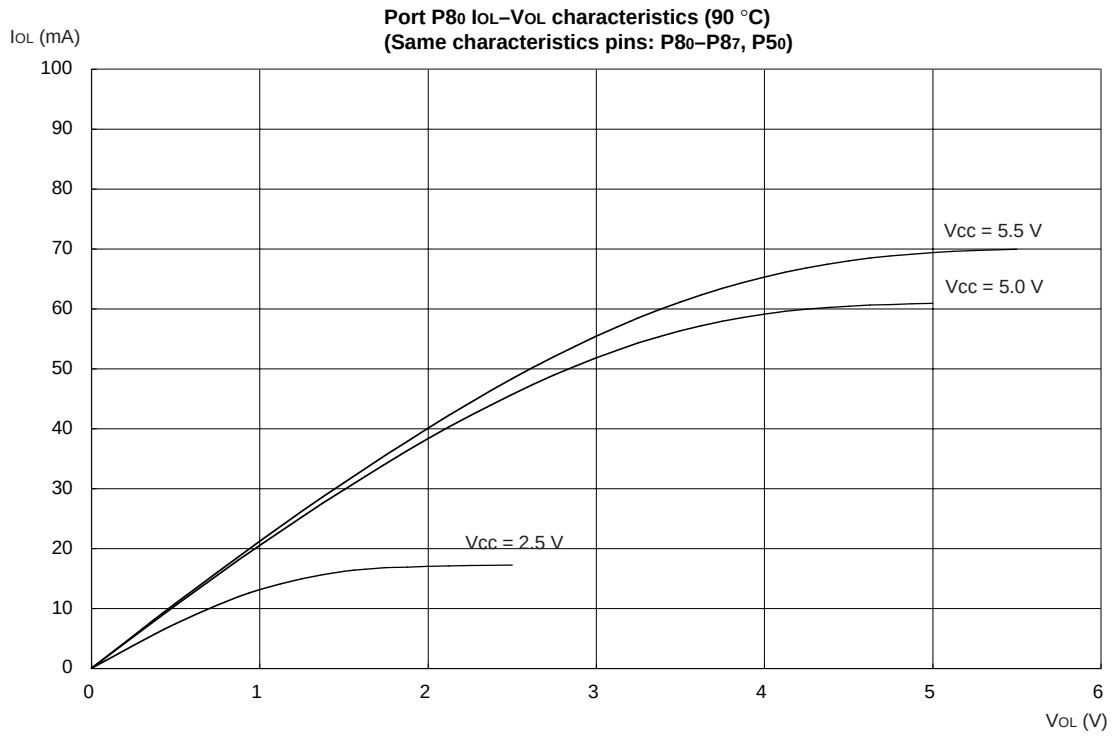


Fig. 3.2.12 CMOS output port ($P5_0$, P8) N-channel side characteristics (90 °C)

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.

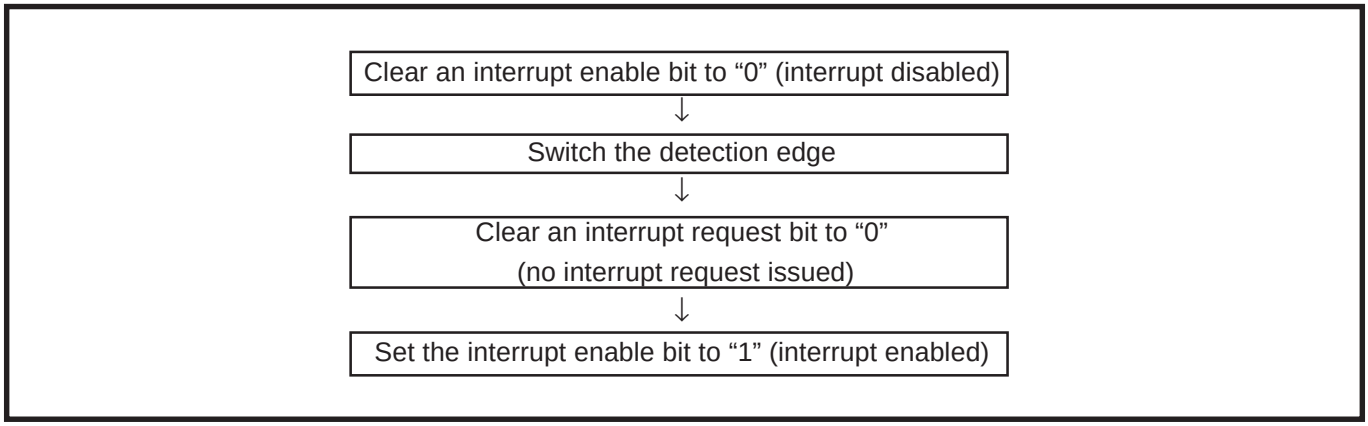


Fig. 3.3.1 Sequence of switch detection edge

■ **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.

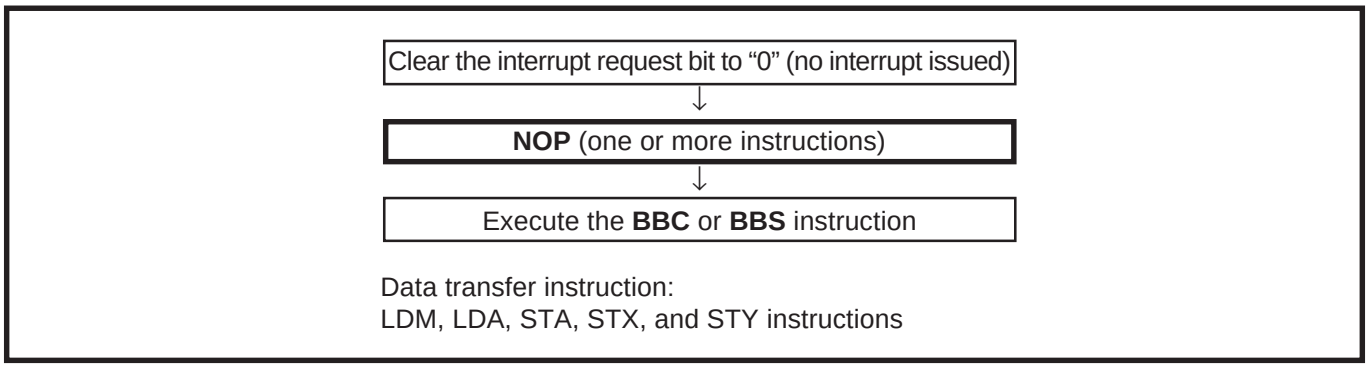


Fig. 3.3.2 Sequence of check of interrupt request bit

(3) Structure of interrupt control register 2

Fix the bit 7 of the interrupt control register 2 to “0”. Figure 3.3.3 shows the structure of the interrupt control register 2.

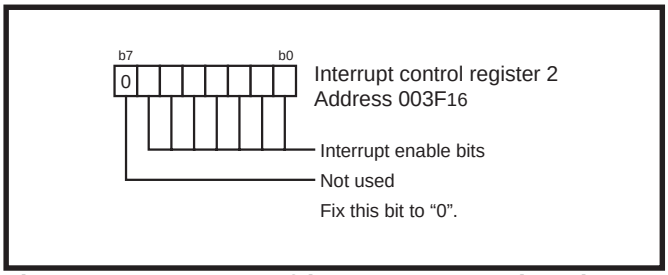


Fig. 3.3.3 Structure of interrupt control register 2

3.3.2 Notes on timer A (PWM mode and IGBT output mode)

(1) When timer starts first or last value of compare register is “0000_{16}”

After “L” level (timer A output active edge switch bit is “0”; when starting from “L” output) is output during 2 cycles (until timer underflows two times), start PWM output or IGBT output.

Reason: When data is written to timer A and compare register, value of timer A and value of compare register are renewed at timer underflow. In case of this, compare register value and timer value are compared before renewal so that they are judged to be equal, and TA_{OUT} output becomes “L”. (Timer A output switch bit = “0”: when starting from “L” output) Timer A underflow should be “H” output, but the match have the priority. (see **“Figure 3.3.4”**)

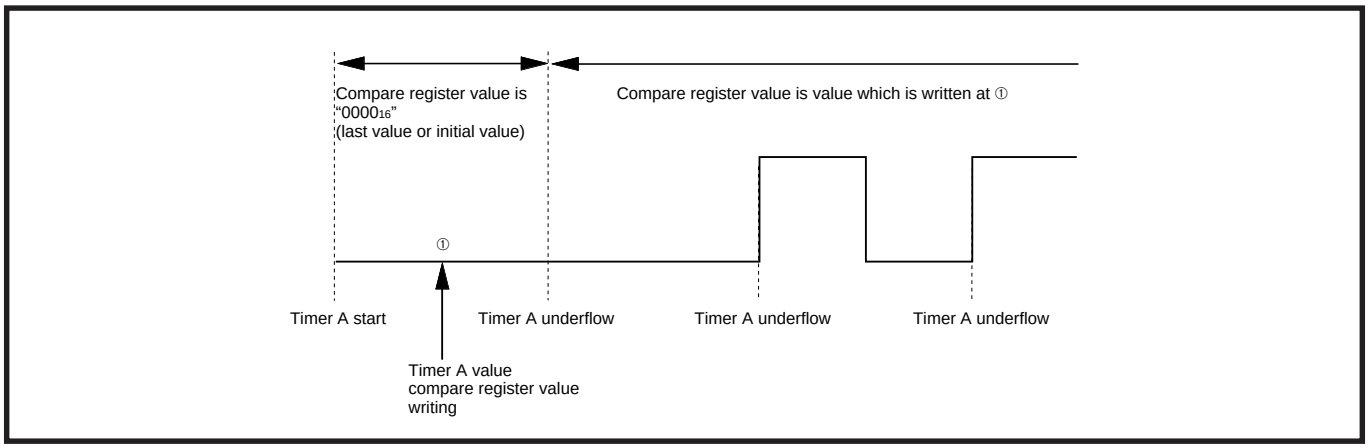


Fig. 3.3.4 PWM output and IGBT output (1)

3.3 Notes on use

(2) When compare register is set to "0000_{16}" (last value is except "0000_{16}")

Next 1 cycle of the cycle which data is written to timer A and compare register is output "H", and "L" is output from the next cycle. (timer A output switch bit = "0": when starting from "L" output) (see **"Figure 3.3.5"**)

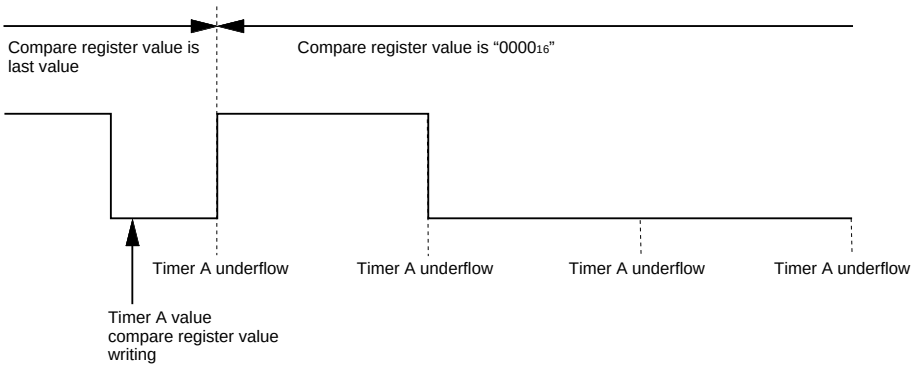


Fig. 3.3.5 PWM output and IGBT output (2)

(3) When timer A and compare register are same value

TA_{OUT} output becomes "H" with underflow immediately after data is written to timer A and compare register. And TA_{OUT} output becomes "L" when timer A is reloaded and the value matches with compare register. This "H" output width becomes 1 count of timer A count source. (timer A output switch bit ="0": when starting from "L" output) (see **"Figure 3.3.6"**)

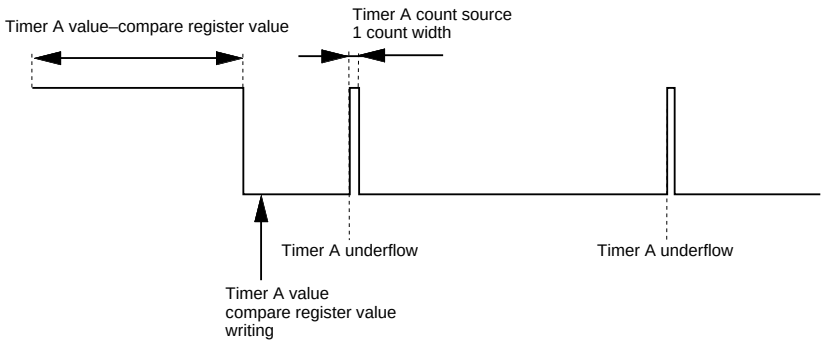


Fig. 3.3.6 PWM output and IGBT output (3)

3.3.3 Notes on serial I/O

(1) Selecting external synchronous clock

When an external synchronous clock is selected, the contents of serial I/O register are being shifted continually while the transfer clock is input to the serial I/O1 clock pin. In this case, control the clock externally.

(2) Transmission data writing

When an external clock is used as the synchronous clock, write the transmit data to the serial I/O shift register at "H" level of transfer clock input.

3.3.4 Notes on LCD controller

●When switching from the high-speed or middle-speed mode to the low-speed mode, switch the mode in the following order:

(1) 32 kHz oscillation selected (bit 4 of CPU mode register (address $003B_{16}$) = "1")
(2) Count source for LCDCK = f(X_{CIN})/32 (bit 7 of LCD mode register (address 0039_{16}) = "0")
(3) Internal system clock: X_{CIN}-X_{COUT} selected (bit 7 of CPU mode register (address $003B_{16}$) = "1")
(4) Main clock X_{IN}–X_{OUT} stopped (bit 5 of CPU mode register (address $003B_{16}$) = "1")

Execute the setting (2) after the oscillation at 32 kHz (setting (1)) becomes completely stable.

●If the STP instruction is executed while the LCD is turned on by setting bit 3 of the LCD mode register (address 0039_{16}) to "1", a DC voltage is applied to the LCD. For this reason, do not execute the STP instruction while the LCD is lighting.

●When the LCD is not used, open the segment and the common pins.
Connect V_{L1} to V_{L3} to V_{SS}.

●For the following products, if the LCD enable bit of the LCD mode register (bit 3 of address 0039_{16}) is set to "0", all LCDs cannot be turned off. When all LCDs are turned off, set "0" (turn off) to all corresponding LCD display RAM.

Corresponding products: M38C34M6AXXXFP, M38C34M6MXXXFP, M38C37ECAXXXFP, M38C37ECMXXXFP, M38C37ECAFP, M38C37ECMFP, M38C37ECAFS, M38C3ECMFS, M38C37RFS, M38C37RMFS

3.3 Notes on use

3.3.5 Notes on A-D converter

(1) Analog input pin

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

● **Reason**

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

(2) A-D converter power source pin

The AVSS pin is A-D converter power source pin. Regardless of using the A-D conversion function or not, connect it as following :

• AVSS : Connect to the VSS line.

● **Reason**

If the AVSS pin is opened, the microcomputer may have a failure because of noise or others.

(3) Clock frequency during A-D conversion

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

• f(XIN) is 500 kHz or more.
• Use clock divided by main clock (f(XIN)) as internal system clock.
• Do not execute the **STP** instruction and **WIT** instruction.

3.3.6 Notes on reset circuit

(1) Reset input voltage control

Make sure that the reset input voltage is 0.5 V or less for Vcc of 2.5 V **(Note)**.
Perform switch to the high-speed mode when power source voltage is within 4.0 to 5.5 V.
Note: M version of mask ROM version is 2.2 V.

(2) Countermeasure when RESET signal rise time is long

In case where the 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.

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) Package

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

● **Reason**

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

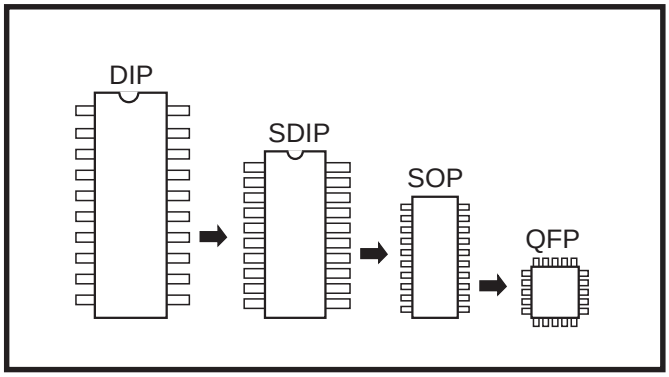


Fig. 3.4.1 Selection of packages

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

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

● **Reason**

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

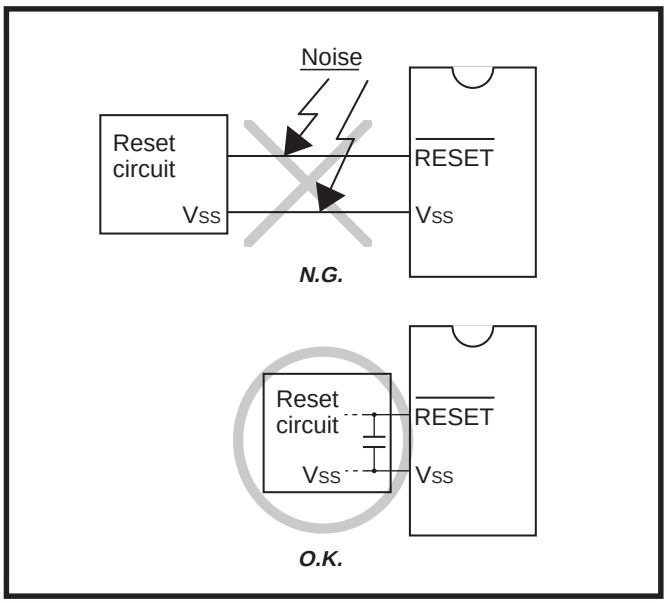


Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ pin

3.4 Countermeasures against noise

(3) Wiring for clock input/output pins

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

● **Reason**

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

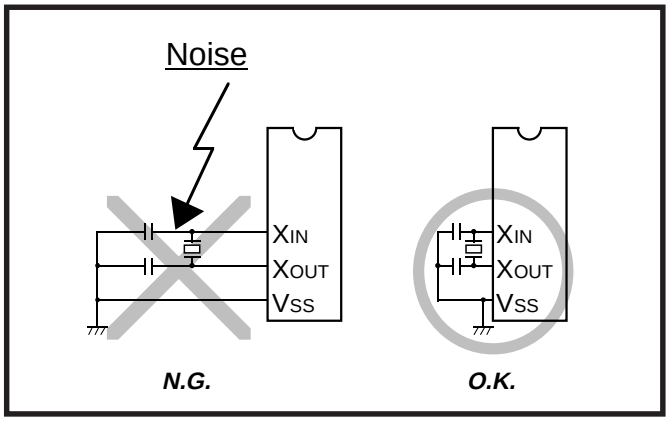


Fig. 3.4.3 Wiring for clock I/O pins

(4) 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 between the V_{PP} pin and the V_{SS} 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.

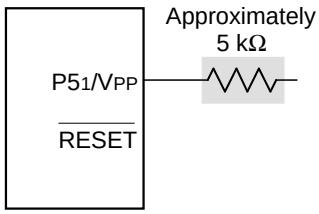


Fig. 3.4.4 Wiring for the V_{PP} pin of the One Time PROM and the EPROM version

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.

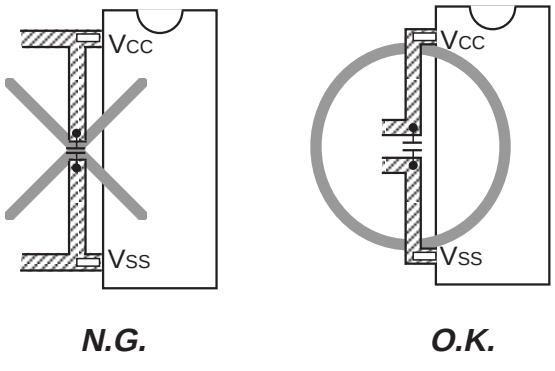


Fig. 3.4.5 Bypass capacitor across the V_{SS} line and the V_{CC} line

3.4 Countermeasures against noise

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 V_{SS} pin and the analog input pin. Besides, connect the capacitor to the V_{SS} pin as close as possible. Also, connect the capacitor across the analog input pin and the V_{SS} 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 V_{SS} pin is grounded at a position far away from the V_{SS} pin, noise on the GND line may enter a microcomputer through the capacitor.

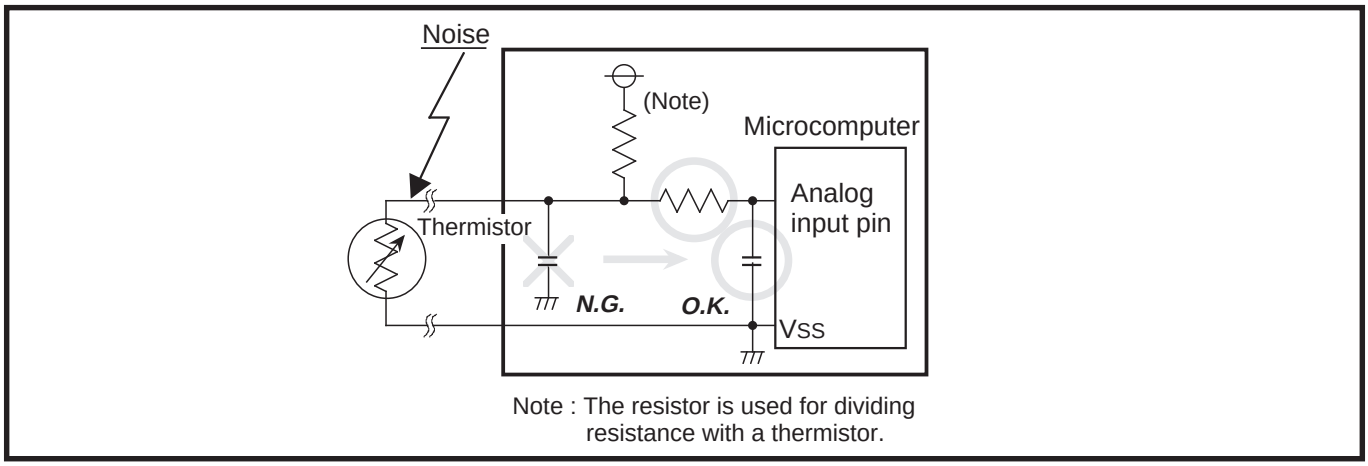


Fig. 3.4.6 Analog signal line and a resistor and a capacitor

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.

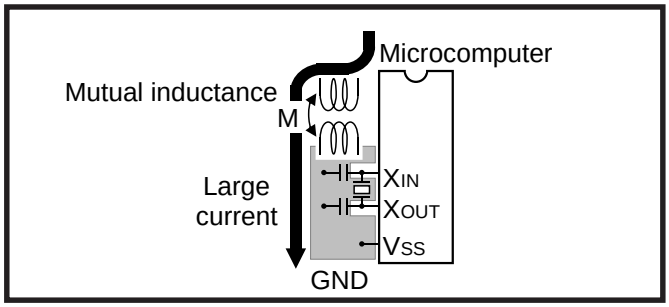


Fig. 3.4.7 Wiring for a large current signal line

(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.

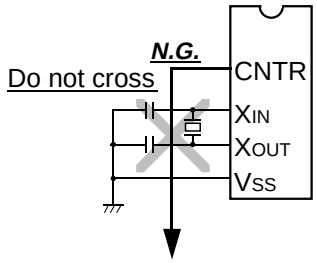


Fig. 3.4.8 Wiring of signal lines where potential levels change frequently

(3) Oscillator protection using Vss pattern

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

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

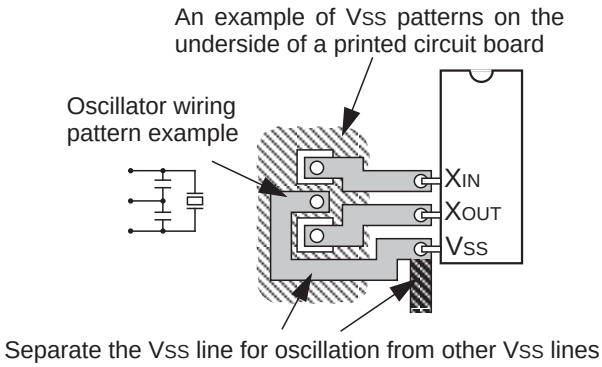


Fig. 3.4.9 Vss pattern on the underside of an oscillator

3.4 Countermeasures against noise

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.

Note: When a direction register is set for input port again at fixed periods, a several-nanosecond short pulse may be output from this port. If this is undesirable, connect a capacitor to this port to remove the noise pulse.

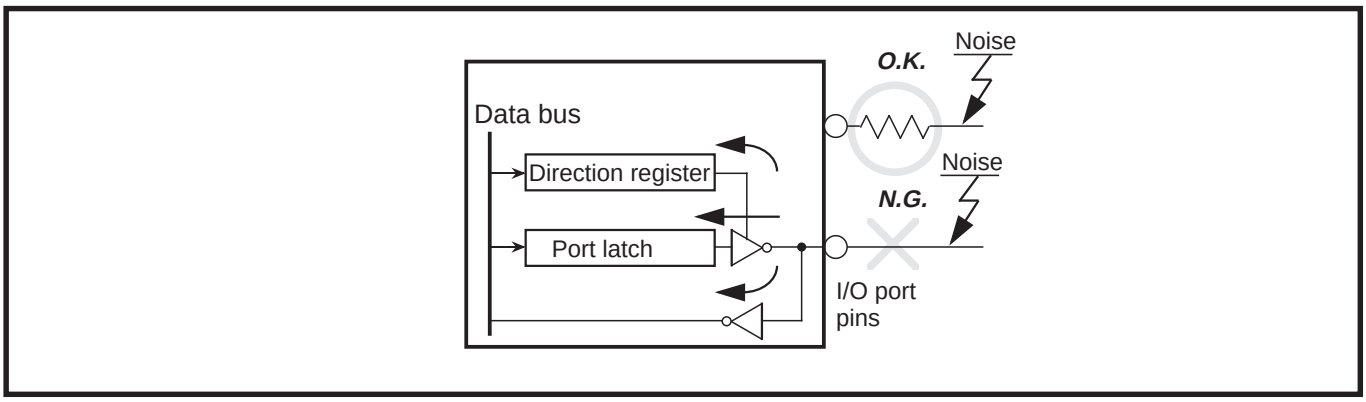


Fig. 3.4.10 Setup for I/O ports

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.

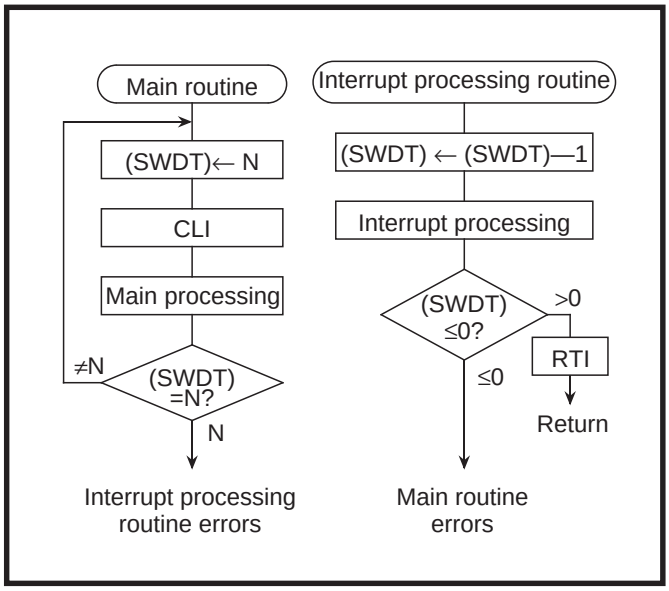


Fig. 3.4.11 Watchdog timer by software

3.5 Control registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (i = 0, 1, 2, 3, 4, 5, 6, 8)
(Pi: addresses 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}, $0A_{16}$, $0C_{16}$, 10_{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

Port P0 direction register, Port P1 direction register

b7 b6 b5 b4 b3 b2 b1 b0

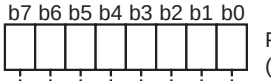
Port P0 direction register (P0D: address 01_{16})
Port P1 direction register (P1D: address 03_{16})

b	Name	Functions	At reset	R	W
0	Ports P0/P1 direction register	0 : All bits of ports P0/P1 input mode 1 : All bits of ports P0/P1 output mode	0	X	○
1	Nothing is arranged for these bits. When these bits are read out, the contents are undefined.		0	X	X
2			0	X	X
3			0	X	X
4			0	X	X
5			0	X	X
6			0	X	X
7			0	X	X

Note: Ports P0 and P1 are switched to input and output by each port. When b0 of corresponding port direction register is set to "0", all 8 bits of port become input port. When b0 of corresponding port direction register is set to "1", all 8 bits of port become output port. Nothing is arranged for b1 to b7 of port P0 and port P1 direction registers. These are write disabled bits.

Fig. 3.5.2 Structure of Port P0 direction register and Port P1 direction register

Port Pi direction register



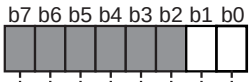
Port Pi direction register (i = 2, 4, 5, 6, 8)
(PiD: addresses 05_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, 11_{16})

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 **(Note)**	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	×	○

Note: Bit 1 of the port P5 direction register (address $0B_{16}$) does not have direction register function, because $P5_1$ is an input port. When writing to bit 1 of the port P5 direction register, write "0" to the bit.

Fig. 3.5.3 Structure of Port Pi direction register

Port P7

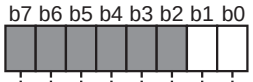


Port P7
(P7: address $0E_{16}$)

b	Name	Functions	At reset	R	W
0	Port $P7_0$	●In output mode Write •••••••• Port latch Read •••••••• Port latch ●In input mode Write •••••••• Port latch Read •••••••• Value of pin	0	○	○
1	Port $P7_1$		0	○	○
2	Nothing is arranged for these bits. When these bits are read out, the contents are undefined.		0	×	×
3			0	×	×
4			0	×	×
5			0	×	×
6			0	×	×
7			0	×	×

Fig. 3.5.4 Structure of Port P7

Port P7 direction register

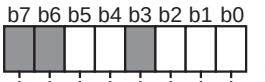


Port P7 direction register
(P7D: address $0F_{16}$)

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	Nothing is arranged for these bits. When these bits are read out, the contents are undefined.		0	×	×
3			0	×	×
4			0	×	×
5			0	×	×
6			0	×	×
7			0	×	×

Fig. 3.5.5 Structure of Port P7 direction register

PULL register A



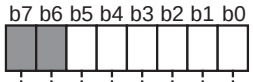
PULL register A
(PULLA: address 16_{16})

b	Name	Functions	At reset	R	W
0	Port $P0_0$–$P0_7$ pull-down control	0: No pull-down control 1: Pull-down control	1	○	○
1	Port $P1_0$–$P1_7$ pull-down control	0: No pull-down control 1: Pull-down control	1	○	○
2	Port $P2_0$–$P2_7$ pull-down control	0: No pull-down control 1: Pull-down control	1	○	○
3	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “1”.		1	○	×
4	Port $P7_0$, $P7_1$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
5	Port $P8_0$–$P8_7$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
6	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Note: The pin which is set to output port is cut off from pull-up control.

Fig. 3.5.6 Structure of PULL register A

PULL register B



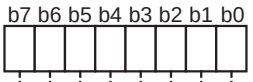
PULL register B
(PULLB: address 17_{16})

b	Name	Functions	At reset	R	W
0	Port $P4_0$–$P4_3$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
1	Port $P4_4$–$P4_7$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
2	Port $P5_0$, $P5_2$, $P5_3$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
3	Port $P5_4$–$P5_7$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
4	Port $P6_0$–$P6_3$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
5	Port $P6_4$–$P6_7$ pull-up control	0: No pull-up control 1: Pull-up control	0	○	○
6	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Note: The pin which is set to output port is cut off from pull-up control.

Fig. 3.5.7 Structure of PULL register B

Port P8 output selection register

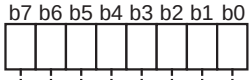


Port P8 output selection register (P8SEL: address 18_{16})

b	Name	Functions	At reset	R	W
0	Port P8 output selection register	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
1		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
2		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
3		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
4		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
6		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
7		0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○

Fig. 3.5.8 Structure of Port P8 output selection register

Serial I/O control register 1



Serial I/O control register 1
(SIOCON1: address 19_{16})

b	Name	Functions	At reset	R	W
0	Internal synchronous clock selection bits	b2b1b0 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 1 0: f(X_{IN})/128 or f(X_{CIN})/128 1 1 1: f(X_{IN})/256 or f(X_{CIN})/256	0	○	○
1			0	○	○
2			0	○	○
3	Serial I/O port selection bit ($P4_0$, $P4_5$, $P4_6$)	0: I/O port 1: S_{OUT}, S_{CLK1}, S_{CLK2} signal pin	0	○	○
4	$\overline{S_{RDY}}$ output selection bit ($P4_7$)	0: I/O port 1: $\overline{S_{RDY}}$ signal pin	0	○	○
5	Transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
6	Synchronous clock selection bit	0: External clock 1: Internal clock	0	○	○
7	P-channel output disable bit ($P4_0$, $P4_5$, $P4_6$)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○

Fig. 3.5.9 Structure of Serial I/O control register 1

3.5 Control registers

Serial I/O control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O control register 2
(SIOCON2: address $1A_{16}$)

b	Name	Functions	At reset	R	W
0	Synchronous clock output pin selection bit	0: S_{CLK1} 1: S_{CLK2}	0	○	○
1	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
2			0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.10 Structure of Serial I/O control register 2

Serial I/O register

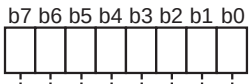
b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O register
(SIO: address $1B_{16}$)

b	Name	Functions	At reset	R	W
0	Serial I/O register	This register becomes shift register. Set transmit data to this register. The serial transfer is started by writing the transmit data.	Undefined	○	○
1			Undefined	○	○
2			Undefined	○	○
3			Undefined	○	○
4			Undefined	○	○
5			Undefined	○	○
6			Undefined	○	○
7			Undefined	○	○

Fig. 3.5.11 Structure of Serial I/O register

Timer i

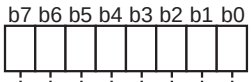


Timer i (i = 1, 3, 4, 5, 6)
(Ti: addresses 20_{16}, 22_{16}, 23_{16}, 24_{16}, 25_{16})

b	Functions	At reset	R	W
0	• Set timer i count value.	1	○	○
1	• The value set in this register is written to both	1	○	○
2	the timer i and the timer i latch at one time.	1	○	○
3	• When the timer i is read out, the count value	1	○	○
4	of the timer i is read out.	1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 3.5.12 Structure of Timer i

Timer 2



Timer 2
(T2: address 21_{16})

b	Functions	At reset	R	W
0	• Set timer 2 count value.	1	○	○
1	• The value set in this register is written to both	0	○	○
2	the timer 2 and the timer 2 latch at one time.	0	○	○
3	• When the timer 2 is read out, the count value	0	○	○
4	of the timer 2 is read out.	0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Fig. 3.5.13 Structure of Timer 2

3.5 Control registers

Timer 6 PWM register

b7 b6 b5 b4 b3 b2 b1 b0

Timer 6 PWM register
(T6PWM: address 27_{16})

b	Functions	At reset	R	W
0	• In timer 6 PWM_1 mode "L" level width of PWM rectangular waveform is set. • Duty of PWM rectangular waveform: n/(n + m) Period: (n + m) × ts n = timer 6 set value m = timer 6 PWM register set value ts = timer 6 count source period At n = 0, all PWM output "L". At m = 0, all PWM output "H". (However, n = 0 has priority.) • Selection of timer 6 PWM_1 mode Set "1" to the timer 6 operation mode selection bit.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 3.5.14 Structure of Timer 6 PWM register

Timer 12 mode register

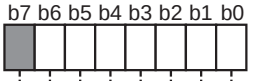
b7 b6 b5 b4 b3 b2 b1 b0

Timer 12 mode register
(T12M: address 28_{16})

b	Name	Functions	At reset	R	W
0	Timer 1 count stop bit	0: Count operation 1: Count stop	0	○	○
1	Timer 2 count stop bit	0: Count operation 1: Count stop	0	○	○
2	Timer 1 count source selection bits	b3 b2 0 0: $f(X_{IN})/16$ or $f(X_{CIN})/16$ 0 1: $f(X_{CIN})$ 1 0: $f(X_{IN})/32$ or $f(X_{CIN})/32$ 1 1: $f(X_{IN})/128$ or $f(X_{CIN})/128$	0	○	○
3			0	○	○
4	Timer 2 count source selection bits	b5 b4 0 0: Timer 1 underflow 0 1: $f(X_{CIN})$ 1 0: External count input $CNTR_0$ 1 1: Not available	0	○	○
5			0	○	○
6	Timer 1 output selection bit ($P4_1$)	0: I/O port 1: Timer 1 output	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×

Fig. 3.5.15 Structure of Timer 12 mode register

Timer 34 mode register

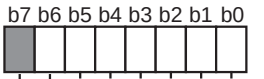


Timer 34 mode register
(T34M: address 29_{16})

b	Name	Functions	At reset	R	W
0	Timer 3 count stop bit	0: Count operation 1: Count stop	0	○	○
1	Timer 4 count stop bit	0: Count operation 1: Count stop	0	○	○
2 3	Timer 3 count source selection bits	b3 b2 0 0: f(XIN)/16 or f(XCIN)/16 0 1: Timer 2 underflow 1 0: f(XIN)/32 or f(XCIN)/32 1 1: f(XIN)/128 or f(XCIN)/128	0	○	○
4 5	Timer 4 count source selection bits	b5 b4 0 0: f(XIN)/16 or f(XCIN)/16 0 1: Timer 3 underflow 1 0: External count input CNTR1 1 1: Not available	0 0	○ ○	○ ○
6	Timer 3 output selection bit (P42)	0: I/O port 1: Timer 3 output	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Fig. 3.5.16 Structure of Timer 34 mode register

Timer 56 mode register



Timer 56 mode register
(T56M: address $2A_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 5 count stop bit	0: Count operation 1: Count stop	0	○	○
1	Timer 6 count stop bit	0: Count operation 1: Count stop	0	○	○
2	Timer 5 count source selection bit	0: f(XIN)/16 or f(XCIN)/16 1: Timer 4 underflow	0	○	○
3	Timer 6 operation mode selection bit	0: Timer mode 1: PWM mode	0	○	○
4 5	Timer 6 count source selection bits	b5 b4 0 0: f(XIN)/16 or f(XCIN)/16 0 1: Timer 5 underflow 1 0: Timer 4 underflow 1 1: Not available	0 0	○ ○	○ ○
6	Timer 6 (PWM) output selection bit (P52)	0: I/O port 1: Timer 6 output	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Fig. 3.5.17 Structure of Timer 56 mode register

3.5 Control registers

φ output control register

b7 b6 b5 b4 b3 b2 b1 b0

φ output control register
(CKOUT: address $2B_{16}$)

b	Name	Functions	At reset	R	W
0	φ output control bit ($P4_3$)	0: Port function 1: φ clock output	0	○	○
1	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
2			0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.18 Structure of φ output control register

Timer A register (low-order, high-order)

b7 b6 b5 b4 b3 b2 b1 b0

Timer A register (low-order, high-order)
(TAL, TAH: addresses $2C_{16}$, $2D_{16}$)

b	Functions	At reset	R	W
0	• Set timer A count value. • When the timer A write control bit of the timer A mode register is "0", the value is written to timer A and the latch at one time. When the timer A write control bit of the timer A mode register is "1", the value is written only to the latch. • The timer A 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 TAH and TAL following.
3: Write both registers in order of TAL and TAH following.
4: Do not read both registers during a write, and do not write to both registers during a read.

Fig. 3.5.19 Structure of Timer A register (low-order, high-order)

Compare register (low-order, high-order)

b7 b6 b5 b4 b3 b2 b1 b0

Compare register (low-order, high-order)
(CONAL, CONAH: addresses $2E_{16}$, $2F_{16}$)

b	Functions	At reset	R	W
0	• Set compare register value.	0	○	○
1		0	○	○
2		0	○	○
3		0	○	○
4		0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Note: Write registers in order of CONAH, CONAL, TAL, and TAH following.

Fig. 3.5.20 Structure of Compare register (low-order, high-order)

Timer A mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer A mode register
(TAM: address 30_{16})

b	Name	Functions	At reset	R	W
0	Timer A operating mode bits	b1b0 0 0: Timer mode 0 1: Pulse output mode 1 0: IGBT output mode 1 1: PWM mode	0	○	○
1			0	○	○
2	Timer A write control bit	0: Write data to both timer latch and timer 1: Write data to timer latch	0	○	○
3	Timer A count source selection bits	b4b3 0 0: f(X_{IN}) 0 1: f(X_{IN})/2 1 0: f(X_{IN})/4 1 1: f(X_{IN})/8	0	○	○
4			0	○	○
5	Timer A output active edge switch bit	0: Output starts with “L” level 1: Output starts with “H” level	0	○	○
6	Timer A count stop bit	0: Count operating 1: Count stop	0	○	○
7	Timer A output selection bit ($P5_0$)	0: I/O port 1: Timer A output	0	○	○

Fig. 3.5.21 Structure of Timer A mode register

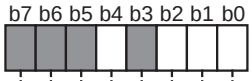
Timer A control register

Timer A control register
(TACON: address 31_{16})

b	Name	Functions	At reset	R	W
0	Noise filter sampling clock selection bit	0: $f(X_{IN})/2$ 1: $f(X_{IN})/4$	0	○	○
1	External trigger delay time selection bits	b2b1 0 0: No delay 0 1: $(4/f(X_{IN}))\mu s$ 1 0: $(8/f(X_{IN}))\mu s$ 1 1: $(16/f(X_{IN}))\mu s$	0	○	○
2			0	○	○
3	Timer A output control bit 1 ($P5_6$)	0: Not used 1: INT1 interrupt used	0	○	○
4	Timer A output control bit 2 ($P5_7$)	0: Not used 1: INT2 interrupt used	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.22 Structure of Timer A control register

A-D control register



A-D control register
(ADCON: address 32_{16})

b	Name	Functions	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0: $P6_0/AN_0$ 0 0 1: $P6_1/AN_1$ 0 1 0: $P6_2/AN_2$ 0 1 1: $P6_3/AN_3$ 1 0 0: $P6_4/AN_4$ 1 0 1: $P6_5/AN_5$ 1 1 0: $P6_6/AN_6$ 1 1 1: $P6_7/AN_7$	0	○	○
1			0	○	○
2			0	○	○
3	Nothing is arranged for this bit. This is write disabled bit. When this bit is read out, the contents are "0".		0	○	×
4	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.23 Structure of A-D control register

A-D conversion register (low-order)

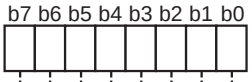
A-D conversion register (low-order)
(ADL: address 33_{16})

b	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".	Undefined	○	×
1		Undefined	○	×
2		Undefined	○	×
3		Undefined	○	×
4		Undefined	○	×
5		Undefined	○	×
6	These are A-D conversion result (low-order 2 bits) stored bits. This is read exclusive register.	Undefined	○	×
7		Undefined	○	×

Note: Do not read this register during A-D conversion.

Fig. 3.5.24 Structure of A-D conversion register (low-order)

A-D conversion register (high-order)



A-D conversion register (high-order)
(ADH: address 34_{16})

b	Functions	At reset	R	W
0	This is A-D conversion result (high-order 8 bits) stored bits. This is read exclusive register.	Undefined	○	×
1		Undefined	○	×
2		Undefined	○	×
3		Undefined	○	×
4		Undefined	○	×
5		Undefined	○	×
6		Undefined	○	×
7		Undefined	○	×

Note: Do not read this register during A-D conversion.

Fig. 3.5.25 Structure of A-D conversion register (high-order)

3.5 Control registers

Segment output enable register

b7 b6 b5 b4 b3 b2 b1 b0

Segment output enable register
(SEG: address 38_{16})

b	Name	Functions	At reset	R	W
0	Segment output enable bit 0	0: I/O ports $P2_0$–$P2_3$ 1: Segment output SEG_0–SEG_3	0	○	○
1	Segment output enable bit 1	0: I/O ports $P2_4$–$P2_7$ 1: Segment output SEG_4–SEG_7	0	○	○
2	Segment output enable bit 2	0: I/O ports $P0_0$–$P0_3$ 1: Segment output SEG_8–SEG_{11}	0	○	○
3	Segment output enable bit 3	0: I/O ports $P0_4$–$P0_7$ 1: Segment output SEG_{12}–SEG_{15}	0	○	○
4	Segment output enable bit 4	0: I/O ports $P1_0$–$P1_3$ 1: Segment output SEG_{16}–SEG_{19}	0	○	○
5	Segment output enable bit 5	0: I/O ports $P1_4$–$P1_7$ 1: Segment output SEG_{20}–SEG_{23}	0	○	○
6	Segment output enable bit 6	0: Output ports $P3_0$–$P3_3$ 1: Segment output SEG_{24}–SEG_{27}	0	○	○
7	Segment output enable bit 7	0: Output ports $P3_4$–$P3_7$ 1: Segment output SEG_{28}–SEG_{31}	0	○	○

Fig. 3.5.26 Structure of Segment output enable register

LCD mode register

b7 b6 b5 b4 b3 b2 b1 b0

LCD mode register
(LM: address 39_{16})

b	Name	Functions	At reset	R	W
0	Duty ratio selection bits	b1b0 0 0: 1 (use COM_0) 0 1: 2 (use COM_0, COM_1) 1 0: 3 (use COM_0–COM_2) 1 1: 4 (use COM_0–COM_3)	0	○	○
1			0	○	○
2	Bias control bit	0: 1/3 bias 1: 1/2 bias	0	○	○
3	LCD enable bit	0: LCD OFF 1: LCD ON	0	○	○
4	Fix “0” to this bit.		0	○	○
5	LCD circuit divider division ratio selection bits	b6b5 0 0: Clock input 0 1: 2 division of clock input 1 0: 4 division of clock input 1 1: 8 division of clock input	0	○	○
6			0	○	○
7	LCDCK count source selection bit **(Note)**	0: f(X_{CIN})/32 1: f(X_{IN})/8192 (f(X_{CIN})/8192 in low-speed mode)	0	○	○

Note: LCDCK is a clock for a LCD timing controller.

Fig. 3.5.27 Structure of LCD mode register

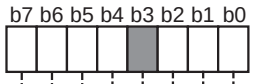
Interrupt edge selection register

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: I/O ports $P2_4$–$P2_7$ 1: Segment output SEG_4–SEG_7	0	○	○
2	INT_2 interrupt edge selection bit	0: I/O ports $P0_0$–$P0_3$ 1: Segment output SEG_8–SEG_{11}	0	○	○
3	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
4			0	○	×
5			0	○	×
6	$CNTR_0$ active edge switch bit	0: Falling edge active, rising edge count 1: Rising edge active, falling edge count	0	○	○
7	$CNTR_1$ active edge switch bit	0: Falling edge active, rising edge count 1: Rising edge active, falling edge count	0	○	○

Fig. 3.5.28 Structure of Interrupt edge selection register

CPU mode register

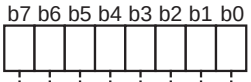


CPU mode register
(CPUM, CM: address $3B_{16}$)

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : 10 : 11 : } Not available	0	○	○
1			0	○	○
2	Stack page selection bit	0 : Page 0 1 : Page 1	0	○	○
3	Nothing is arranged for this bit. When this bit is read out, the contents is “1”. Do not write “0” to this bit.		1	○	○
4	Port Xc switch bit	0: I/O port function 1: X_{CIN}-X_{COUT} oscillation function	0	○	○
5	Main clock (X_{IN}-X_{OUT}) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bit	0: f(X_{IN})/2 (high-speed mode) 1: f(X_{IN})/8 (middle-speed mode)	1	○	○
7	Internal system clock selection bit	0: X_{IN}–X_{OUT} selection (middle-/high-speed mode) 1: X_{CIN}–X_{COUT} selection (low-speed mode)	0	○	○

Fig. 3.5.29 Structure of CPU mode register

Interrupt request register 1



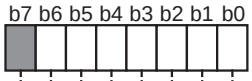
Interrupt request register 1
(IREQ1 : address $3C_{16}$)

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	INT_2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer A interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer 1 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. 3.5.30 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	Timer 4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Timer 5 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Timer 6 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	$CNTR_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	$CNTR_1$ 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	AD conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

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

Fig. 3.5.31 Structure of Interrupt request register 2

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

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	INT_2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer A interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 1 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. 3.5.32 Structure of Interrupt control register 1

Interrupt control register 2

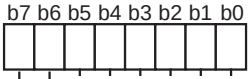
b7 b6 b5 b4 b3 b2 b1 b0

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

b	Name	Functions	At reset	R	W
0	Timer 4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Timer 5 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
2	Timer 6 interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
3	$CNTR_0$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
4	$CNTR_1$ interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key input interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD conversion interrupt enable bit	0 : interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

Fig. 3.5.33 Structure of Interrupt control register 2

ROM correct enable register 1



ROM correct enable register 1
(RC1: address 0F01$_{16}$)

b	Name	Functions	At reset	R	W
0	ROM correct address 1 enable bit	0: Disabled 1: Enabled	0	○	○
1	ROM correct address 2 enable bit	0: Disabled 1: Enabled	0	○	○
2	ROM correct address 3 enable bit	0: Disabled 1: Enabled	0	○	○
3	ROM correct address 4 enable bit	0: Disabled 1: Enabled	0	○	○
4	ROM correct address 5 enable bit	0: Disabled 1: Enabled	0	○	○
5	ROM correct address 6 enable bit	0: Disabled 1: Enabled	0	○	○
6	ROM correct address 7 enable bit	0: Disabled 1: Enabled	0	○	○
7	ROM correct address 8 enable bit	0: Disabled 1: Enabled	0	○	○

Fig. 3.5.34 Structure of ROM correct enable register 1

3.6 Mask ROM confirmation form

GZZ-SH56-24B<91A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38C34M6AXXXFP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

					Issuance signature	Submitted by	Supervisor
✵	Customer	Company name		TEL ( )			
		Date issued	Date:				

✵ 1. Confirmation
Three EPROMs are required for each pattern if this order is performed by EPROMs.
One floppy disk is required for each pattern if this order is performed by a floppy disk.

Microcomputer name: ☐ M38C34M6AXXXFP

☐ Ordering by EPROMs
Specify the type of EPROMs submitted.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Product name ASCII code : 'M38C34M6A'	0000_{16} Product name ASCII code : 'M38C34M6A'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$207F_{16}$	$A07F_{16}$
2080_{16} data	$A080_{16}$ data
$7FFD_{16}$ ROM (24K-130) bytes	$FFFD_{16}$ ROM (24K-130) bytes
$7FFE_{16}$	$FFFE_{16}$
$7FFF_{16}$	$FFFF_{16}$

In the address space of the microcomputer, the internal ROM area is from address $A080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38C34M6A" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$.
The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address	
0000_{16}	'M' = $4D_{16}$
0001_{16}	'3' = 33_{16}
0002_{16}	'8' = 38_{16}
0003_{16}	'C' = 43_{16}
0004_{16}	'3' = 33_{16}
0005_{16}	'4' = 34_{16}
0006_{16}	'M' = $4D_{16}$
0007_{16}	'6' = 36_{16}

Address	
0008_{16}	'A' = 41_{16}
0009_{16}	FF_{16}
$000A_{16}$	FF_{16}
$000B_{16}$	FF_{16}
$000C_{16}$	FF_{16}
$000D_{16}$	FF_{16}
$000E_{16}$	FF_{16}
$000F_{16}$	FF_{16}

GZZ-SH56-24B<91A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38C34M6AXXXFP
MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000_{16} to 0008_{16} of EPROM.

EPROM type	27256	27512
The pseudo-command	*= △$8000 .BYTE △'M38C34M6A'	*= △$0000 .BYTE △'M38C34M6A'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

☐ Ordering by floppy disk

We will produce masks based on the mask files generated by the mask file generating utility. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this mask file. Thus, extreme care must be taken to verify the mask file in the submitted floppy disk.

The submitted floppy disk must be 3.5-inch 2HD type and DOS/V format. And the number of the mask files must be 1 in one floppy disk.

File code ☐☐☐☐☐☐☐☐ (hexadecimal notation)

Mask file name ☐☐☐☐☐☐☐☐ .MSK (equal or less than eight characters)

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (80P6N for M38C34M6AXXXFP) and attach it to the mask ROM confirmation form.

✵ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the XIN-XOUT oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? f(XIN) = ☐ MHz

(2) Which function will you use the pins P70/XCIN and P71/XCOUT as P70 and P71, or XCIN and XCOUT ?

☐ Ports P70 and P71 function ☐ XCIN and XCOUT function (external resonator)

✵ 4. Comments

APPENDIX

3.6 Mask ROM confirmation form

GZZ-SH56-25B<91A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38C34M6MXXXFP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

✵	Customer	Company name	TEL ( )	Issuance signature	Submitted by	Supervisor
		Date issued	Date:			

✵ 1. Confirmation
Three EPROMs are required for each pattern if this order is performed by EPROMs.
One floppy disk is required for each pattern if this order is performed by a floppy disk.

Microcomputer name: ☐ M38C34M6MXXXFP

☐ Ordering by EPROMs
Specify the type of EPROMs submitted.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Product name ASCII code : 'M38C34M6M'	0000_{16} Product name ASCII code : 'M38C34M6M'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$207F_{16}$	$A07F_{16}$
2080_{16} data ROM (24K-130) bytes	$A080_{16}$ data ROM (24K-130) bytes
$7FFD_{16}$	$FFFD_{16}$
$7FFE_{16}$	$FFFE_{16}$
$7FFF_{16}$	$FFFF_{16}$

In the address space of the microcomputer, the internal ROM area is from address $A080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38C34M6M" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$.
The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address	
0000_{16}	'M' = $4D_{16}$
0001_{16}	'3' = 33_{16}
0002_{16}	'8' = 38_{16}
0003_{16}	'C' = 43_{16}
0004_{16}	'3' = 33_{16}
0005_{16}	'4' = 34_{16}
0006_{16}	'M' = $4D_{16}$
0007_{16}	'6' = 36_{16}

Address	
0008_{16}	'M' = $4D_{16}$
0009_{16}	FF_{16}
$000A_{16}$	FF_{16}
$000B_{16}$	FF_{16}
$000C_{16}$	FF_{16}
$000D_{16}$	FF_{16}
$000E_{16}$	FF_{16}
$000F_{16}$	FF_{16}

(1/2)

GZZ-SH56-25B<91A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38C34M6MXXXFP
MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000_{16} to 0008_{16} of EPROM.

EPROM type	27256	27512
The pseudo-command	*= △$8000 .BYTE △'M38C34M6M'	*= △$0000 .BYTE △'M38C34M6M'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

☐ Ordering by floppy disk

We will produce masks based on the mask files generated by the mask file generating utility. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this mask file. Thus, extreme care must be taken to verify the mask file in the submitted floppy disk.

The submitted floppy disk must be 3.5-inch 2HD type and DOS/V format. And the number of the mask files must be 1 in one floppy disk.

File code ☐☐☐☐☐☐☐☐ (hexadecimal notation)

Mask file name ☐☐☐☐☐☐☐☐ .MSK (equal or less than eight characters)

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (80P6N for M38C34M6MXXXFP) and attach it to the mask ROM confirmation form.

✵ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the XIN-XOUT oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? f(XIN) = ☐ MHz

(2) Which function will you use the pins P70/XCIN and P71/XCOUT as P70 and P71, or XCIN and XCOUT ?

☐ Ports P70 and P71 function ☐ XCIN and XCOUT function (external resonator)

✵ 4. Comments

3.7 ROM programming confirmation form

GZZ-SH56-29B<91A0>

ROM number	

740 FAMILY WRITING TO PROM CONFIRMATION FORM
SINGLE-CHIP 8-BIT MICROCOMPUTER M38C37ECAXXXFP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

✵	Customer	Company name	TEL ( )	Issuance signature	Submitted by	Supervisor
		Date issued	Date:			

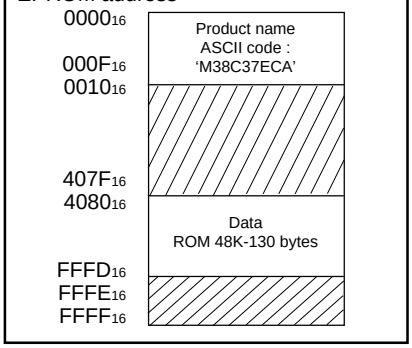
✵ 1. Confirmation
Three EPROMs are required for each pattern if this order is performed by EPROMs.
One floppy disk is required for each pattern if this order is performed by a floppy disk.

☐ Ordering by EPROMs
If at least two of the three sets of EPROMs submitted contain identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

EPROM address Checksum code for entire EPROM [ ][ ][ ][ ] (hexadecimal notation)

EPROM type (indicate the type used)

☐ **27512**



In the address space of the microcomputer, the internal ROM area is from address 4080_{16} to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38C37ECA" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' A ' =41_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'C' = 43_{16}	$000B_{16}$	FF_{16}
0004_{16}	'3' = 33_{16}	$000C_{16}$	FF_{16}
0005_{16}	'7' = 37_{16}	$000D_{16}$	FF_{16}
0006_{16}	'E' = 45_{16}	$000E_{16}$	FF_{16}
0007_{16}	'C' = 43_{16}	$000F_{16}$	FF_{16}

(1/2)

GZZ-SH56-29B<91A0>

ROM number	

740 FAMILY WRITING TO PROM CONFIRMATION FORM
SINGLE-CHIP 8-BIT MICROCOMPUTER M38C37ECAXXXFP
MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000_{16} to 0008_{16} of EPROM.

EPROM type	27512
The pseudo-command	*= △$0000 .BYTE △'M38C37ECA'

Note : If the name of the product written to the EPROMs does not match the name of the writing to PROM confirmation form, the ROM will not be processed.

☐ Ordering by floppy disk

We will produce writing to PROM based on the mask files generated by the mask file generating utility. We shall assume the responsibility for errors only if the written PROM data on the products we produce differs from this mask file. Thus, extreme care must be taken to verify the mask file in the submitted floppy disk.

The submitted floppy disk must be 3.5-inch 2HD type and DOS/V format. And the number of the mask files must be 1 in one floppy disk.

File code ☐☐☐☐☐☐☐☐ (hexadecimal notation)

Mask file name ☐☐☐☐☐☐☐☐ .MSK (equal or less than eight characters)

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate 80P6N mark specification form and attach it to the writing to PROM confirmation form.

✵ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the XIN-XOUT oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? f(XIN) = ☐ MHz

(2) Which function will you use the pins P70/XCIN and P71/XCOUT as P70 and P71, or XCIN and XCOUT ?

☐ Ports P70 and P71 function ☐ XCIN and XCOUT function (external resonator)

✵ 4. Comments

APPENDIX

3.7 ROM programming confirmation form

GZZ-SH56-30B<91A0>

ROM number	

740 FAMILY WRITING TO PROM CONFIRMATION FORM
SINGLE-CHIP 8-BIT MICROCOMPUTER M38C37ECMXXXFP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

✵	Customer	Company name	TEL ( )	Issuance signature	Submitted by	Supervisor
		Date issued	Date:			

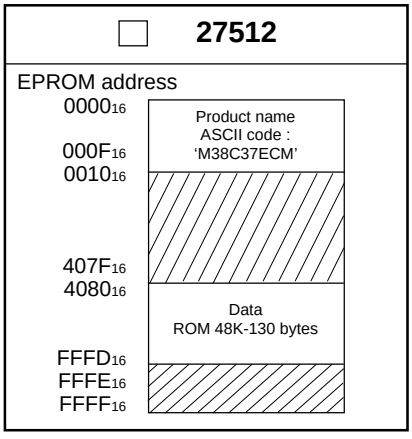
✵ 1. Confirmation
Three EPROMs are required for each pattern if this order is performed by EPROMs.
One floppy disk is required for each pattern if this order is performed by a floppy disk.

☐ Ordering by EPROMs
If at least two of the three sets of EPROMs submitted contain identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

EPROM address Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ **27512**



In the address space of the microcomputer, the internal ROM area is from address 4080_{16} to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38C37ECM" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' M ' =$4D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'C' = 43_{16}	$000B_{16}$	FF_{16}
0004_{16}	'3' = 33_{16}	$000C_{16}$	FF_{16}
0005_{16}	'7' = 37_{16}	$000D_{16}$	FF_{16}
0006_{16}	'E' = 45_{16}	$000E_{16}$	FF_{16}
0007_{16}	'C' = 43_{16}	$000F_{16}$	FF_{16}

(1/2)

GZZ-SH56-30B<91A0>

ROM number	

740 FAMILY WRITING TO PROM CONFIRMATION FORM
SINGLE-CHIP 8-BIT MICROCOMPUTER M38C37ECMXXXFP
MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000_{16} to 0008_{16} of EPROM.

EPROM type	27512
The pseudo-command	*= △$0000 .BYTE △'M38C37ECM'

Note : If the name of the product written to the EPROMs does not match the name of the writing to PROM confirmation form, the ROM will not be processed.

☐ Ordering by floppy disk

We will produce writing to PROM based on the mask files generated by the mask file generating utility. We shall assume the responsibility for errors only if the written PROM data on the products we produce differs from this mask file. Thus, extreme care must be taken to verify the mask file in the submitted floppy disk.

The submitted floppy disk must be 3.5-inch 2HD type and DOS/V format. And the number of the mask files must be 1 in one floppy disk.

File code ☐☐☐☐☐☐☐☐ (hexadecimal notation)

Mask file name ☐☐☐☐☐☐☐☐ .MSK (equal or less than eight characters)

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate 80P6N mark specification form and attach it to the writing to PROM confirmation form.

✵ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the XIN-XOUT oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? f(XIN) = ☐ MHz

(2) Which function will you use the pins P70/XCIN and P71/XCOUT as P70 and P71, or XCIN and XCOUT ?

☐ Ports P70 and P71 function ☐ XCIN and XCOUT function (external resonator)

✵ 4. Comments

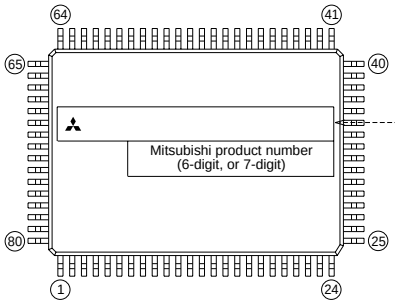
3.8 Mark specification form

80P6N (80-PIN QFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

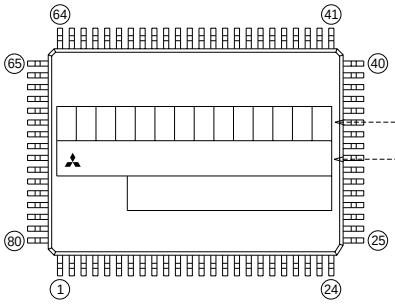
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

B. Customer's Parts Number + Mitsubishi IC Catalog Name



Customer's Parts Number

Note : The fonts and size of characters are standard Mitsubishi type.

Mitsubishi IC catalog name

Notes 1 : The mark field should be written right aligned.

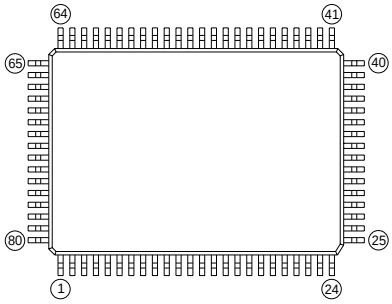
2 : The fonts and size of characters are standard Mitsubishi type.

3 : Customer's parts number can be up to 14 alphanumeric characters for capital letters, hyphens, commas, periods and so on.

4 : If the Mitsubishi logo is not required, check the box below.

Mitsubishi logo is not required ☐

C. Special Mark Required



Notes1 : If special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated technically as close as possible.
Mitsubishi product number (6-digit, or 7-digit) and Mask ROM number (3-digit) are always marked for sorting the products.

2 : If special character fonts (e,g., customer's trade mark logo) must be used in Special Mark, check the box below.
For the new special character fonts, a clean font original (ideally logo drawing) must be submitted.

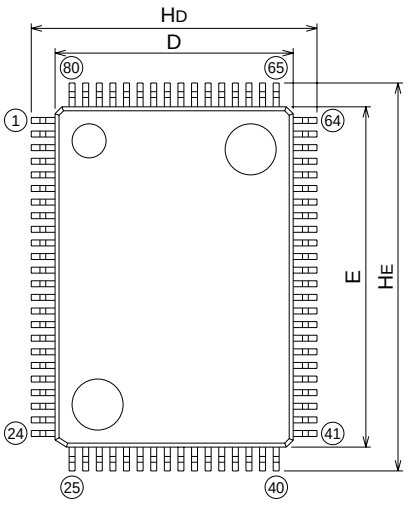
Special character fonts required ☐

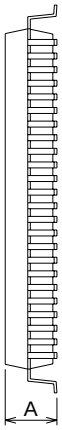
3.9 Package outline

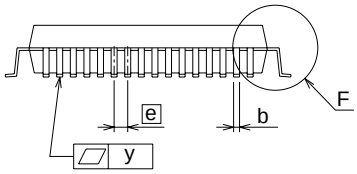
80P6N-A

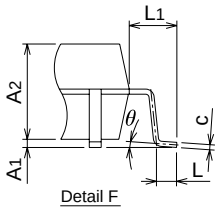
Plastic 80pin 14×20mm body QFP

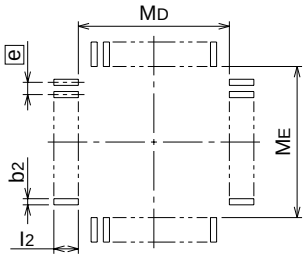
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP80-P-1420-0.80	–	1.58	Alloy 42











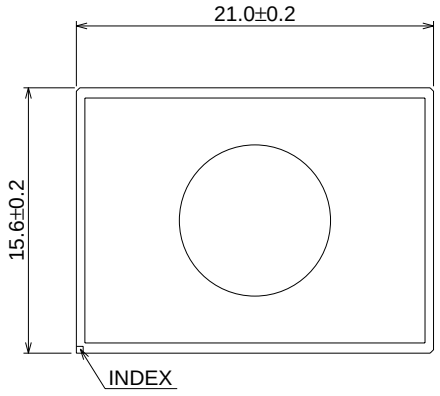
Recommended Mount Pad

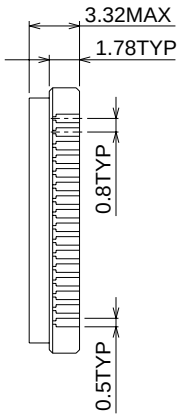
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A1	0	0.1	0.2
A2	–	2.8	–
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	–	0.8	–
HD	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L1	–	1.4	–
y	–	–	0.1
θ	0°	–	10°
b2	–	0.5	–
l2	1.3	–	–
MD	–	14.6	–
ME	–	20.6	–

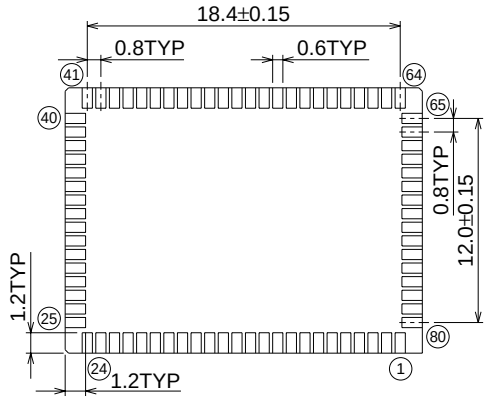
80D0

Glass seal 80pin QFN

EIAJ Package Code	JEDEC Code	Weight(g)
–	–	







3.10 Machine instructions

Symbol	Function	Details	Addressing mode																						Processor status register							
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R			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	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
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				75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2							N	V	•	•	•	•	Z	C
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				35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2							N	•	•	•	•	•	Z	•
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				16	6	2				0E	6	3	1E	7	3																			N	•	•	•	•	•	Z	C
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.																																								90	2	2							•	•	•	•	•	•	•	•
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.																																								B0	2	2							•	•	•	•	•	•	•	•
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.																																								F0	2	2							•	•	•	•	•	•	•	•
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										2C	4	3																							M_7	M_6	•	•	•	•	Z	•
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.																																								30	2	2							•	•	•	•	•	•	•	•
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.																																								D0	2	2							•	•	•	•	•	•	•	•

3.10 Machine instructions

Symbol	Function	Details	Addressing mode																													Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			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	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
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.																																													10	2	2				•	•	•	•	•	•	•	•
BRA	PC ← PC ± offset	This instruction branches to the appointed address. The branch address is specified by a relative address.																																												80	4	2				•	•	•	•	•	•	•	•
BRK	B ← 1 (PC) ← (PC) + 2 M(S) ← PCH S ← S – 1 M(S) ← PCL S ← S – 1 M(S) ← PS S ← S – 1 I← 1 PCL ← ADL PCH ← ADH	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																																													•	•	•	1	•	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.																																												50	2	2				•	•	•	•	•	•	•	•
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.																																												70	2	2				•	•	•	•	•	•	•	•
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																																																•	•	•	•	•	•	•	0
CLD	D ← 0	This instruction clears D.	D8	2	1																																																	•	•	•	•	0	•	•	•
CLI	I ← 0	This instruction clears I.	58	2	1																																																	•	•	•	•	•	0	•	•
CLT	T ← 0	This instruction clears T.	12	2	1																																																	•	•	0	•	•	•	•	•
CLV	V ← 0	This instruction clears V.	B8	2	1																																																	•	0	•	•	•	•	•	•
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				D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
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																																				N	•	•	•	•	•	Z	•
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										EC	4	3																						N	•	•	•	•	•	Z	C
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										CC	4	3																						N	•	•	•	•	•	Z	C
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				D6	6	2				CE	6	3	DE	7	3																					N	•	•	•	•	•	Z	•

Symbol	Function	Details
DEX	X ← X – 1	This instruction subtracts one from the current contents of X.
DEY	Y ← Y – 1	This instruction subtracts one from the current contents of Y.
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.
INC	A ← A + 1 or M ← M + 1	This instruction adds one to the contents of A or M.
INX	X ← X + 1	This instruction adds one to the contents of X.
INY	Y ← Y + 1	This instruction adds one to the contents of Y.
JMP	If addressing mode is ABS PC_L ← AD_L PC_H ← AD_H If addressing mode is IND PC_L ← M (AD_H, AD_L) PC_H ← M (AD_H, AD_L + 1) If addressing mode is ZP, IND PC_L ← M(00, AD_L) PC_H ← 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) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 After executing the above, if addressing mode is ABS, PC_L ← AD_L PC_H ← AD_H if addressing mode is SP, PC_L ← AD_L PC_H ← FF If addressing mode is ZP, IND, PC_L ← M(00, AD_L) PC_H ← 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.
LDM	M ← nn	This instruction loads the immediate value in M.
LDX	X ← M	This instruction loads the contents of M in X.
LDY	Y ← M	This instruction loads the contents of M in Y.

Addressing mode

Symbol	IMP			IMM			A			BIT, A			ZP			BIT, ZP		
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
DEX	CA	2	1															
DEY	88	2	1															
DIV																		
EOR				49	2	2							45	3	2			
INC							3A	2	1				E6	5	2			
INX	E8	2	1															
INY	C8	2	1															
JMP																		
JSR																		
LDA				A9	2	2							A5	3	2			
LDM													3C	4	3			
LDX				A2	2	2							A6	3	2			
LDY				A0	2	2							A4	3	2			

APPENDIX
3.10 Machine instructions

Addressing mode

Symbol	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y		
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
DEX															
DEY															
DIV	E2	16	2												
EOR	55	4	2				4D	4	3	5D	5	3	59	5	3
INC	F6	6	2				EE	6	3	FE	7	3			
INX															
INY															
JMP							4C	3	3						
JSR							20	6	3						
LDA	B5	4	2				AD	4	3	BD	5	3	B9	5	3
LDM															
LDX				B6	4	2	AE	4	3				BE	5	3
LDY	B4	4	2				AC	4	3	BC	5	3			

Symbol	IND			ZP, IND			IND, X			IND, Y			REL			SP		
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
DEX																		
DEY																		
DIV																		
EOR							41	6	2	51	6	2						
INC																		
INX																		
INY																		
JMP	6C	5	3	B2	4	2												
JSR				02	7	2										22	5	2
LDA							A1	6	2	B1	6	2						
LDM																		
LDX																		
LDY																		

Processor status register

Symbol	7	6	5	4	3	2	1	0
	N	V	T	B	D	I	Z	C
DEX	N	•	•	•	•	•	Z	•
DEY	N	•	•	•	•	•	Z	•
DIV	•	•	•	•	•	•	•	•
EOR	N	•	•	•	•	•	Z	•
INC	N	•	•	•	•	•	Z	•
INX	N	•	•	•	•	•	Z	•
INY	N	•	•	•	•	•	Z	•
JMP	•	•	•	•	•	•	•	•
JSR	•	•	•	•	•	•	•	•
LDA	N	•	•	•	•	•	Z	•
LDM	•	•	•	•	•	•	•	•
LDX	N	•	•	•	•	•	Z	•
LDY	N	•	•	•	•	•	Z	•

APPENDIX
3.10 Machine instructions

Symbol	Function	Details	Addressing mode																						Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			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	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
LSR	7 0 0→□→C	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				56	6	2				4E	6	3	5E	7	3																				0	•	•	•	•	•	Z	C
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.																			62	15	2																											•	•	•	•	•	•	•	•
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				15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•
PHA	M(S) ← A 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																																												N	•	•	•	•	•	Z	•
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																																												(Value saved in stack)							
ROL	7 0 ←□←C←	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				36	6	2				2E	6	3	3E	7	3																				N	•	•	•	•	•	Z	C
ROR	7 0 C→□→	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				76	6	2				6E	6	3	7E	7	3																				N	•	•	•	•	•	Z	C
RRF	7 0 →□→	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 PCL ← M(S) S ← S + 1 PCH ← 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 PCL. S is again incremented by one and stores the contents of memory location designated by S in PCH.	40	6	1																																												(Value saved in stack)							
RTS	S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S) (PC) ← (PC) + 1	This instruction increments S by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and the contents of the memory location is stored in PCH. PC is incremented by 1.	60	6	1																																												•	•	•	•	•	•	•	•

3.10 Machine instructions

Symbol	Function	Details	Addressing mode																						Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			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	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
SBC (Note 1) (Note 5)	When T = 0 A ← A – M – $\overline{C}$ When T = 1 M(X) ← 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				F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
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																																													•	•	•	•	•	•	•	1
SED	D ← 1	This instruction set D.	F8	2	1																																													•	•	•	•	1	•	•	•
SEI	I ← 1	This instruction set I.	78	2	1																																													•	•	•	•	•	1	•	•
SET	T ← 1	This instruction set T.	32	2	1																																													•	•	1	•	•	•	•	•
STA	M ← A	This instruction stores the contents of A in M. The contents of A does not change.													85	4	2				95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	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							96	5	2	8E	5	3																						•	•	•	•	•	•	•	•
STY	M ← Y	This instruction stores the contents of Y in M. The contents of Y does not change.													84	4	2				94	5	2				8C	5	3																						•	•	•	•	•	•	•	•
TAX	X ← A	This instruction stores the contents of A in X. The contents of A does not change.	AA	2	1																																												N	•	•	•	•	•	Z	•
TAY	Y ← A	This instruction stores the contents of A in Y. The contents of A does not change.	A8	2	1																																												N	•	•	•	•	•	Z	•
TST	M = 0?	This instruction tests whether the contents of M are “0” or not and modifies the N and Z.													64	3	2																																	N	•	•	•	•	•	Z	•
TSX	X ← S	This instruction transfers the contents of S in X.	BA	2	1																																												N	•	•	•	•	•	Z	•
TXA	A ← X	This instruction stores the contents of X in A.	8A	2	1																																												N	•	•	•	•	•	Z	•
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																																												N	•	•	•	•	•	Z	•
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.

APPENDIX

3.10 Machine instructions

APPENDIX

3.10 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

3.11 List of instruction code

D3 – D0 / Hexadecimal notation / D7 – D4		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		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

3.12 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}	
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}	
0013_{16}	
0014_{16}	
0015_{16}	
0016_{16}	PULL register A (PULLA)
0017_{16}	PULL register B (PULLB)
0018_{16}	Port P8 output selection register (P8SEL)
0019_{16}	Serial I/O control register 1 (SIOCON1)
$001A_{16}$	Serial I/O control register 2 (SIOCON2)
$001B_{16}$	Serial I/O register (SIO)
$001C_{16}$	
$001D_{16}$	
$001E_{16}$	
$001F_{16}$	

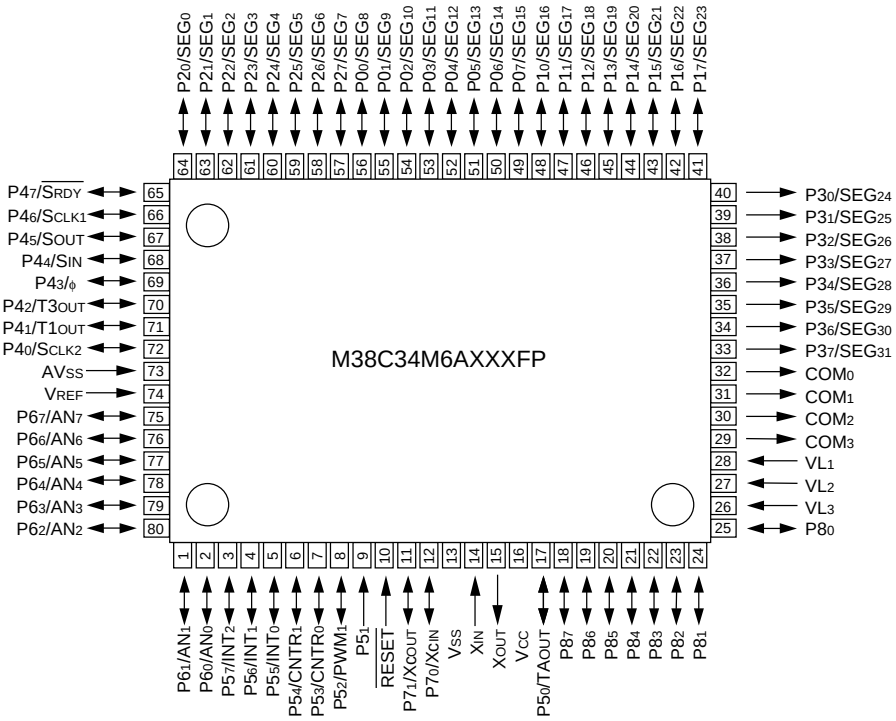
0020_{16}	Timer 1 (T1)
0021_{16}	Timer 2 (T2)
0022_{16}	Timer 3 (T3)
0023_{16}	Timer 4 (T4)
0024_{16}	Timer 5 (T5)
0025_{16}	Timer 6 (T6)
0026_{16}	
0027_{16}	Timer 6 PWM register (T6PWM)
0028_{16}	Timer 12 mode register (T12M)
0029_{16}	Timer 34 mode register (T34M)
$002A_{16}$	Timer 56 mode register (T56M)
$002B_{16}$	ϕ output control register (CKOUT)
$002C_{16}$	Timer A register (low) (TAL)
$002D_{16}$	Timer A register (high) (TAH)
$002E_{16}$	Compare register (low) (CONAL)
$002F_{16}$	Compare register (high) (CONAH)
0030_{16}	Timer A mode register (TAM)
0031_{16}	Timer A control register (TACON)
0032_{16}	A-D control register (ADCON)
0033_{16}	A-D conversion register (low) (ADL)
0034_{16}	A-D conversion register (high) (ADH)
0035_{16}	
0036_{16}	
0037_{16}	
0038_{16}	Segment output enable register (SEG)
0039_{16}	LCD mode register (LM)
$003A_{16}$	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)

$0F01_{16}$	ROM correct enable register 1 **(Note)**
$0F02_{16}$	ROM correct high-order address register 1 **(Note)**
$0F03_{16}$	ROM correct low-order address register 1 **(Note)**
$0F04_{16}$	ROM correct high-order address register 2 **(Note)**
$0F05_{16}$	ROM correct low-order address register 2 **(Note)**
$0F06_{16}$	ROM correct high-order address register 3 **(Note)**
$0F07_{16}$	ROM correct low-order address register 3 **(Note)**
$0F08_{16}$	ROM correct high-order address register 4 **(Note)**
$0F09_{16}$	ROM correct low-order address register 4 **(Note)**

$0F0A_{16}$	ROM correct high-order address register 5 **(Note)**
$0F0B_{16}$	ROM correct low-order address register 5 **(Note)**
$0F0C_{16}$	ROM correct high-order address register 6 **(Note)**
$0F0D_{16}$	ROM correct low-order address register 6 **(Note)**
$0F0E_{16}$	ROM correct high-order address register 7 **(Note)**
$0F0F_{16}$	ROM correct low-order address register 7 **(Note)**
$0F10_{16}$	ROM correct high-order address register 8 **(Note)**
$0F11_{16}$	ROM correct low-order address register 8 **(Note)**

Note: This register is valid only in mask ROM version.

3.13 Pin configuration



(Top view)

Package type: 80P6N-A
80-pin plastic molded QFP

MEMORANDUM

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
38C3 Group

Apr. First Edition 1999

Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

User's Manual
38C3 Group

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: MITSUBISHI DENKI BLDG., MARUNOUCHI, TOKYO 100. TELEX: J24532 CABLE: MELCO TOKYO

New publication, effective Apr. 1999.
Specifications subject to change without notice.