To all our customers

Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

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

3886 Group

User's Manual

RENESAS
Renesas Technology Corp.

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, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents 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.
  The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
  Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (http://www.mitsubishichips.com).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained 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	3886 GROUP USER'S MANUAL

Rev. No.	Revision Description	Rev. date
1.0	First Edition	990215
2.0	•Explanations of "1. Organization" of "BEFORE USING THIS MANUAL" are partly revised. •Page 1-2; Explanations of "●Power dissipation" of "FEATURES" are partly eliminated. •Page 1-2; Explanations of "●Memory expansion possible" of "FEATURES" are partly revised. •Page 1-2; Value of "●Program/Erase voltage" of "<Flash memory mode>" is revised. •Page 1-2; "Operating temperature range" of "<Flash memory mode>" is added. •Page 1-2; Explanations of "■Notes" are partly revised. •Page 1-2; Explanations of "APPLICATION" are partly added. •Page 1-3; Product name and note into Figure 1 are partly added. •Page 1-3; Note into Figure 2 is added. •Page 1-4; Figure 3 is added. •Page 1-5; Figure 4 is partly revised. •Page 1-8; Figure 5 is partly revised. •Page 1-9; Explanations of "Packages" are partly added. •Page 1-9; Figure 6 is partly revised. •Page 1-9; Table 3 is partly added. •Page 1-13; Figure 9 is partly revised. •Page 1-14; Notes into Figure 10 are partly revised. •Page 1-16; "Related SFRs" of "$P4_2$/INT_0/OBF_{00}, $P4_3$/INT_1/OBF_{01}" into Table 6 are partly added. •Page 1-42; "[Port Control Register 2 (PCTL2)]" are added. •Page 1-46; Explanations of "Bit 5" of "[I^2c Clock Control Register (S2)]" are partly revised. •Page 1-48; Bit name of bit 4 of "[I^2c Status Register (S1)]" is revised. •Page 1-49; Bit name of bit 4 into Figure 41 is revised. •Page 1-54; Explanations of "(3) RESTART condition generating procedure" are partly revised. •Page 1-54; "(6) STOP condition input at 7th clock pulse" is added. •Page 1-54; "(7) ES0 bit switch" is added. •Page 1-55; Figure 50 is partly revised.	000922

Rev. No.	Revision Description	Rev. date
2.0	•Page 1-55; Figure 50 is partly revised. •Page 1-60; Explanations of "RESET CIRCUIT" are partly revised. •Page 1-60; Explanations of note into Figure 57 are added. •Page 1-63; Note 2 into Figure 62 is added. •Page 1-64; Figure 63 is partly revised. •Page 1-65; Explanations of "PROCESSOR MODE" are partly revised. •Page 1-65; Explanations of "(2) Memory expansion mode" are partly added. •Page 1-65; Explanations of "(3) Microprocessor mode" are partly revised. •Page 1-65; Explanations into Figure 64 are partly eliminated. •Page 1-65; Note into Figure 65 is partly revised. •Page 1-66; Explanations of "BUS CONTROL AT MEMORY EXPANSION" are partly revised. •Page 1-69; Explanations of CNVss into Table 22 are partly revised. •Page 1-70; Figure 68 is partly revised. •Page 1-78; Figure 74 is partly revised. •Page 1-79; Explanations of CNVss into Table 27 are partly revised. •Page 1-83; Note is added. •Page 1-85; Explanations of "Functional Outline" of "(3) Flash memory mode 3 (CPU reprogram -ming mode)" are partly added. •Page 1-85; Note into Figure 81 is partly eliminated. •Page 1-86; Explanations of "<Beginning procedure>" of "●CPU reprogramming mode operation procedure" are partly eliminated. •Page 1-86; Figure 83 is partly revised. •Page 1-89; Explanations of "A-D Converter" of "NOTES ON PROGRAMMING" are partly eliminated. •Page 1-90; Explanations of "Handling of Power Source Pins" of "NOTES ON USAGE" are partly revised. •Page 1-90; "Erasing of Flash memory version" is added.	000922

Rev. No.	Revision Description	Rev. date
2.0	•Page 1-90; Explanations of "DATA REQUIRED FOR One Time PROM PROGRAMMING ORDERS" are partly added. •Page 1-91; "Interrupt" of "FUNCTIONAL DESCRIPTION SUPPLEMENT" is eliminated. •Page 1-91; "Timing After Interrupt" of "FUNCTIONAL DESCRIPTION SUPPLEMENT" is eliminated. •"2.2 Interrupt" is added. •"2.8 D-A converter" is added. •"2.12 Clock generating circuit" is added. •"2.13 Standby function" is added. •"2.15 Flash memory" is added. •Page 2-4; Bit attributes into Figure 2.1.4 are partly revised. •Page 2-4; Bit attributes into Figure 2.1.5 are partly revised. •Page 2-5; Explanations of "2.1.3 Port P4/P7 input register" are partly added. •Page 2-6; Explanations of "●Reason" of "(1) Notes in stand-by state" are partly revised. •Page 2-7; Explanations of "②Input ports" and "③I/O ports" of "(1) Terminate unused pins" are partly revised. •Page 2-24; Figure 2.3.1 is partly revised. •Page 2-27; Figure 2.3.6 is added. •Page 2-31; Figure 2.3.12 is partly revised. •Page 2-32; Figure 2.3.13 is partly revised. •Page 2-41; Explanations of "2.3.4 Notes on timer" are partly revised. •Page 2-75; Explanations of "(7) Transmit interrupt request when transmit enable bit is set" are partly revised. •Page 2-75; "(8) Transmit data writing" is added. •Page 2-87; Clause name and explanations of "2.5.6 I^2C-BUS communication usage example" are partly revised. •Page 2-88; Figure 2.5.17 is partly revised. •Page 2-104; Explanations of "(2) Procedure for generating START condition" are partly added.	000922

REVISION DESCRIPTION LIST	3886 GROUP USER'S MANUAL

Rev. No.	Revision Description	Rev. date
2.0	•Page 2-104; Sub clause name and explanations of "(3) Procedure for generating RESTART condition" are partly revised. •Page 2-105; Explanations of "(6) STOP condition input at 7th clock pulse" are partly revised. •Page 2-105; Clause of "Notes on programming for SMBUS interface" in Rev.1.0 is eliminated. •Page 2-118; Explanations of note into Figure 2.7.9 are partly revised. •Page 2-120; Explanations of "(2) A-D converter power source pin" of "2.7.4 Notes on A-D converter" are partly eliminated. •Page 2-120; Explanations of "(3) Clock frequency during A-D conversion" are partly eliminated. •Page 2-128; Figure 2.9.1 is partly revised. •Page 2-129; Figure 2.9.3 is partly revised. •Page 2-132; Figure 2.9.8 is added. •Page 2-138; Figure 2.10.3 is partly revised. •Page 2-139; Figure 2.10.4 is partly revised. •Page 2-141; Figure 2.11.2 is partly revised. •Page 2-150; Explanations of "(3) Notes on using stop mode" are partly revised. •Page 2-154; Figure 2.14.2 is partly revised. •Page 2-156; Figure 2.14.4 is partly revised. •Page 2-156; Figure 2.14.5 is partly revised. •Page 2-157; Figure 2.14.6 is partly revised. •Page 2-160; Figure 2.14.9 is partly revised. •Page 2-160; Figure 2.14.10 is partly revised. •Page 2-161; Figure 2.14.11 is partly revised. •Page 2-165; Table 2.15.2 is partly revised. •Paragraph of "Mask ROM confirmation" in Rev.1.0 is eliminated. •Paragraph of "ROM programming confirmation form" in Rev.1.0 is eliminated. •Paragraph of "Mark specification form" in Rev.1.0 is eliminated.	000922

Rev. No.	Revision Description	Rev. date
2.0	For the mask ROM confirmation form, the ROM programming confirmation form, and the mark specifications, refer to the "Mitsubishi MCU Technical Information" Homepage. *Data required for ROM orders (mask ROM confirmation forms, ROM programming confirmation forms) http://www.infomicom.mesc.co.jp/38000/38ordere.htm *Mark specification forms http://www.infomicom.mesc.co.jp/mela/markform.htm •Page 3-8; Limit of tw($\overline{\text{RESET}}$) into Table 3.1.11 is revised. •Page 3-9; Limit of tw($\overline{\text{RESET}}$) into Table 3.1.12 is revised. •Page 3-18; Figure 3.2.1 is partly revised. •Page 3-18; Figure 3.2.2 is revised. •Page 3-19; Figure 3.2.3 is revised. •Page 3-19; Figure 3.2.4 is revised. •Page 3-20; Figure 3.2.5 is revised. •Page 3-20; Figure 3.2.6 is revised. •Page 3-21; Figure 3.2.7 is revised. •Page 3-24; Figure 3.2.12 is partly revised. •Page 3-28; Explanations of "■Reason" of "(1) Notes in stand-by state" of "3.3.1 Notes on input and output pins" are partly added. •Page 3-29; Explanations of "②Input ports" of "(1) Terminate unused pins" of "3.3.2 Termination of unused pins" is partly revised. •Page 3-29; Explanations of "③I/O ports" of "(1) Terminate unused pins" of "3.3.2 Termination of unused pins" is partly revised. •Page 3-30; Sub clause of "Setting of interrupt request bit and interrupt enable bit" in Rev. 1.0 is eliminated. •Page 3-31; "(3) Change of relevant register setting" of "3.3.3 Notes on interrupts" is added. •Page 3-34; Explanations of "(5) Data transmission control with referring to transmit shift register completion flag" are partly added.	000922

REVISION DESCRIPTION LIST	3886 GROUP USER'S MANUAL

Rev. No.	Revision Description	Rev. date
2.0	•Page 3-34; Explanations of "(7) Transmit interrupt request when transmit enable bit is set" are partly revised. •Page 3-35; Explanations of "(2) Procedure for generating START condition using multi-master" are partly added. •Page 3-36; Explanations of "(3) Procedure for generating RESTART condition" are partly added. •Page 3-36; Sub clause of "STOP condition generating procedure in master" is eliminated. •Page 3-36; Explanations of "(6) STOP condition input at 7th clock pulse" are partly added. •Page 3-37; Explanations of "(2) A-D converter power source pin" are partly revised. •Page 3-37; Explanations of "(3) Clock frequency during A-D conversion" are partly revised. •Page 3-37; "3.3.9 Notes on D-A converter" is added. •Page 3-38; "3.3.12 Notes on CPU reprogramming mode" is added. •Page 3-38; "3.3.13 Notes on using stop mode" is added. •Page 3-39; "3.3.14 Notes on wait mode" is added. •Page 3-39; Explanations of "3.3.16 Notes on restarting oscillation" are partly revised. •Page 3-41; Figure 3.3.10 is partly revised. •Page 3-50; Figure 3.5.1 is partly revised. •Page 3-50; Figure 3.5.2 is partly revised. •Page 3-62; Bit attributes into Figure 3.5.22 are partly revised. •Page 3-64; Bit attributes into Figure 3.5.25 are partly revised. •Page 3-64; Bit attributes into Figure 3.5.26 are partly revised. •Page 3-70; Figure 3.5.37 is partly revised. •Page 3-73; Figure 3.5.42 is added. •Page 3-74; Figure 3.5.43 is added. •Page 3-87; Product name and note into Figure 3.10.1 are partly added. •Page 3-87; Note into Figure 3.10.2 is added. •Page 3-88; Figure 3.10.3 is added.	000922

Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 3886 Group.
After reading this manual, the user should have a through knowledge of the functions and features of the 3886 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 "Mitsubishi Microcomputer Development Support Tools" Homepage (http://www.tool-spt.mesc.co.jp/index_e.htm).

BEFORE USING THIS 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. Chapter 3 also includes necessary information for systems development. You must refer to that chapter.

1. Organization

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

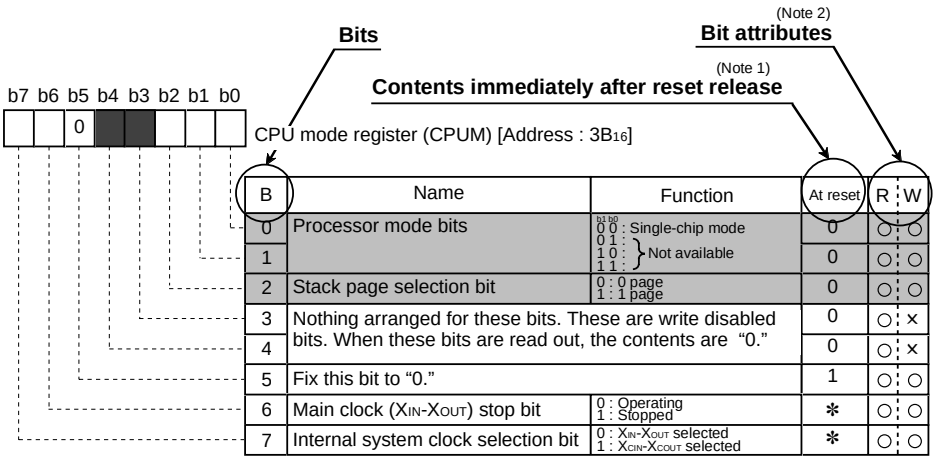
● **CHAPTER 2 APPLICATION**
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of relevant registers.

● **CHAPTER 3 APPENDIX**
This chapter includes necessary information for systems development using the microcomputer, such as the electrical characteristics, the notes, and the list of registers.

✻For the mask ROM confirmation form, the ROM programming confirmation form, and the mark specifications, refer to the "Mitsubishi MCU Technical Information" Homepage (http:// www.infomicom.mesc.co.jp/).

2. Structure of register

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



B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : 1 0 : 1 1 : } Not available	0	○	○
1			0	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Nothing arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0."		0	○	×
4			0	○	×
5	Fix this bit to "0."		1	○	○
6	Main clock (X_{IN}-X_{OUT}) stop bit	0 : Operating 1 : Stopped	✻	○	○
7	Internal system clock selection bit	0 : X_{IN}-X_{OUT} selected 1 : X_{CIN}-X_{COUT} selected	✻	○	○

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

Note 1:. Contents immediately after reset release
0..... "0" at reset release
1..... "1" at reset release
?..... Undefined at reset release
✻.....Contents determined by option at reset release

Note 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

CHAPTER 3 APPENDIX

Table of contents

List of figures

CHAPTER 1 HARDWARE

List of figures

CHAPTER 2 APPLICATION

List of figures

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

DESCRIPTION

The 3886 group is the 8-bit microcomputer based on the 740 family core technology.

The 3886 group is designed for controlling systems that require analog signal processing and include two serial I/O functions, A-D converters, D-A converters, system data bus interface function, watchdog timer, and comparator circuit.

The multi-master I^2C-BUS interface can be added by option.

FEATURES

<Microcomputer mode>

- Basic machine-language instructions 71
- Minimum instruction execution time ..... 0.4 µs
  (at 10 MHz oscillation frequency)
- Memory size
  ROM ..... 32K to 60K bytes
  RAM ..... 1024 to 2048 bytes
- Programmable input/output ports 72
- Software pull-up resistors ..... Built-in
- Interrupts ..... 21 sources, 16 vectors
  (Included key input interrupt)
- Timers ..... 8-bit × 4
- Serial I/O1 ..... 8-bit × 1(UART or Clock-synchronized)
- Serial I/O2 ..... 8-bit × 1(Clock-synchronized)
- PWM output circuit ..... 14-bit × 2
- Bus interface ..... 2 bytes
- I^2C-BUS interface (option) ..... 1 channel
- A-D converter ..... 10-bit × 8 channels
- D-A converter ..... 8-bit × 2 channels
- Comparator circuit ..... 8 channels
- Watchdog timer ..... 16-bit × 1
- Clock generating circuit ..... Built-in 2 circuits
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage
  In high-speed mode ..... 4.0 to 5.5 V
  (at 10 MHz oscillation frequency)
  In middle-speed mode ..... 2.7 to 5.5 V(*)
  (at 10 MHz oscillation frequency)
  In low-speed mode ..... 2.7 to 5.5 V (*)
  (at 32 kHz oscillation frequency)
  (*: 4.0 to 5.5 V for Flash memory version)
- Power dissipation
  In high-speed mode ..... 40 mW
  (at 10 MHz oscillation frequency, at 5 V power source voltage)
  In low-speed mode ..... 60 µW
  (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Memory expansion possible (only for M38867M8A/E8A)
- Operating temperature range ..... –20 to 85°C

<Flash memory mode>

- Supply voltage (at programming/erasing) ..... V_{CC} = 5 V ± 10 %
- Program/Erase voltage ..... V_{PP} = 11.7 to 12.6 V
- Programming method ..... Programming in unit of byte
- Erasing method
  Batch erasing ..... Parallel/Serial I/O mode
  Block erasing ..... CPU reprogramming mode
- Program/Erase control by software command
- Number of times for programming/erasing 100
- Operating temperature range (at programming/erasing)
  ..... Normal temperature

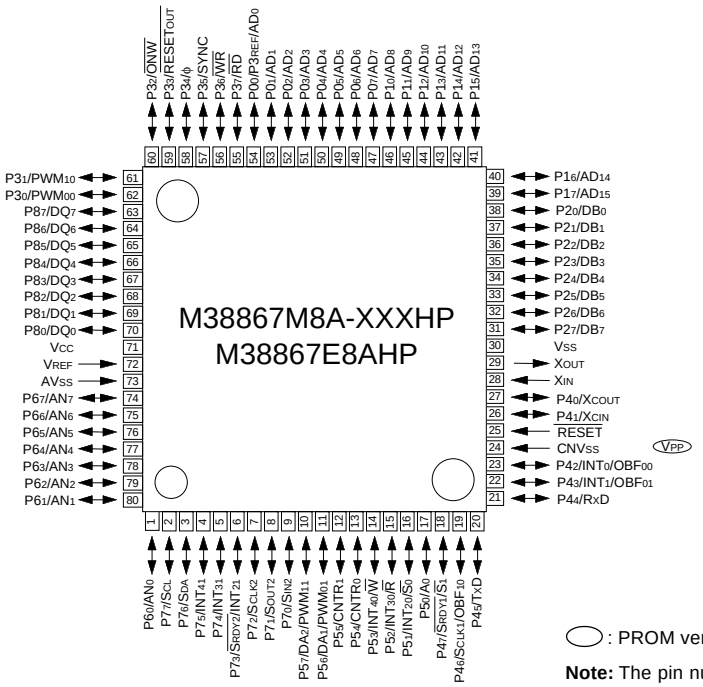
■Notes

1. The flash memory version cannot be used for application embedded in the MCU card.
2. Power source voltage Vcc of the flash memory version is 4.0 to 5.5 V.

APPLICATION

Household product, consumer electronics, communications, note book PC, etc.

PIN CONFIGURATION (TOP VIEW)



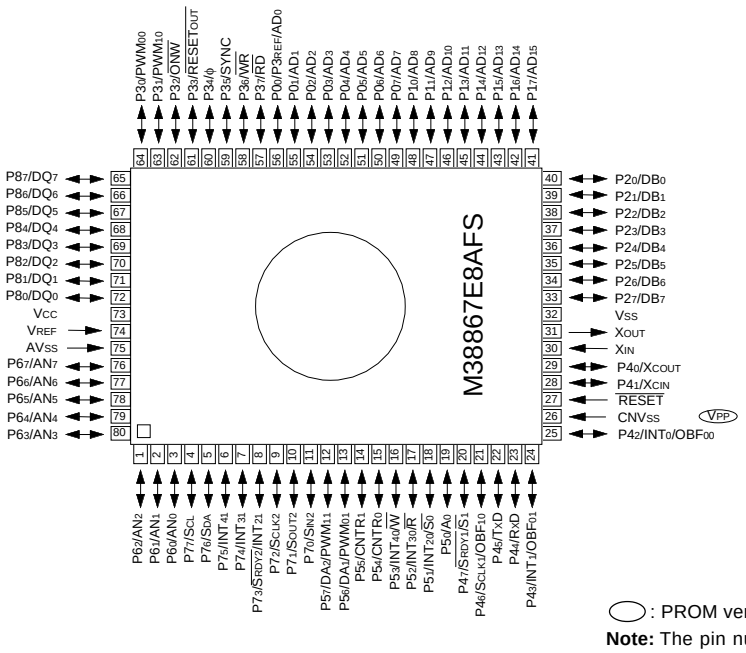
Package type : 80P6Q-A

: PROM version

Note: The pin number and the position of the function pin may change by the kind of package.

Fig. 1 M38867M8A-XXXHP, M38867E8AHP pin configuration

PIN CONFIGURATION (TOP VIEW)



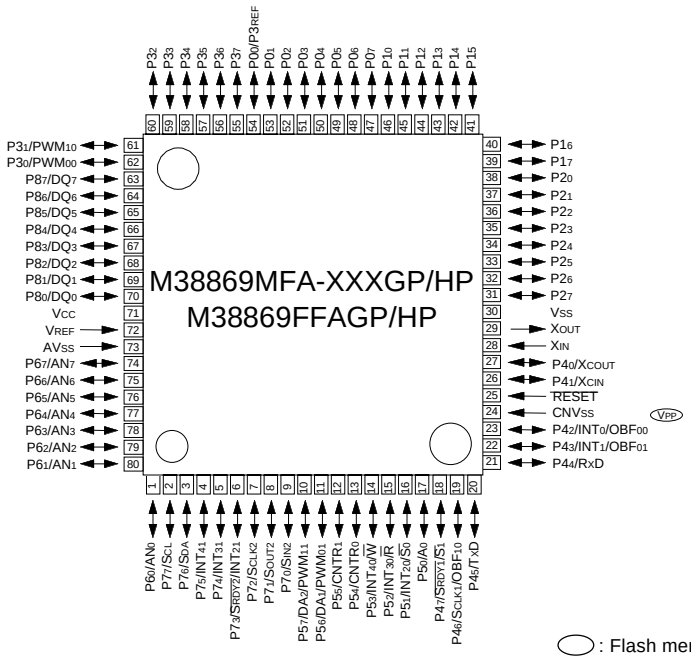
Package type : 80D0

: PROM version

Note: The pin number and the position of the function pin may change by the kind of package.

Fig. 2 M38867E8AFS pin configuration

PIN CONFIGURATION (TOP VIEW)



: Flash memory version

Note: The pin number and the position of the function pin may change by the kind of package.

Package type : 80P6S-A/80P6Q-A

Fig. 3 M38869MFA-XXXGP/HP, M38869FFAGP/HP pin configuration

FUNCTIONAL BLOCK

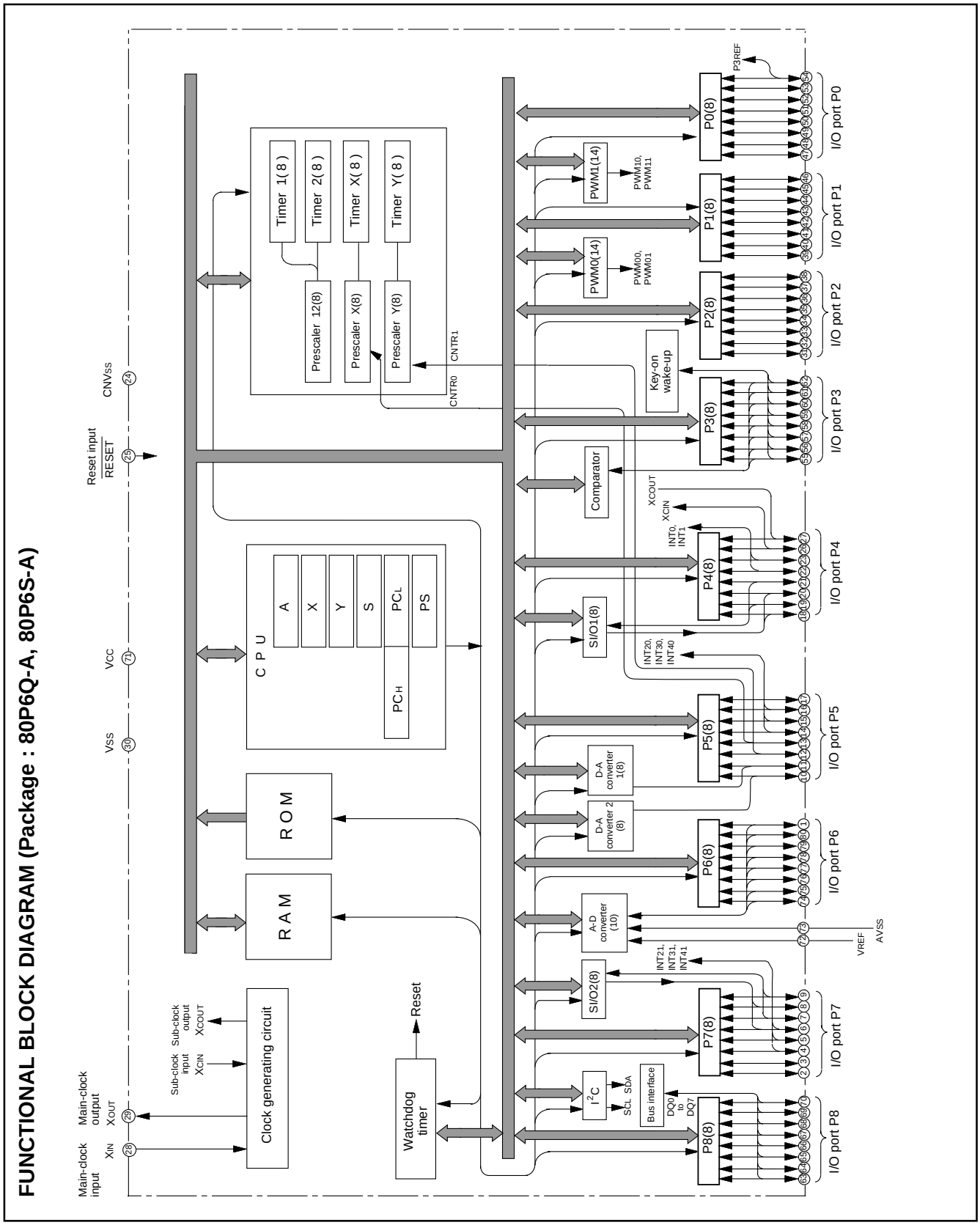


Fig. 4 Functional block diagram

PIN DESCRIPTION

Table 1 Pin description (1)

Pin	Name	Functions	Function except a port function
VCC, VSS	Power source	•Apply voltage of 2.7 V – 5.5 V to Vcc, and 0 V to Vss. •In the flash memory version, apply voltage of 4.0 V – 5.5 V to Vcc, and 0 V to Vss.	
CNVSS	CNVSS input	•This pin controls the operation mode of the chip. •Normally connected to VSS. •If this pin is connected to Vcc, the internal ROM is inhibited and an external memory is accessed. •In the flash memory version, connected to VSS. •In the EPROM version or the flash memory version, this pin functions as the VPP power source input pin.	
VREF	Reference voltage	•Reference voltage input pin for A-D and D-A converters.	
AVSS	Analog power source	•Analog power source input pin for A-D and D-A converters. •Connect to VSS.	
$\overline{\text{RESET}}$	Reset input	•Reset input pin for active “L”.	
XIN	Clock input	•Input and output pins for the clock generating circuit. •Connect a ceramic resonator or quartz-crystal oscillator between the XIN and XOUT pins to set the oscillation frequency. •When an external clock is used, connect the clock source to the XIN pin and leave the XOUT pin open.	
XOUT	Clock output		
P00/P3REF	I/O port P0	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •When the external memory is used, these pins are used as the address bus. •CMOS compatible input level. •CMOS 3-state output structure or N-channel open-drain output structure.	•Comparator reference power source input pin
P01–P07			
P10–P17	I/O port P1	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •When the external memory is used, these pins are used as the address bus. •CMOS compatible input level. •CMOS 3-state output structure or N-channel open-drain output structure.	
P20–P27	I/O port P2	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •When the external memory is used, these pins are used as the data bus. •CMOS compatible input level. •CMOS 3-state output structure. •P24 to P27 (4 bits) are enabled to output large current for LED drive (only in single-chip mode).	
P30/PWM00 P31/PWM10	I/O port P3	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •When the external memory is used, these pins are used as the control bus. •CMOS compatible input level. •CMOS 3-state output structure. •These pins function as key-on wake-up and comparator input. •These pins are enabled to control pull-up.	•Key-on wake-up input pin •Comparator input pin •PWM output pin
P32–P37			•Key-on wake-up input pin •Comparator input pin

Table 2 Pin description (2)

Pin	Name	Functions	Function except a port function
P40/XCOUT P41/XCIN	I/O port P4	•8-bit I/O port with the same function as port P0. <Input level> P40, P41 : CMOS input level P42–P46 : CMOS compatible input level or TTL input level P47 : CMOS compatible input level or TTL input level in the bus interface function <Output structure> P40, P41, P47 : CMOS 3-state output structure P42–P46 : CMOS 3-state output structure or N-channel open-drain output structure •Regardless of input or output port, P42 to P46 can be input every pin level.	•Sub-clock generating circuit I/O pins (Connect a resonator.)
P42/INT0/OBF00 P43/INT1/OBF01			•Interrupt input pins •Bus interface function pins
P44/RxD P45/TxD			•Serial I/O1 function pins
P46/SCLK1/OBF10 P47/$\overline{SRDY1}$/$\overline{S1}$		•When P42 and P43 are used as output port, the function which makes P42 and P43 clear to "0" when the host CPU reads the output data bus buffer 0 can be added.	•Serial I/O1 function pins •Bus interface function pins
P50/A0	I/O port P5	•8-bit I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure. •P50 to P53 can be switched between CMOS compatible input level or TTL input level in the bus interface function.	•Bus interface function pins
P51/INT20/$\overline{S0}$ P52/INT30/$\overline{R}$ P53/INT40/$\overline{W}$			•Interrupt input pins •Bus interface function pins
P54/CNTR0 P55/CNTR1			•Timer X, timer Y function pins
P56/DA1/PWM01 P57/DA2/PWM11			•D-A converter output pin •PWM output pin
P60/AN0–P67/AN7	I/O port P6	•8-bit I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure.	•A-D converter output pin
P70/SIN2 P71/SOUT2 P72/SCLK2	I/O port P7	•8-bit I/O port with the same function as port P0. P70–P75 : CMOS compatible input level or TTL input level P76, P77 : CMOS compatible input level or SMBUS input level in the I^2C-BUS interface function, N-channel open-drain output structure •Regardless of input or output port, P70 to P75 can be input every pin level.	•Serial I/O2 function pin
P73/$\overline{SRDY2}$/INT21			•Serial I/O2 function pin •Interrupt input pin
P74/INT31 P75/INT41			•Interrupt input pin
P76/SDA P77/SCL			•I^2C-BUS interface function pin
P80/DQ0–P87/DQ7	I/O port P8	•8-bit I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure. •CMOS compatible input level or TTL input level in the bus interface function.	•Bus interface function pin

PART NUMBERING

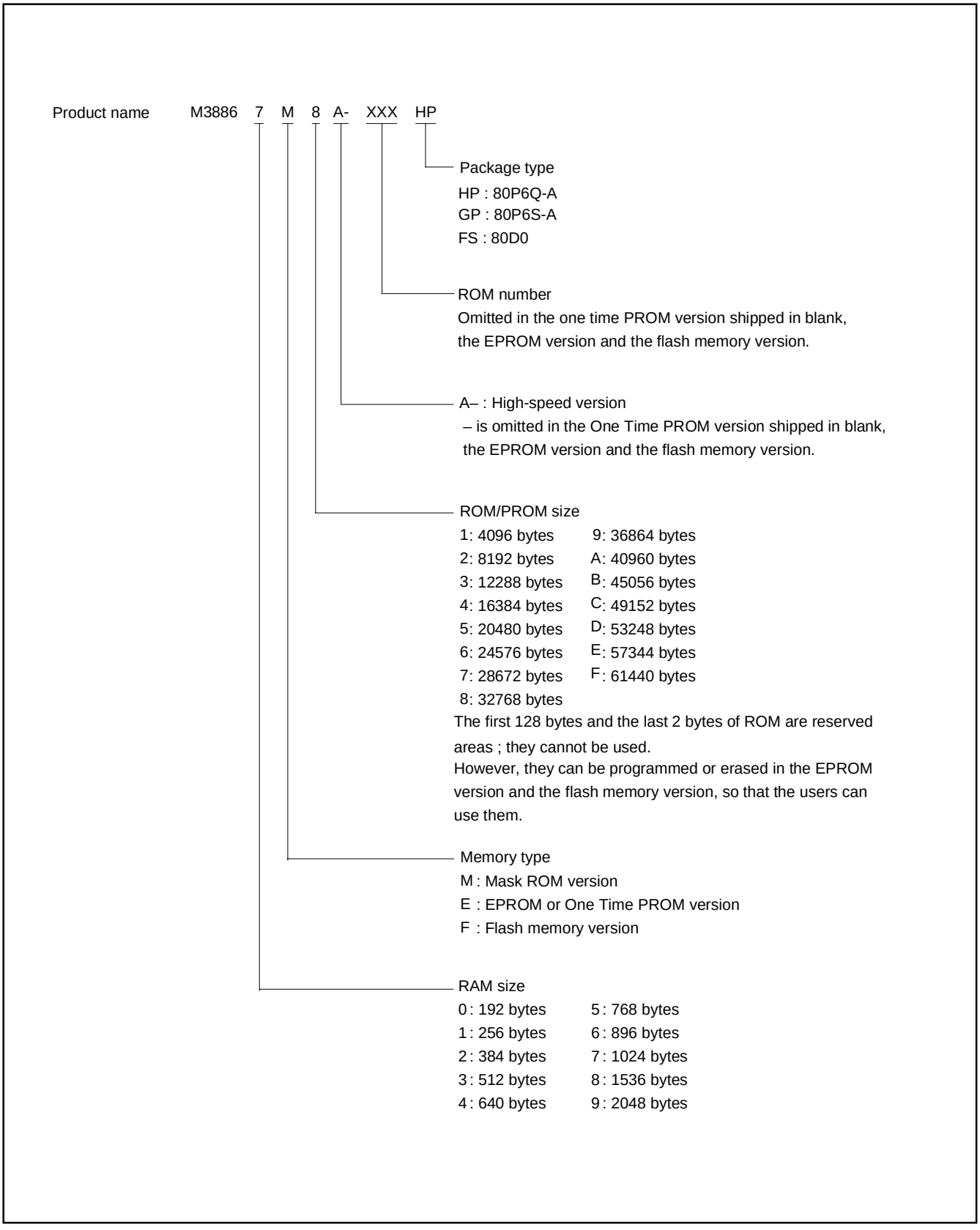


Fig. 5 Part numbering

GROUP EXPANSION

Mitsubishi plans to expand the 3886 group as follows.

Memory Type

Support for mask ROM, One Time PROM, EPROM and flash memory version.

Memory Size

ROM size ..... 32 K to 60 K bytes
RAM size ..... 1024 to 2048 bytes

Packages

80P6Q-A ..... 0.5 mm-pitch plastic molded LQFP
80P6S-A ..... 0.65mm-pitch plastic molded QFP
80D0 ..... 0.8 mm-pitch ceramic LCC (EPROM version)

The pin number and the position of the function pin may change by the kind of package.

Memory Expansion

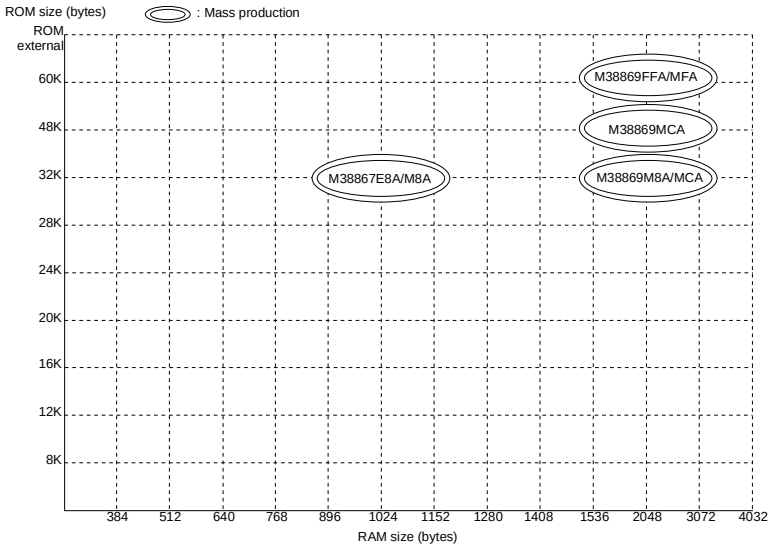


Fig. 6 Memory expansion plan

Currently products are listed below.

Table 3 Support products **As of Sep. 2000**

Product name	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38867M8A-XXXHP	32768 (32638)	1024	80P6Q-A	Mask ROM version
M38867E8A-XXXHP				One Time PROM version
M38867E8AHP				One Time PROM version (blank)
M38867E8AFS			80D0	EPROM version
M38869M8A-XXXHP		2048	80P6Q-A	Mask ROM version
M38869M8A-XXXGP			80P6S-A	
M38869MCA-XXXHP	49152 (49022)		80P6Q-A	
M38869MCA-XXXGP			80P6S-A	
M38869MFA-XXXHP	61440 (61310)		80P6Q-A	
M38869MFA-XXXGP			80P6S-A	
M38869FFAHP			80P6Q-A	Flash memory version
M38869FFAGP			80P6S-A	

FUNCTIONAL DESCRIPTION
Central Processing Unit (CPU)

The 3886 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.

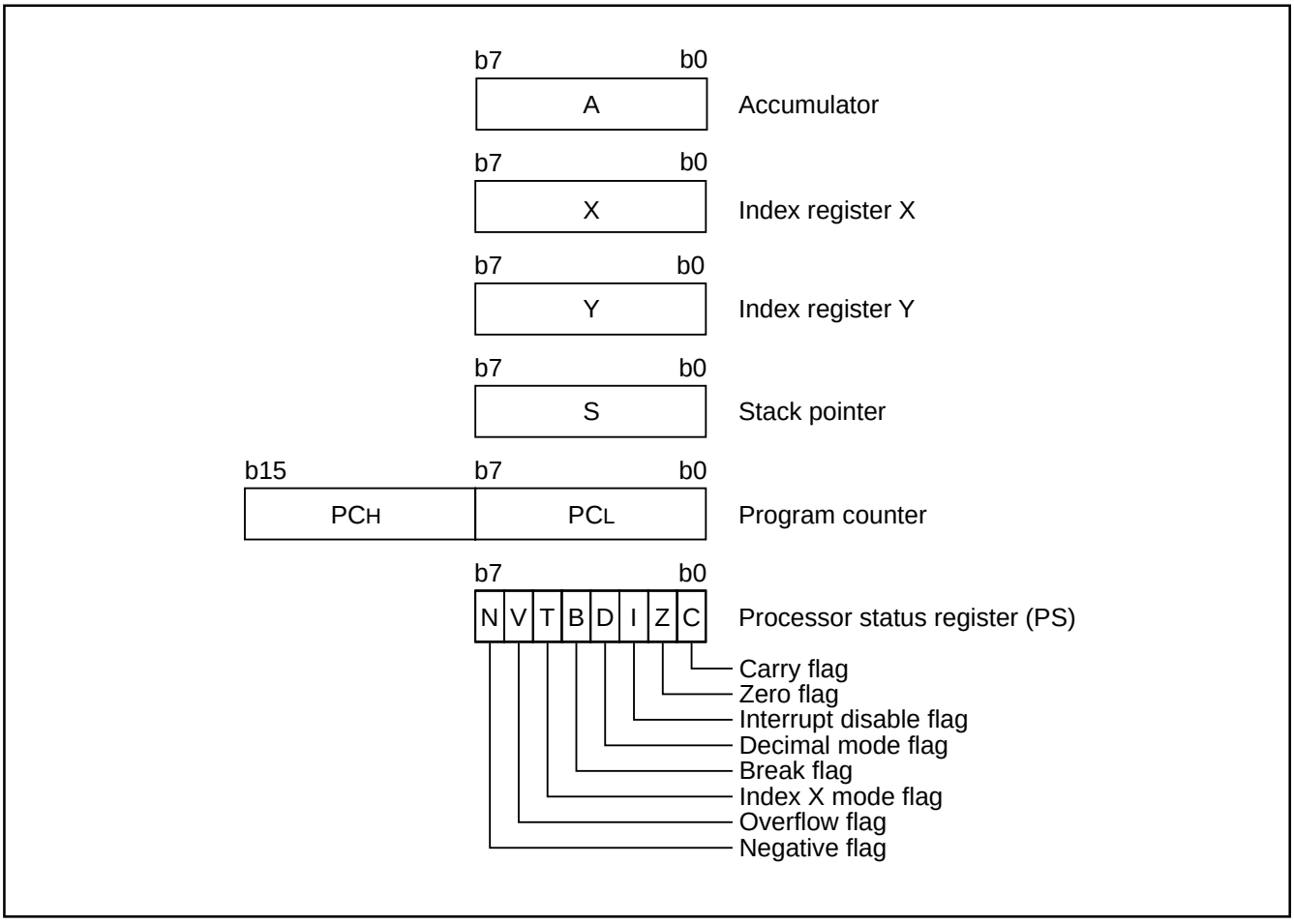


Fig. 7 740 Family CPU register structure

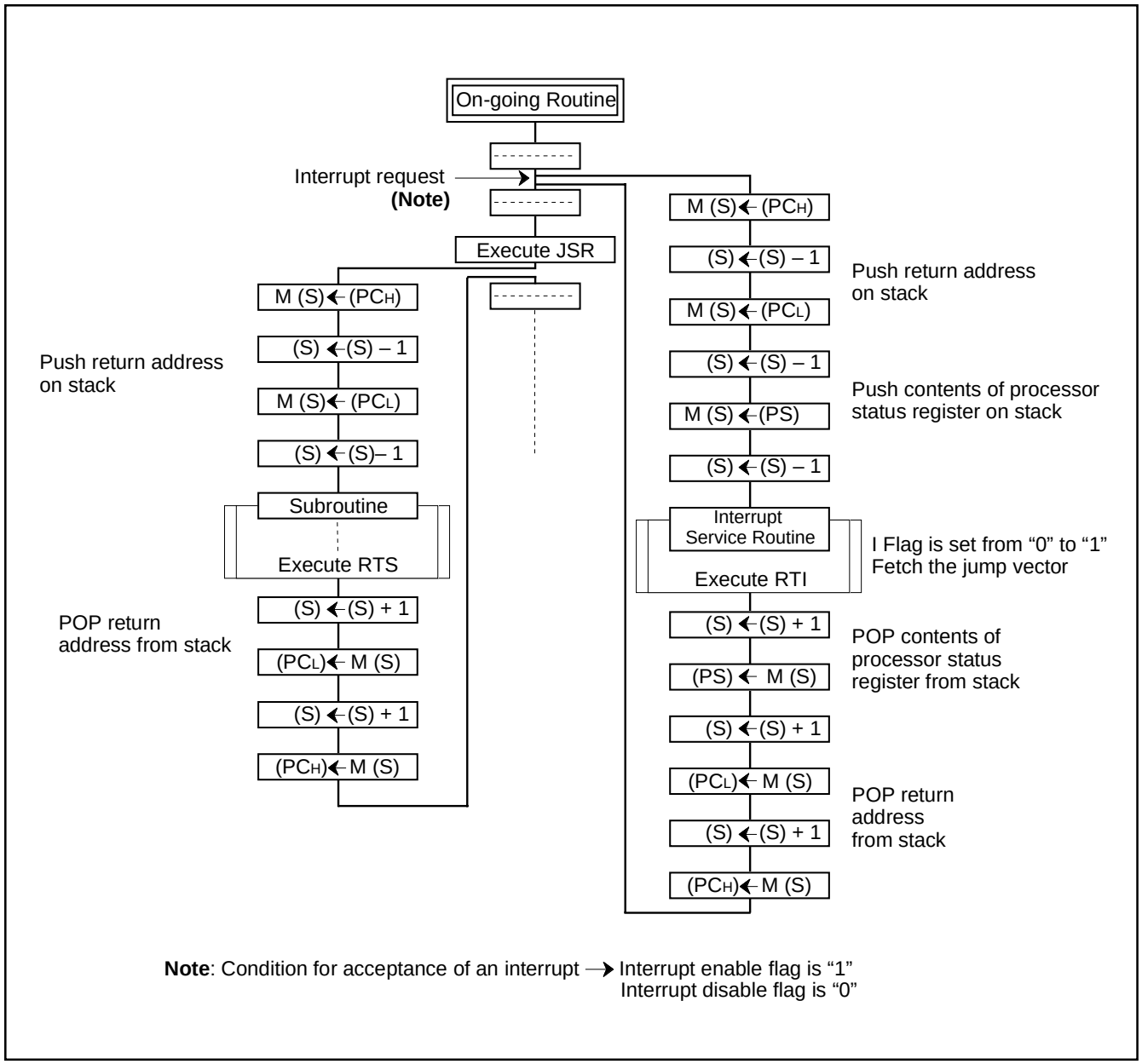


Fig. 8 Register push and pop at interrupt generation and subroutine call

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

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

[Processor status register (PS)]

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

•Bit 0: Carry flag (C)

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

•Bit 1: Zero flag (Z)

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

•Bit 2: Interrupt disable flag (I)

The I flag disables all interrupts except for the interrupt generated by the BRK instruction.

Interrupts are disabled when the I flag is “1”.

•Bit 3: Decimal mode flag (D)

The D flag determines whether additions and subtractions are executed in binary or decimal. Binary arithmetic is executed when this flag is “0”; decimal arithmetic is executed when it is “1”. Decimal correction is automatic in decimal mode. Only the ADC and SBC instructions can 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, the processor mode bits specifying the chip operation mode, etc. The CPU mode register is allocated at address 003B16.

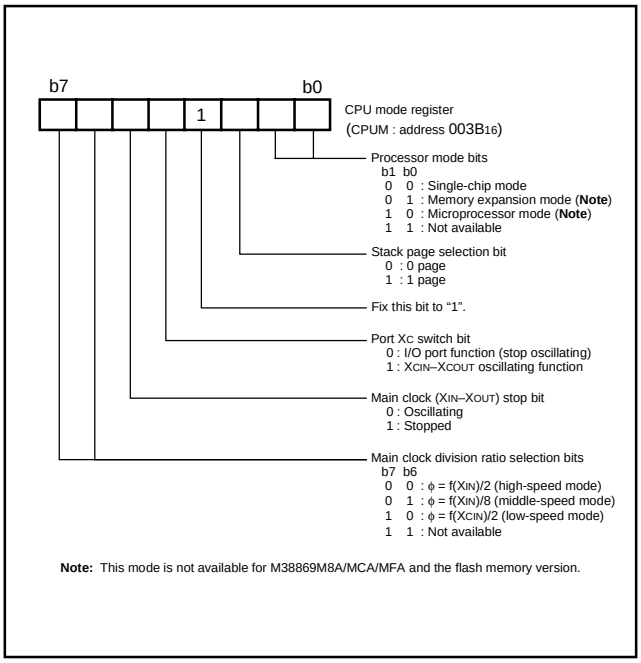


Fig. 9 Structure of CPU mode register

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. Program/Erase of the reserved ROM area is possible in the EPROM version and the flash memory version.

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

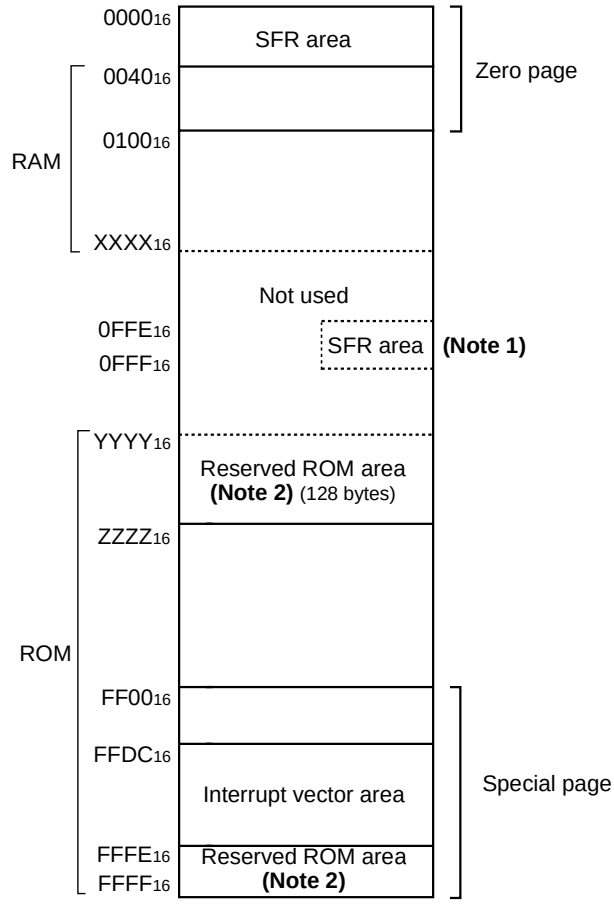
Access to this area with only 2 bytes is possible in the special page addressing mode.

RAM area

RAM size (bytes)	Address XXXX16
192	00FF16
256	013F16
384	01BF16
512	023F16
640	02BF16
768	033F16
896	03BF16
1024	043F16
1536	063F16
2048	083F16

ROM area

ROM size (bytes)	Address YYYY16	Address ZZZZ16
4096	F00016	F08016
8192	E00016	E08016
12288	D00016	D08016
16384	C00016	C08016
20480	B00016	B08016
24576	A00016	A08016
28672	900016	908016
32768	800016	808016
36864	700016	708016
40960	600016	608016
45056	500016	508016
49152	400016	408016
53248	300016	308016
57344	200016	208016
61440	100016	108016



Notes 1: This area is SFR in M38869FFA.
This area is Reserved in M38869MFA/MCA/M8A.
This area is not used in M38867M8A/E8A.
2: This area is usable in EPROM version and flash memory version.

Fig. 10 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}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	Port P5 (P5)
$000B_{16}$	Port P5 direction register (P5D)
$000C_{16}$	Port P6 (P6)
$000D_{16}$	Port P6 direction register (P6D)
$000E_{16}$	Port P7 (P7)
$000F_{16}$	Port P7 direction register (P7D)
0010_{16}	Port P8 (P8)/Port P4 input register (P4I)
0011_{16}	Port P8 direction register (P8D)/Port P7 input register (P7I)
0012_{16}	I^2C data shift register (S0)
0013_{16}	I^2C address register (S0D)
0014_{16}	I^2C status register (S1)
0015_{16}	I^2C control register (S1D)
0016_{16}	I^2C clock control register (S2)
0017_{16}	I^2C start/stop condition control register (S2D)
0018_{16}	Transmit/Receive buffer register (TB/RB)
0019_{16}	Serial I/O1 status register (SIO1STS)
$001A_{16}$	Serial I/O1 control register (SIO1CON)
$001B_{16}$	UART control register (UARTCON)
$001C_{16}$	Baud rate generator (BRG)
$001D_{16}$	Serial I/O2 control register (SIO2CON)
$001E_{16}$	Watchdog timer control register (WDTCON)
$001F_{16}$	Serial I/O2 register (SIO2)

0020_{16}	Prescaler 12 (PRE12)	
0021_{16}	Timer 1 (T1)	
0022_{16}	Timer 2 (T2)	
0023_{16}	Timer XY mode register (TM)	
0024_{16}	Prescaler X (PREX)	
0025_{16}	Timer X (TX)	
0026_{16}	Prescaler Y (PREY)	
0027_{16}	Timer Y (TY)	
0028_{16}	Data bus buffer register 0 (DBB0)	
0029_{16}	Data bus buffer status register 0 (DBBSTS0)	
$002A_{16}$	Data bus buffer control register (DBBCON)	
$002B_{16}$	Data bus buffer register 1 (DBB1)	
$002C_{16}$	Data bus buffer status register 1 (DBBSTS1)	
$002D_{16}$	Comparator data register (CMPD)	
$002E_{16}$	Port control register 1 (PCTL1)	
$002F_{16}$	Port control register 2 (PCTL2)	
0030_{16}	PWM0H register (PWM0H)	
0031_{16}	PWM0L register (PWM0L)	
0032_{16}	PWM1H register (PWM1H)	
0033_{16}	PWM1L register (PWM1L)	
0034_{16}	AD/DA control register (ADCON)	
0035_{16}	A-D conversion register 1 (AD1)	
0036_{16}	D-A1 conversion register (DA1)	
0037_{16}	D-A2 conversion register (DA2)	
0038_{16}	A-D conversion register 2 (AD2)	
0039_{16}	Interrupt source selection register (INTSEL)	
$003A_{16}$	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)	

$0FFE_{16}$	Flash memory control register (FCON)	**(Note)**
$0FFF_{16}$	Flash command register (FCMD)	**(Note)**

Note: Only for flash memory version

Fig. 11 Memory map of special function register (SFR)

I/O PORTS

The I/O ports have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, and each pin can be set to be input port or output port.

When "0" is written to the bit corresponding to a pin, that pin becomes an input pin. When "1" is written to that bit, that pin becomes an output pin.

If data is read from a pin which is set to output, the value of the port output latch is read, not the value of the pin itself. Pins set to input are floating. If a pin set to input is written to, only the port output latch is written to and the pin remains floating.

When the P8 function select bit of the port control register 2 (address $002F_{16}$) is set to "1", read from address 0010_{16} becomes the port P4 input register, and read from address 0011_{16} becomes the port P7 input register.

As the particular function, value of $P4_2$ to $P4_6$ pins and $P7_0$ to $P7_5$ pins can be read regardless of setting direction registers, by reading the port P4 input register (address 0010_{16}) or the port P7 input register (address 0011_{16}) respectively.

Table 6 I/O port function (1)

Pin	Name	Input/Output	I/O Structure	Non-Port Function	Related SFRs	Ref.No.
$P0_0$/$P3_{REF}$	Port P0	Input/output, individual bits	CMOS compatible input level CMOS 3-state output or N-channel open-drain output	Address low-order byte output Analog comparator power source input pin	CPU mode register Port control register 1 Serial I/O2 control register	(1)
$P0_1$–$P0_7$				Address low-order byte output	CPU mode register Port control register 1	(2)
$P1_0$–$P1_7$	Port P1			Address high-order byte output		
$P2_0$–$P2_7$	Port P2		CMOS compatible input level CMOS 3-state output	Data bus I/O	CPU mode register	(3)
$P3_0$/PWM_{00} $P3_1$/PWM_{10}	Port P3			Control signal I/O PWM output Key-on wake up input Comparator input	CPU mode register Port control register 1 AD/DA control register	(4) (5)
$P3_2$–$P3_7$				Control signal I/O Key-on wake up input Comparator input	CPU mode register Port control register 1	(6)
$P4_0$/X_{COUT} $P4_1$/X_{CIN}	Port P4			Sub-clock generating circuit	CPU mode register	(7) (8)
$P4_2$/INT_0/ OBF_{00} $P4_3$/INT_1/ OBF_{01}			CMOS compatible input level or TTL input level CMOS 3-state output or N-channel open-drain output	External interrupt input Bus interface function I/O	Interrupt edge selection register Port control register 2 Data bus buffer control register	(9) (10)
$P4_4$/RxD				Serial I/O1 function input	Serial I/O1 control register Port control register 2	(11)
$P4_5$/TxD				Serial I/O1 function output	Serial I/O1 control register UART control register Port control register 2	(12)
$P4_6$/S_{CLK1} /OBF_{10}				Serial I/O1 function I/O Bus interface function output	Serial I/O1 control register Data bus buffer control register Port control register 2	(13)
$P4_7$/$\overline{S_{RDY1}}$ /$\overline{S_1}$			CMOS compatible input level CMOS 3-state output (when selecting bus interface function) CMOS compatible input level or TTL input level	Serial I/O1 function output Bus interface function input	Serial I/O1 control register Data bus buffer control register	(14)

Table 7 I/O port function (2)

Pin	Name	Input/Output	I/O Format	Non-Port Function	Related SFRs	Ref.No.
P5$_0$/A$_0$	Port P5	Input/output, individual bits	CMOS compatible input level CMOS 3-state output (when selecting bus interface function) CMOS compatible input level or TTL input level	Bus interface function input	Data bus buffer control register	(15)
P5$_1$/INT$_{20}$ /$\overline{S_0}$ P5$_2$/INT$_{30}$ /$\overline{R}$ P5$_3$/INT$_{40}$ /$\overline{W}$				External interrupt input Bus interface function input	Interrupt edge selection register Data bus buffer control register	(16)
P5$_4$/CNTR$_0$ P5$_5$/CNTR$_1$			CMOS compatible input level CMOS 3-state output	Timer X, timer Y function I/O	Timer XY mode register	(17)
P5$_6$/DA$_1$/ PWM$_{01}$ P5$_7$/DA$_2$/ PWM$_{11}$				D-A converter output PWM output	AD/DA control register UART control register	(18) (19)
P6$_0$/AN$_0$– P6$_7$/AN$_7$	Port P6			A-D converter input	AD/DA control register	(20)
P7$_0$/S$_{IN2}$ P7$_1$/S$_{OUT2}$ P7$_2$/S$_{CLK2}$	Port P7		CMOS compatible input level or TTL input level N-channel open-drain output	Serial I/O2 function I/O	Serial I/O2 control register Port control register 2	(21) (22) (23)
P7$_3$/$\overline{S_{RDY2}}$/ INT$_{21}$				Serial I/O2 function output Bus interface function input	Serial I/O2 control register Port control register 2	(24)
P7$_4$/INT$_{31}$ P7$_5$/INT$_{41}$				External interrupt input	Interrupt edge selection register Port control register 2	(25)
P7$_6$/S$_{DA}$ P7$_7$/S$_{CL}$			CMOS compatible input level N-channel open-drain output (when selecting I^2C-BUS interface function) CMOS compatible input level or SMBUS input level	I^2C-BUS interface function I/O	I^2C control register	(26) (27)
P8$_0$/DQ$_0$– P8$_7$/DQ$_7$	Port P8		CMOS compatible input level CMOS 3-state output (when selecting bus interface function) CMOS compatible input level or TTL input level	Bus interface function I/O	Data bus buffer control register	(28)

Notes1: For details of the functions of ports P0 to P3 in modes other than single-chip mode, and how to use double-function ports as function I/O ports, refer to the applicable sections.

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

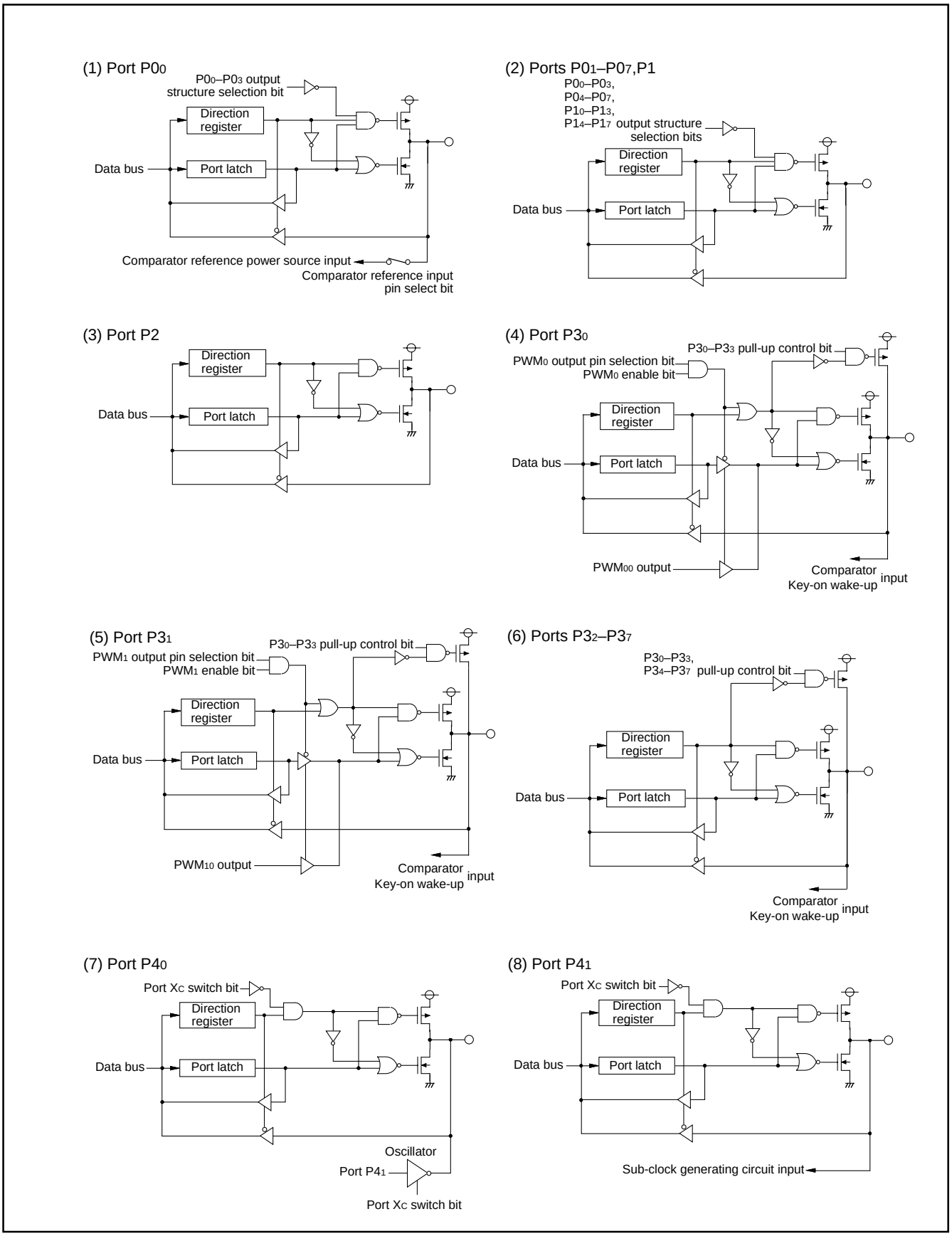
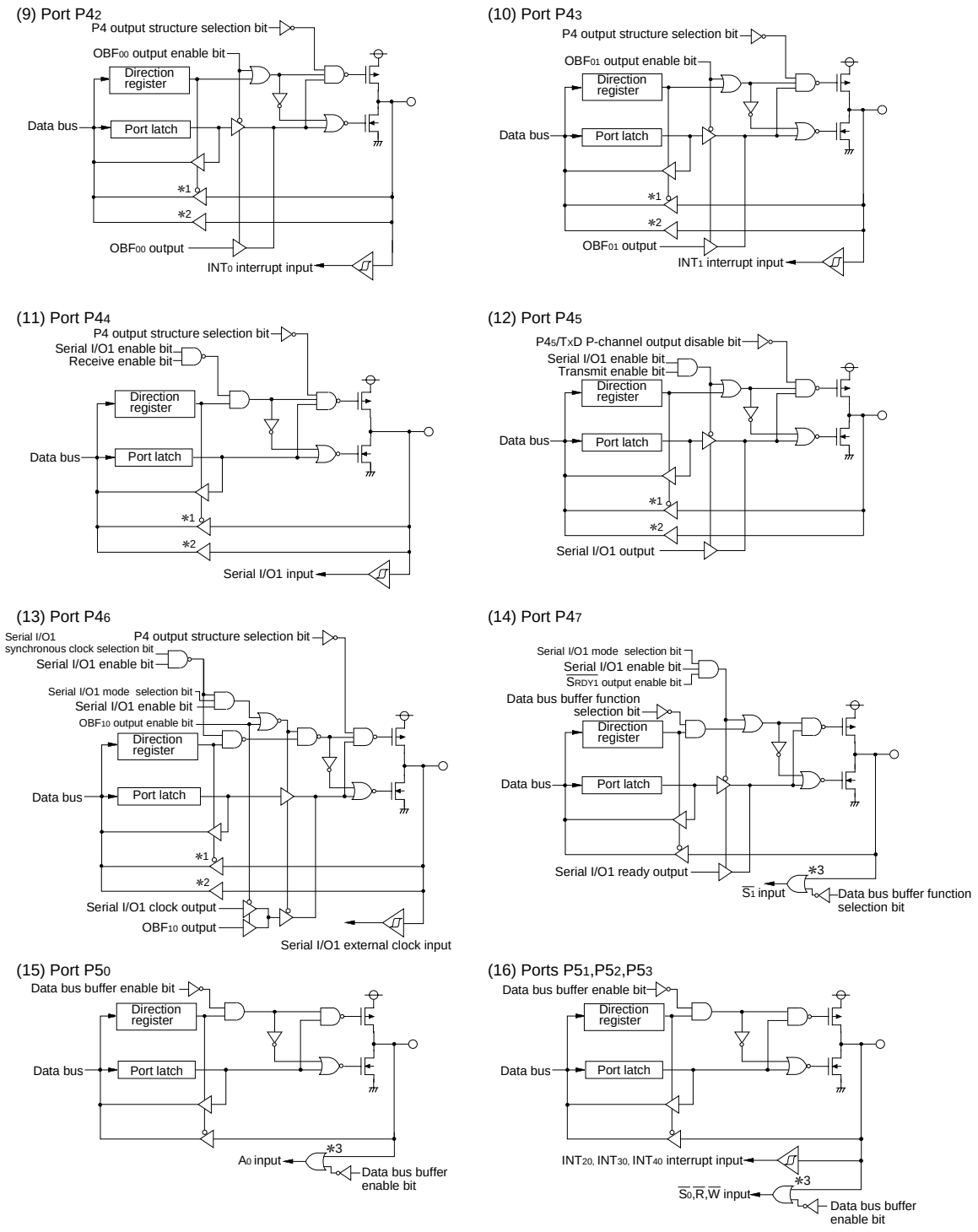
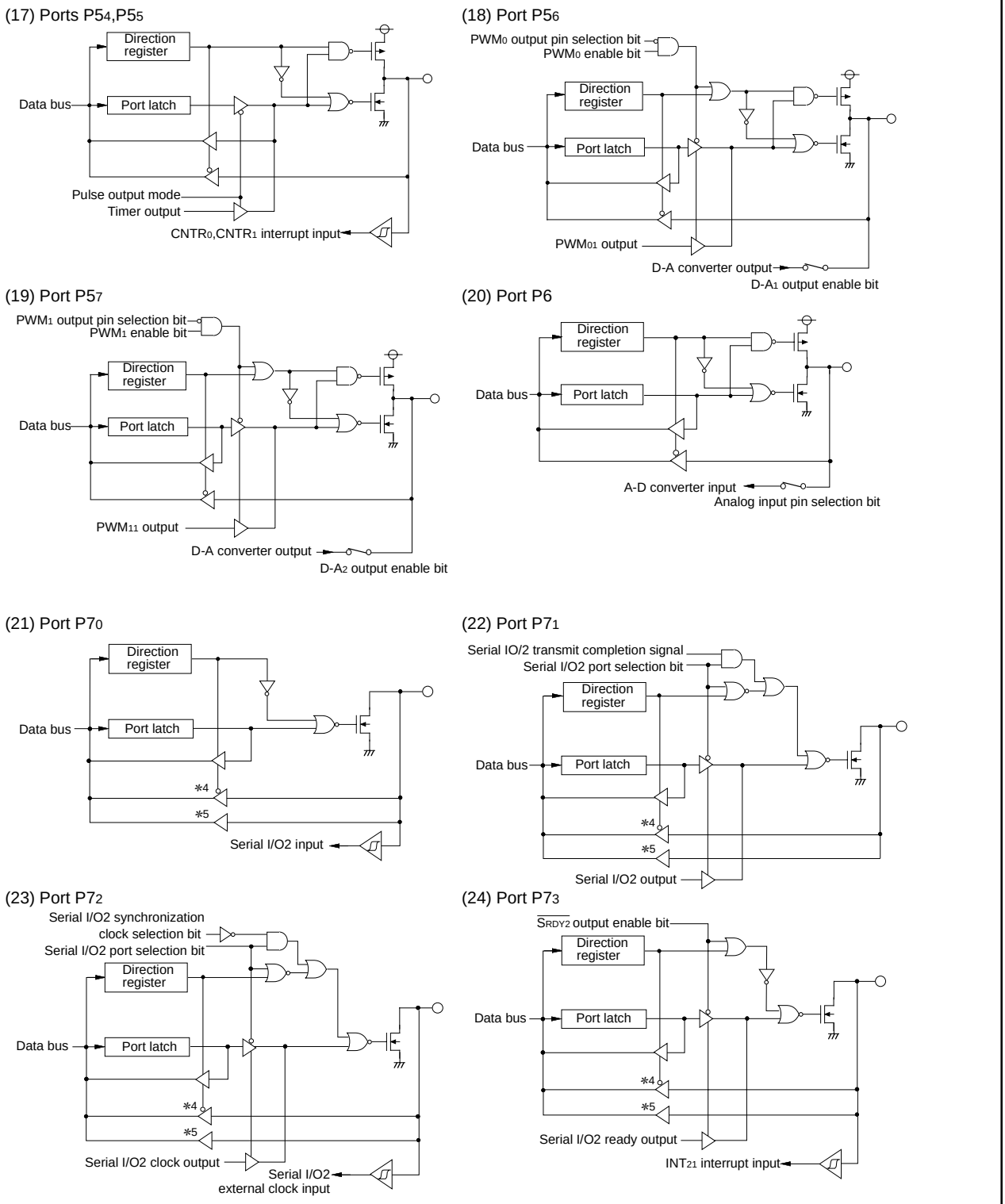


Fig. 12 Port block diagram (1)



*1. The input level can be switched between CMOS compatible input level and TTL level by the P4 input level selection bit of the port control register 2 (address 002F16).
*2. The input level can be switched between CMOS compatible input level and TTL level by the P4 input level selection bit of the port control register 2 (address 002F16).
The port P8 and port P4 input register can be switched by the P8 function selection bit of the port control register 2 (address 002F16).
*3. The input level can be switched between CMOS compatible input level and TTL level by the input level selection bit of the data bus buffer control register (address 002A16).

Fig. 13 Port block diagram (2)

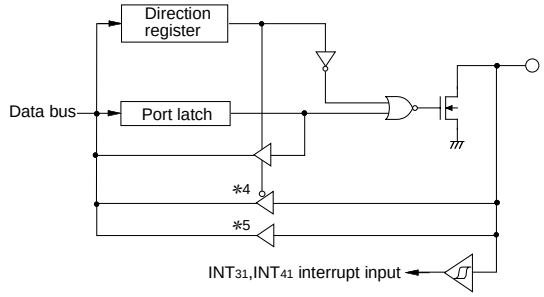


*4. The input level can be switched between CMOS compatible input level and TTL level by the P7 input level selection bit of the port control register 2 (address 002F16).

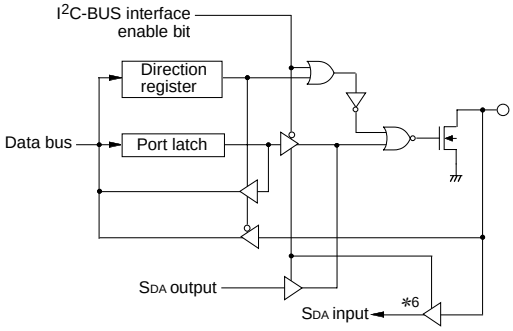
*5. The input level can be switched between CMOS compatible input level and TTL level by the P7 input level selection bit of the port control register 2 (address 002F16).
The port P8 direction register and port P7 input register can be switched by the P8 function selection bit of the port control register 2 (address 002F16).

Fig. 14 Port block diagram (3)

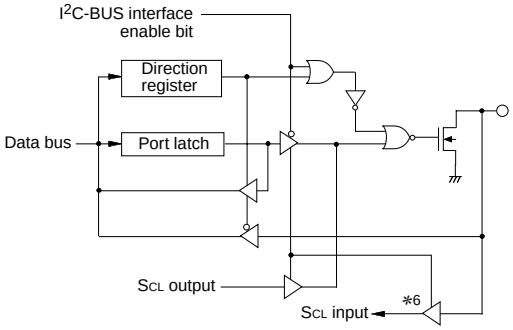
(25) Ports $P7_4$,$P7_5$



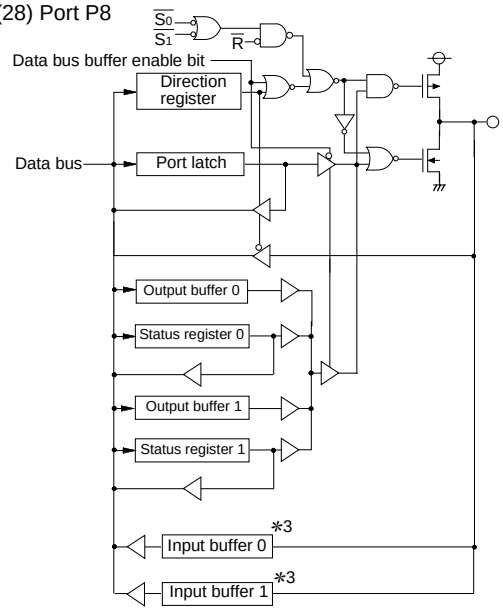
(26) Port $P7_6$



(27) Port $P7_7$



(28) Port P8



*6. The input level can be switched between CMOS compatible input level and SMBUS level by the I^2C-BUS interface pin input selection bit of the I^2C control register (address 001516).

Fig. 15 Port block diagram (4)

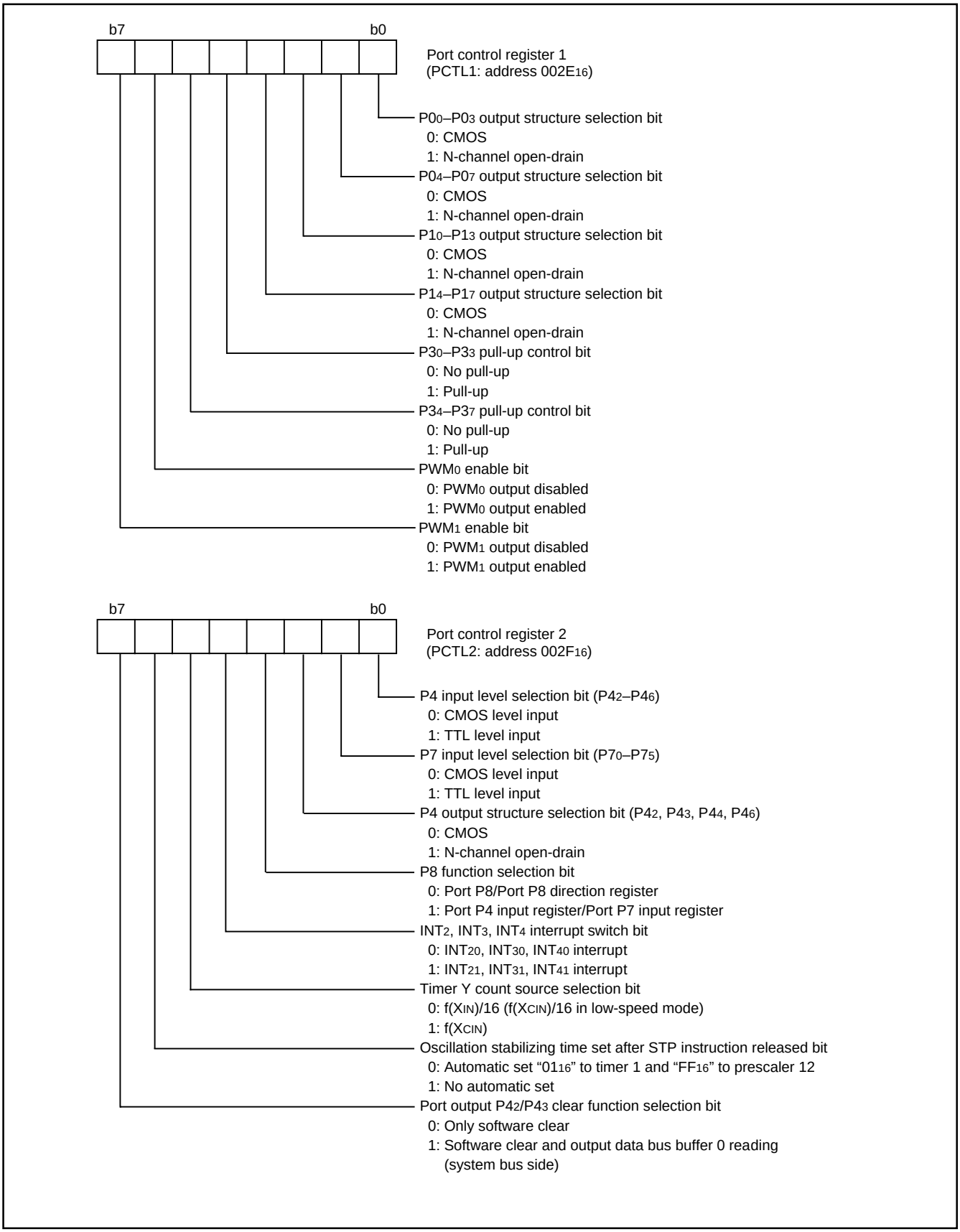


Fig. 16 Structure of port I/O related register

INTERRUPTS

Interrupts occur by 16 sources among 21 sources: nine external, eleven internal, and one software.

Interrupt Control

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

Interrupt enable bits can be set or cleared by software.

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

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

When several interrupts occur at the same time, the interrupts are received according to priority.

Interrupt Operation

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

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

Interrupt Source Selection

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

1. INT_0 or Input buffer full
2. INT_1 or Output buffer empty
3. Serial I/O1 transmission or SCL_{SDA}
4. $CNTR_0$ or SCL_{SDA}
5. Serial I/O2 or I^2C
6. INT_2 or I^2C
7. $CNTR_1$ or Key-on wake-up
8. A-D conversion or Key-on wake-up

External Interrupt Pin Selection

The occurrence sources of the external interrupt INT_2, INT_3, and INT_4 can be selected from either input from INT_{20}, INT_{30}, INT_{40} pin, or input from INT_{21}, INT_{31}, INT_{41} pin by the INT_2, INT_3, INT_4 interrupt switch bit (bit 4 of address $002F_{16}$).

■ Notes

When setting of the following register or bit is changed, the interrupt request bit may be set to "1."

- Interrupt edge selection register (address $003A_{16}$)
- Interrupt source selection register (address 0039_{16})
- INT_2, INT_3, INT_4 interrupt switch bit of Port control register 2 (bit 4 of address $002F_{16}$)

Accept the interrupt after clearing the interrupt request bit to "0" after interrupt is disabled and the above register or bit is set.

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)
Input buffer full (IBF)				At input data bus buffer writing	
INT1	3	FFF916	FFF816	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
Output buffer empty (OBE)				At output data bus buffer reading	
Serial I/O1 reception	4	FFF716	FFF616	At completion of serial I/O1 data reception	Valid when serial I/O1 is selected
Serial I/O1 transmission	5	FFF516	FFF416	At completion of serial I/O1 transfer shift or when transmission buffer is empty	Valid when serial I/O1 is selected
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
Timer X	6	FFF316	FFF216	At timer X underflow	
Timer Y	7	FFF116	FFF016	At timer Y underflow	
Timer 1	8	FFEF16	FFEE16	At timer 1 underflow	STP release timer underflow
Timer 2	9	FFED16	FFEC16	At timer 2 underflow	
CNTR0	10	FFEB16	FFEA16	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
CNTR1	11	FFE916	FFE816	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
Key-on wake-up				At falling of port P3 (at input) input logical level AND	External interrupt (falling edge valid)
Serial I/O2	12	FFE716	FFE616	At completion of serial I/O2 data transfer	Valid when serial I/O2 is selected
I^2C				At completion of data transfer	
INT2	13	FFE516	FFE416	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
I^2C				At completion of data transfer	
INT3	14	FFE316	FFE216	At detection of either rising or falling edge of INT3 input	External interrupt (active edge selectable)
INT4	15	FFE116	FFE016	At detection of either rising or falling edge of INT4 input	External interrupt (active edge selectable)
A-D converter	16	FFDF16	FFDE16	At completion of A-D conversion	
Key-on wake-up				At falling of port P3 (at input) input logical level AND	External interrupt (falling edge valid)
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.

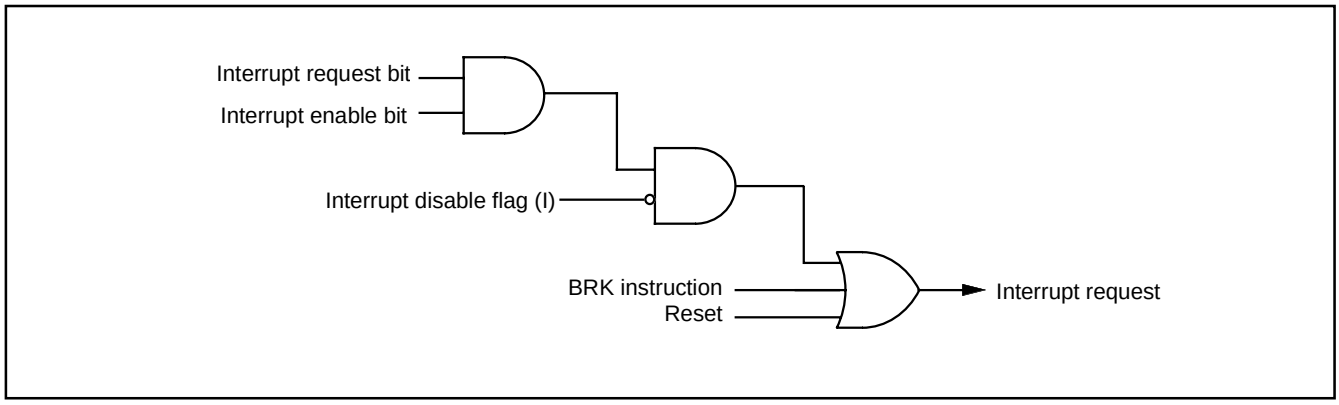


Fig. 17 Interrupt control

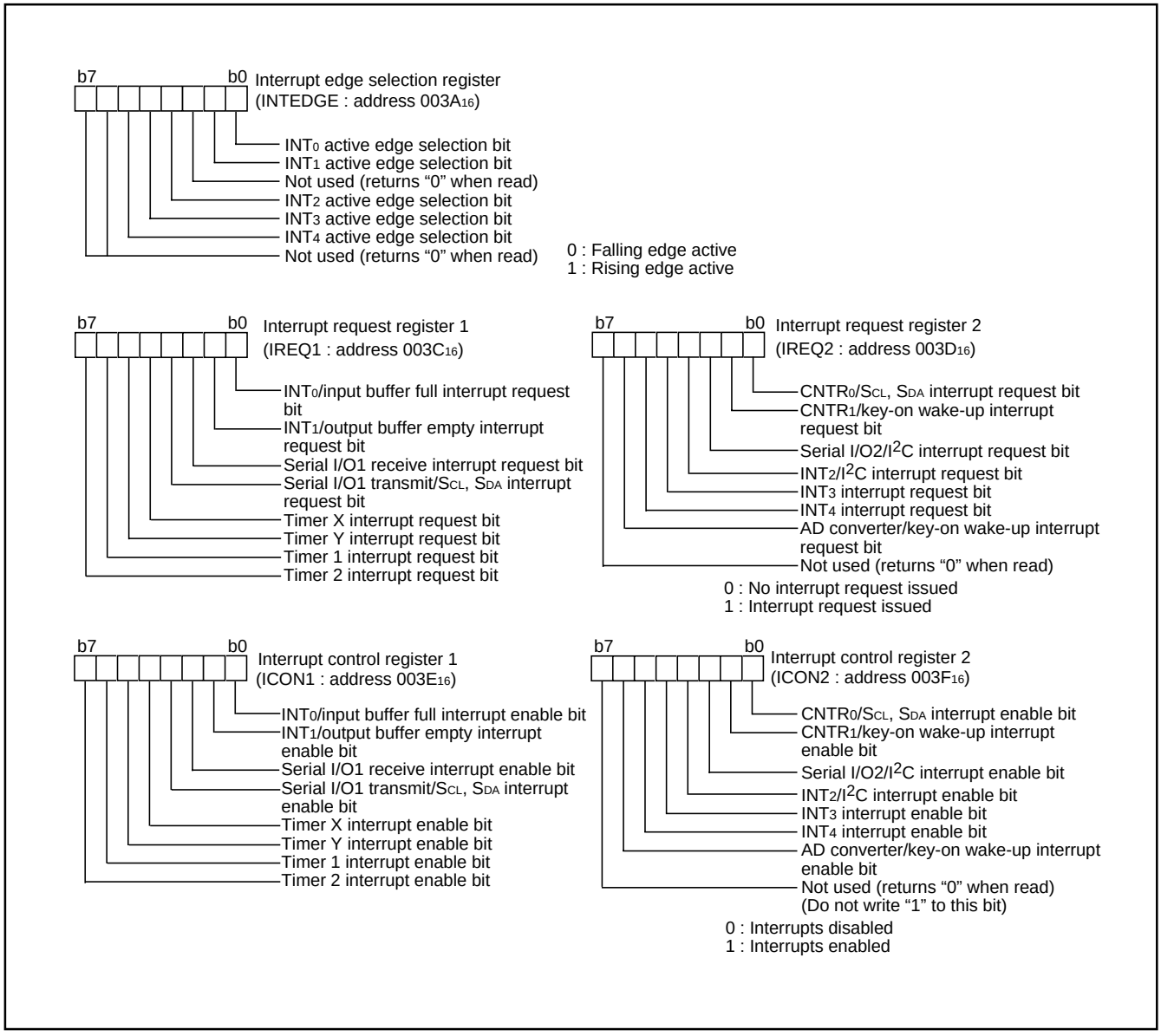


Fig. 18 Structure of interrupt-related registers (1)

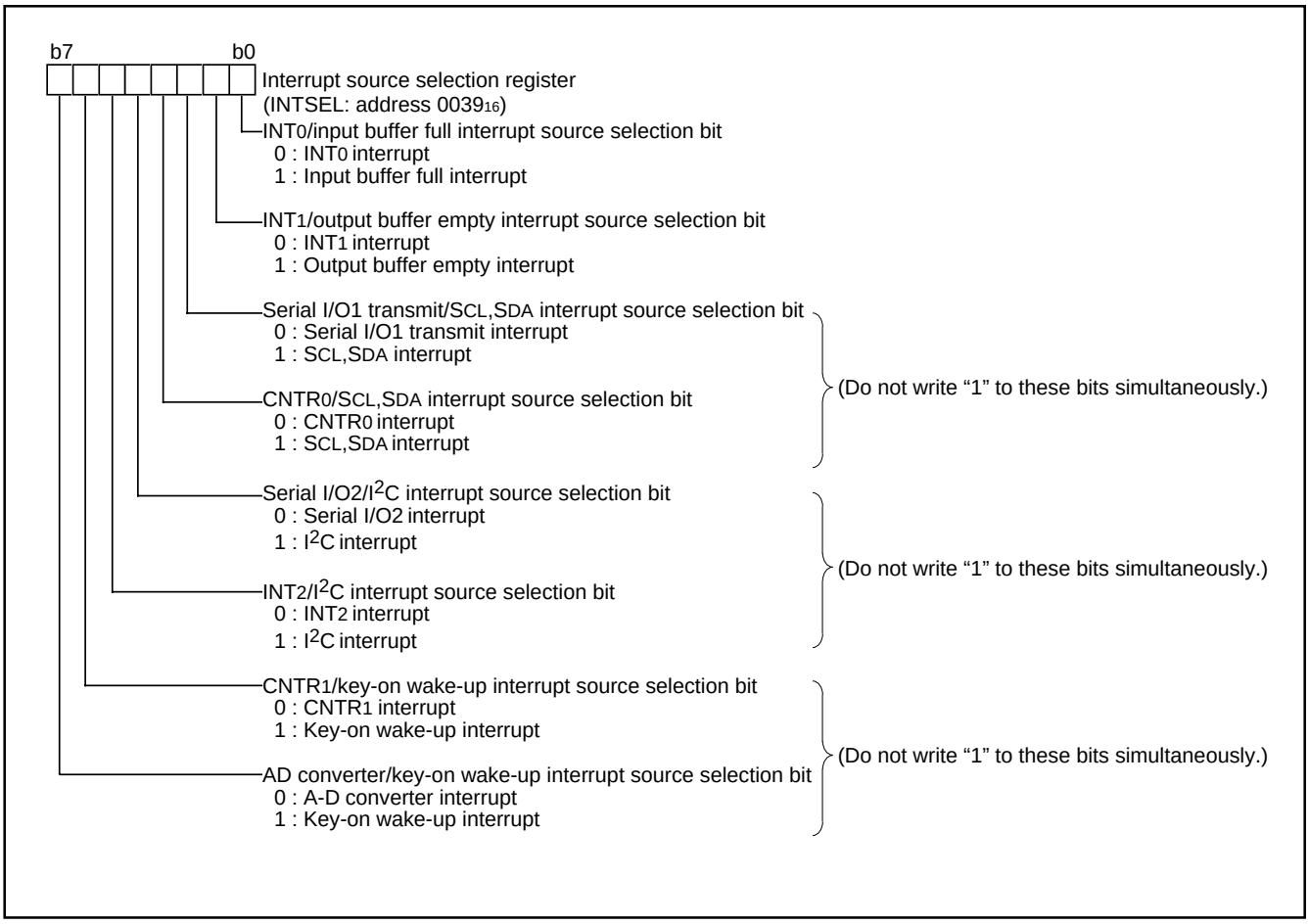


Fig. 19 Structure of interrupt-related registers (2)

Key Input Interrupt (Key-on Wake Up)

A Key input interrupt request is generated by applying "L" level to any pin of port P3 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 20, where an interrupt request is generated by pressing one of the keys consisted as an active-low key matrix which inputs to ports $P3_0$–$P3_3$.

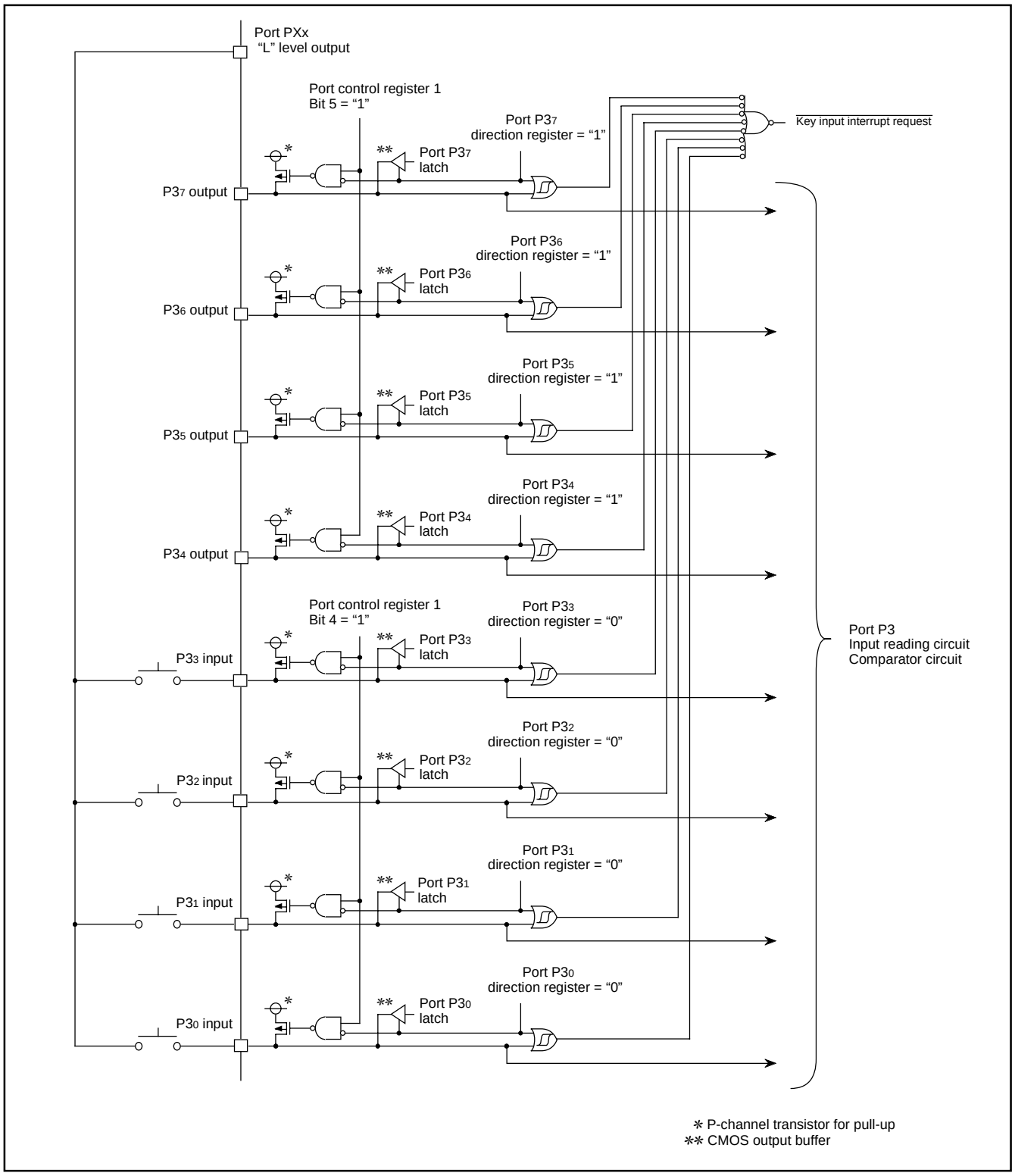


Fig. 20 Connection example when using key input interrupt and port P3 block diagram

TIMERS

The 3886 group has four timers: timer X, timer Y, timer 1, and timer 2.

The division ratio of each timer or prescaler is given by 1/(n + 1), where n is the value in the corresponding timer or prescaler latch.

All timers are count down. When the timer reaches "00_{16}", an underflow occurs at the next count pulse and the corresponding timer latch is reloaded into the timer and the count is continued. When a timer underflows, the interrupt request bit corresponding to that timer is set to "1".

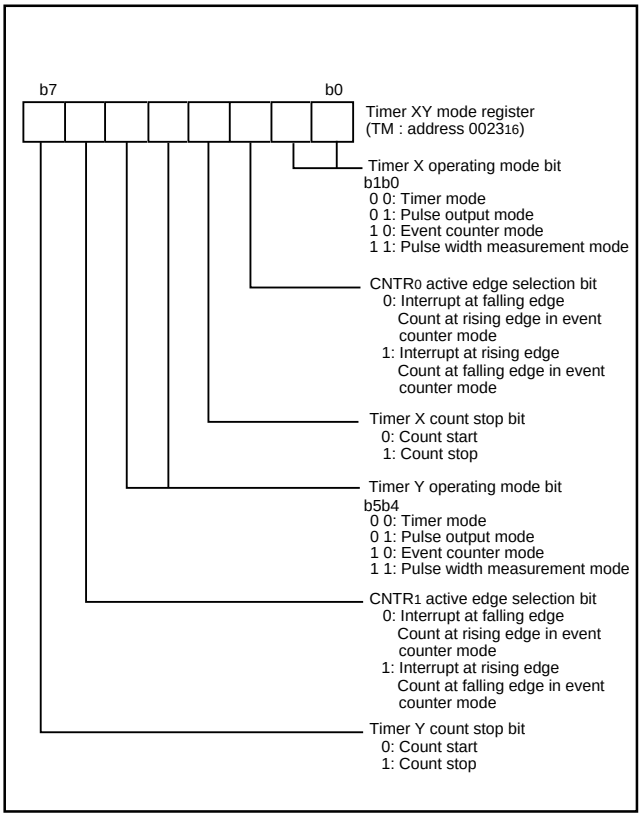


Fig. 21 Structure of timer XY mode register

Timer 1 and Timer 2

The count source of prescaler 12 is the oscillation frequency divided by 16. The output of prescaler 12 is counted by timer 1 and timer 2, and a timer underflow sets the interrupt request bit.

Timer X and Timer Y

Timer X and Timer Y can each select in one of four operating modes by setting the timer XY mode register.

(1) Timer Mode

The timer counts f(X_{IN})/16.

(2) Pulse Output Mode

Timer X (or timer Y) counts f(X_{IN})/16. Whenever the contents of the timer reach "00_{16}", the signal output from the $CNTR_0$ (or $CNTR_1$) pin is inverted. If the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "0", output begins at " H".

If it is "1", output starts at "L". When using a timer in this mode, set the corresponding port $P5_4$ (or port $P5_5$) direction register to output mode.

(3) Event Counter Mode

Operation in event counter mode is the same as in timer mode, except that the timer counts signals input through the $CNTR_0$ or $CNTR_1$ pin.

When the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "0", the rising edge of the $CNTR_0$ (or $CNTR_1$) pin is counted.

When the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "1", the falling edge of the $CNTR_0$ (or $CNTR_1$) pin is counted.

(4) Pulse Width Measurement Mode

If the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "0", the timer counts f(X_{IN})/16 while the $CNTR_0$ (or $CNTR_1$) pin is at "H". If the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "1", the timer counts while the $CNTR_0$ (or $CNTR_1$) pin is at "L".

The count can be stopped by setting "1" to the timer X (or timer Y) count stop bit in any mode. The corresponding interrupt request bit is set each time a timer overflows.

The count source for timer Y in the timer mode or the pulse output mode can be selected from either f(X_{IN})/16 or f(X_{CIN}) by the timer Y count source selection bit of the port control register 2 (bit 5 of address $002F_{16}$).

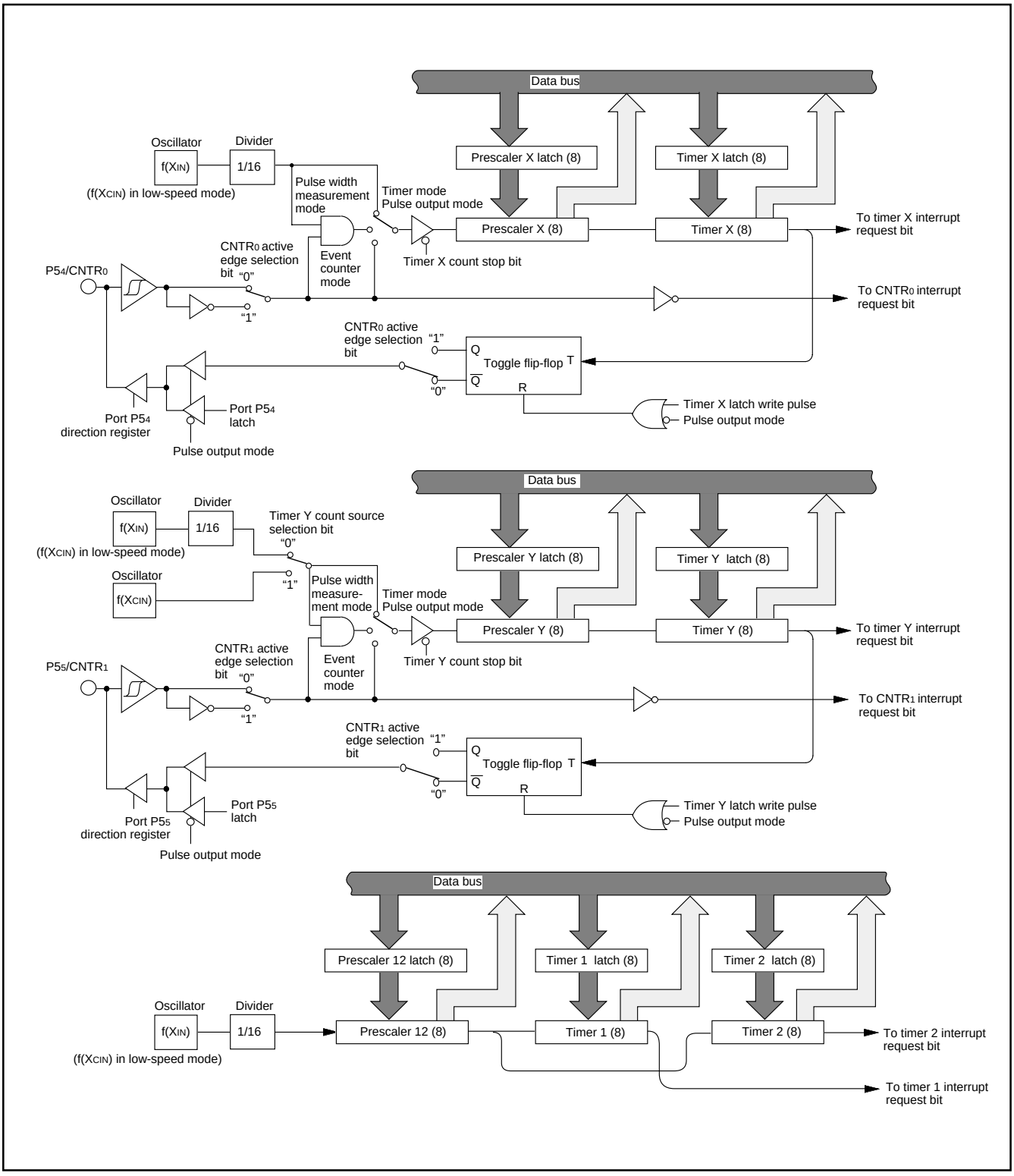


Fig. 22 Block diagram of timer X, timer Y, timer 1, and timer 2

SERIAL I/O
Serial I/O1

Serial I/O1 can be used as either clock synchronous or asynchronous (UART) serial I/O. A dedicated timer is also provided for baud rate generation.

(1) Clock Synchronous Serial I/O Mode

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

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

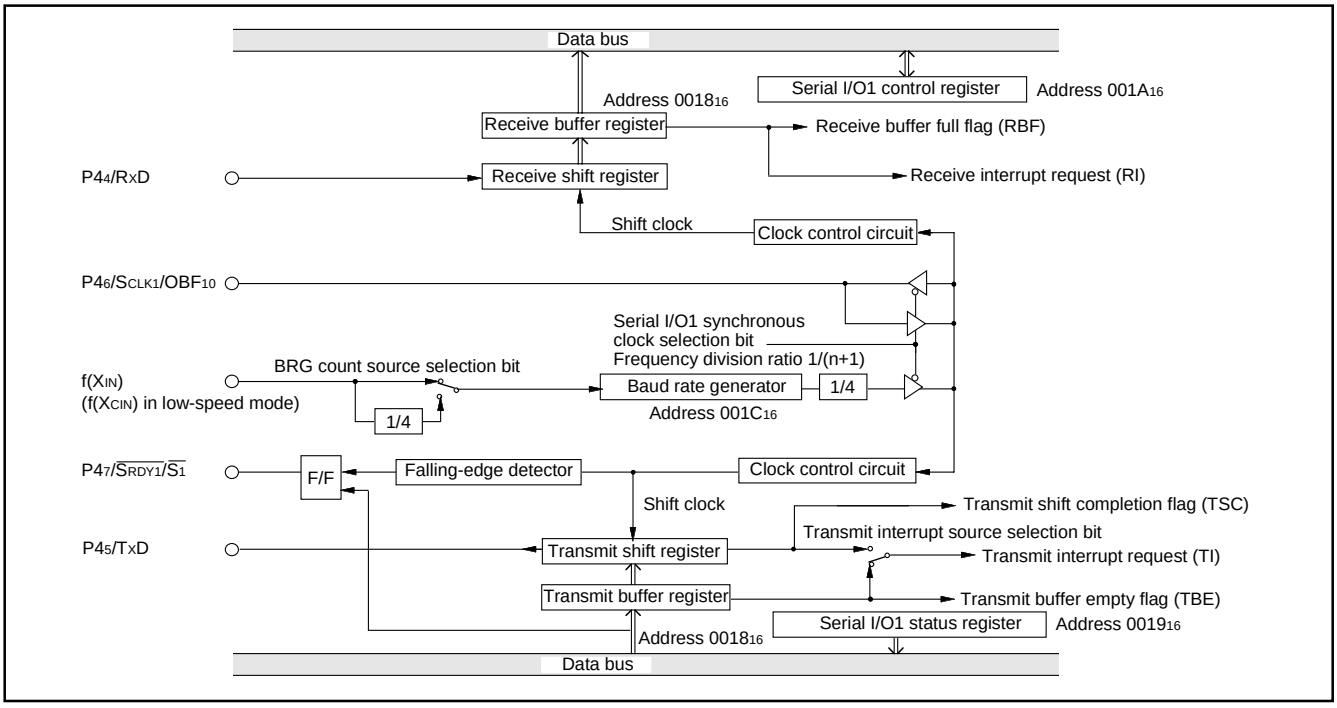
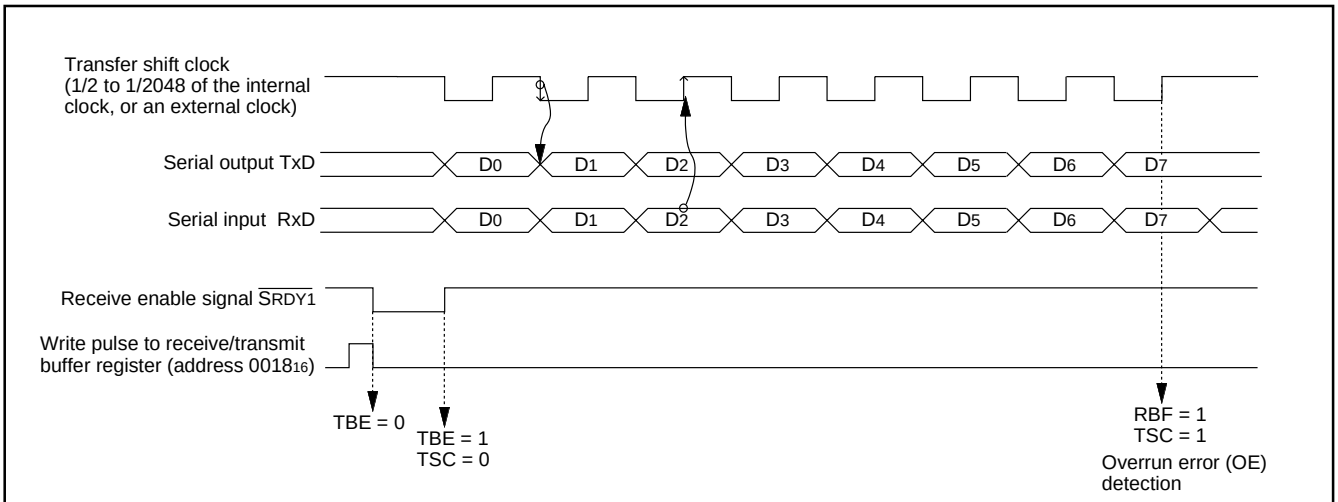


Fig. 23 Block diagram of clock synchronous serial I/O1



Notes 1: As the transmit interrupt (TI), which can be selected, either when the transmit buffer has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.

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

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

Fig. 24 Operation of clock synchronous serial I/O1 function

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

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

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

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

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

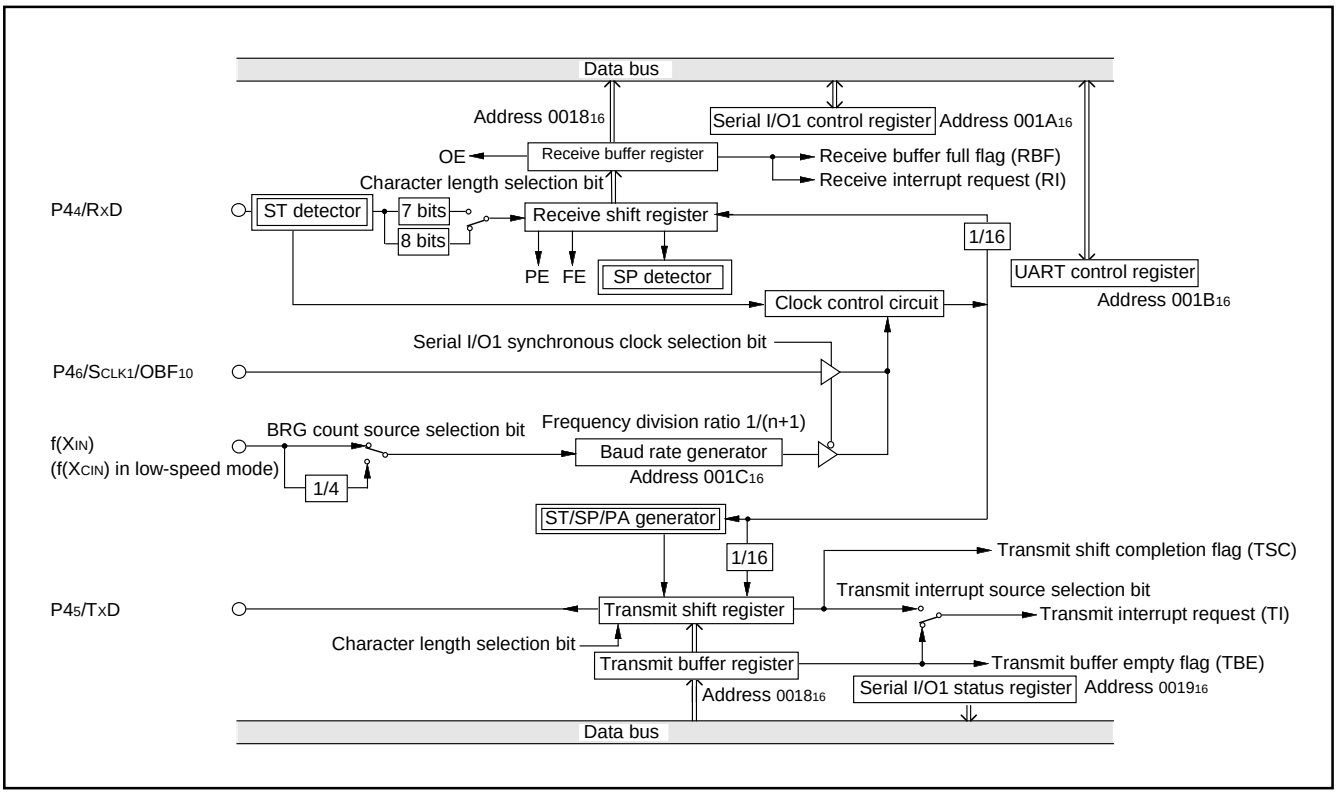
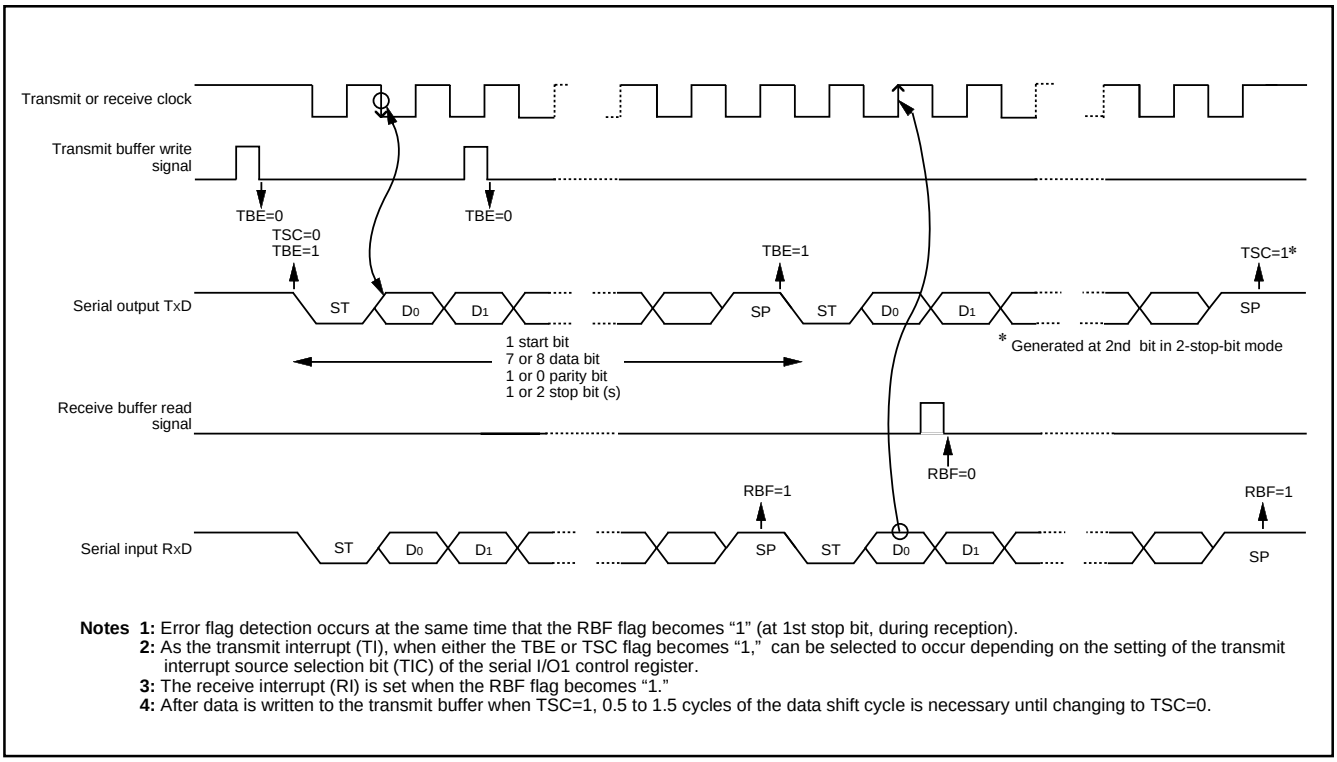


Fig. 25 Block diagram of UART serial I/O1



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

Fig. 26 Operation of UART serial I/O1 function

[Serial I/O1 Control Register (SIO1CON)] 001A16

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

[UART Control Register (UARTCON)] 001B16

The UART control register consists of four control bits (bits 0 to 3) which are valid when asynchronous serial I/O is selected and set the data format of an data transfer and one bit (bit 4) which is always valid and sets the output structure of the P45/TxD pin.

[Serial I/O1 Status Register (SIO1STS)] 001916

The read-only serial I/O1 status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O function and various errors.

Three of the flags (bits 4 to 6) are valid only in UART mode.

The receive buffer full flag (bit 1) is cleared to "0" when the receive buffer register is read.

If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/O1 status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing "0" to the serial I/O1 enable bit SIOE (bit 7 of the serial I/O control register) also clears all the status flags, including the error flags.

Bits 0 to 6 of the serial I/O1 status register are initialized to "0" at reset, but if the transmit enable bit (bit 4) of the serial I/O1 control register has been set to "1", the transmit shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become "1".

[Transmit Buffer Register/Receive Buffer Register (TB/RB)] 001816

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

[Baud Rate Generator (BRG)] 001C16

The baud rate generator determines the baud rate for serial transfer.

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

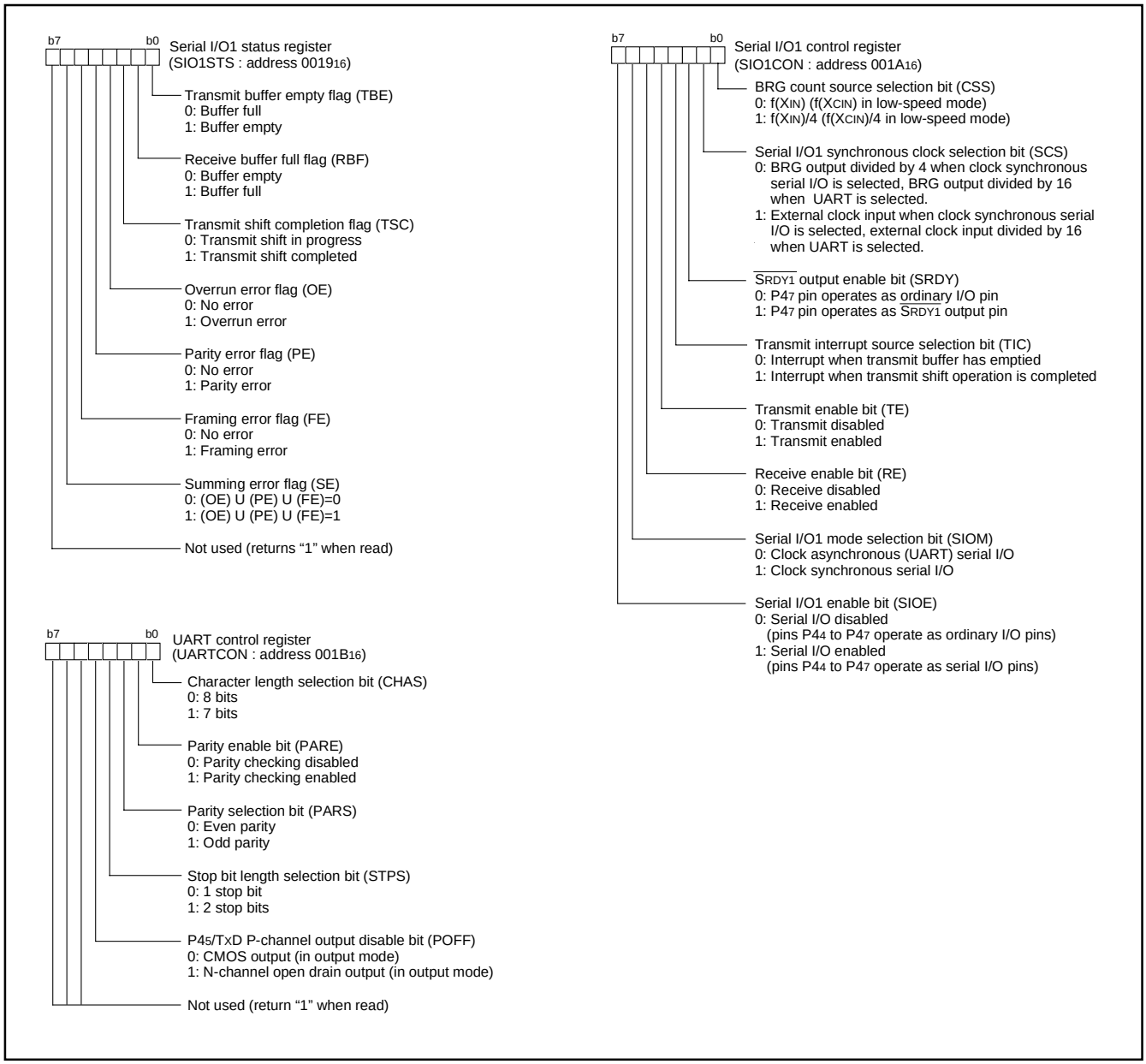


Fig. 27 Structure of serial I/O1 control registers

Serial I/O2

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

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

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

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

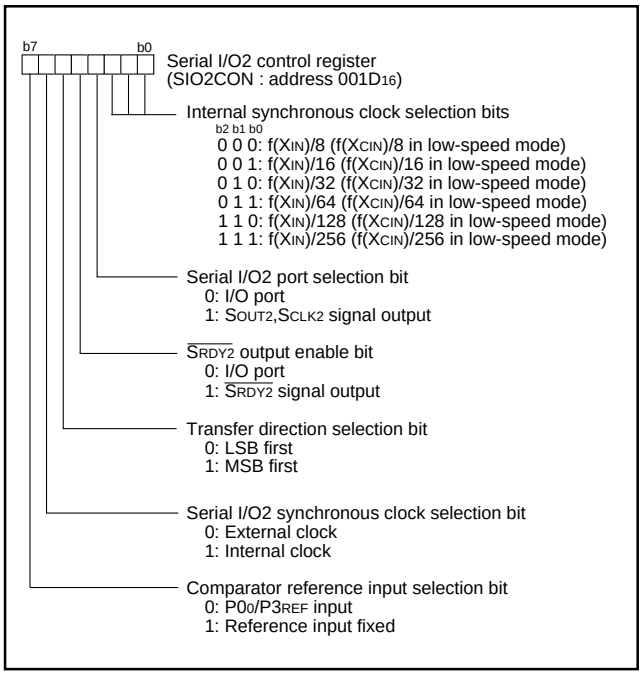


Fig. 28 Structure of serial I/O2 control register

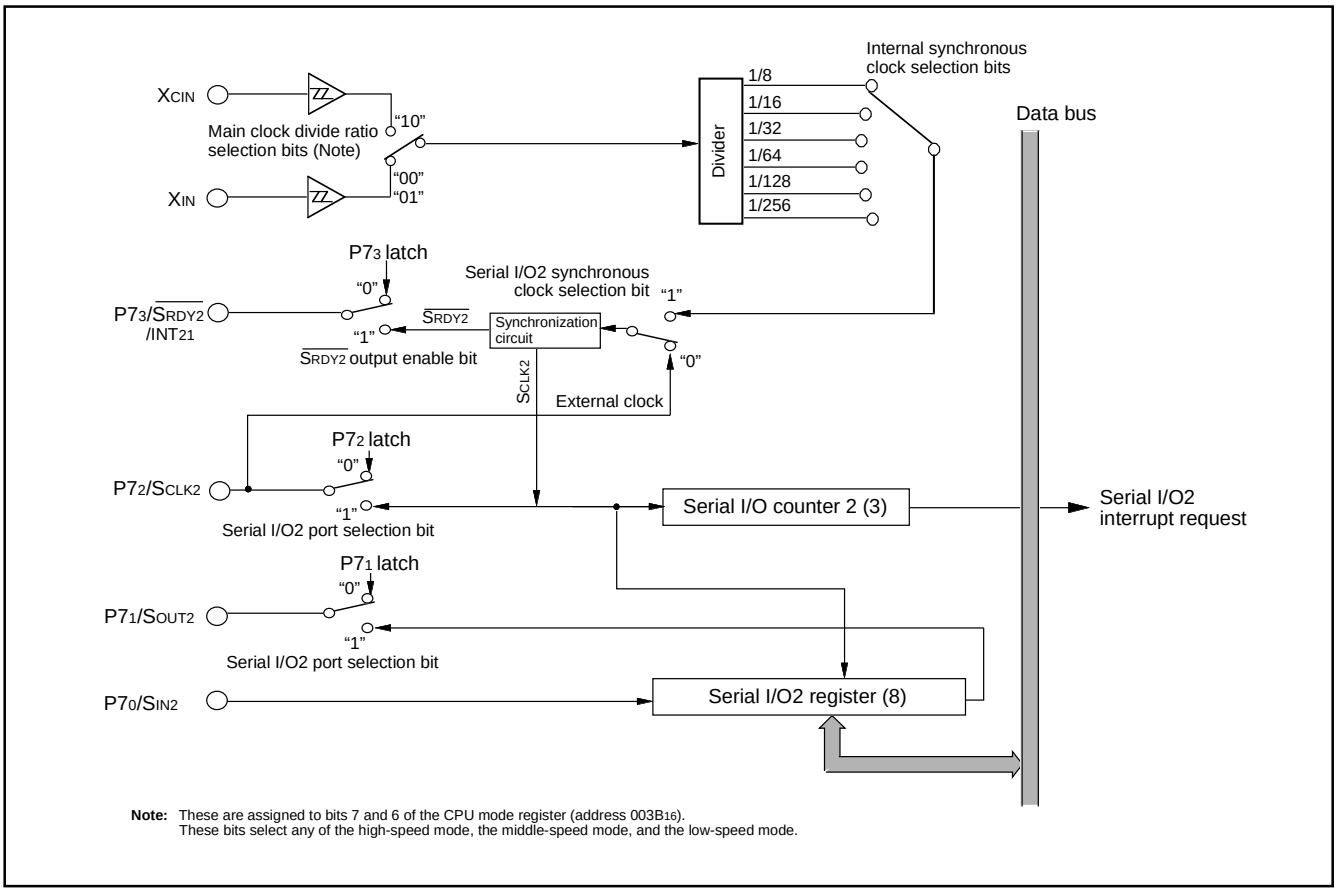
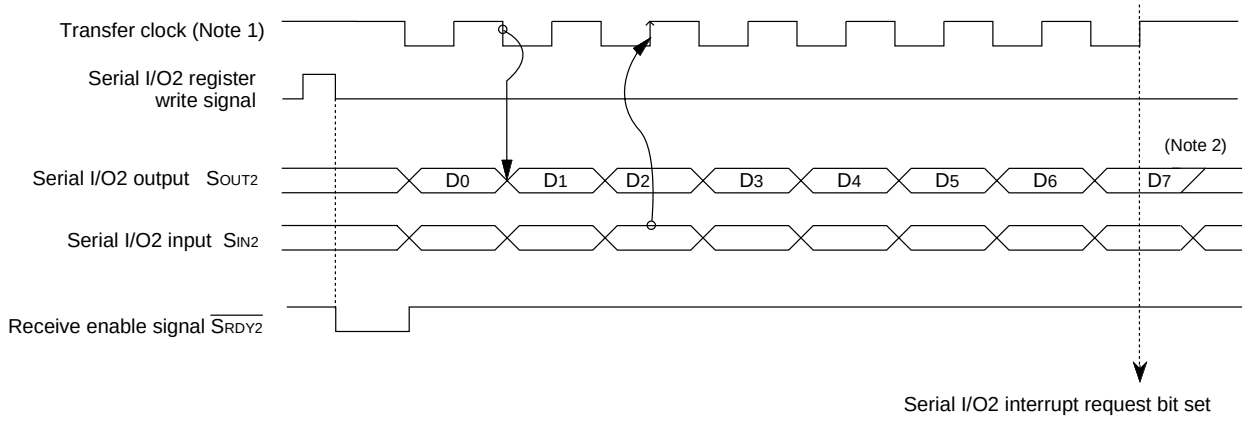


Fig. 29 Block diagram of serial I/O2 function



Notes 1: When the internal clock is selected as the transfer clock, the divide ratio can be selected by setting bits 0 to 2 of the serial I/O2 control register.
2: When the internal clock is selected as the transfer clock, the SOUT2 pin goes to high impedance after transfer completion.

Fig. 30 Timing of serial I/O2 function

PULSE WIDTH MODULATION (PWM) OUTPUT CIRCUIT

The 3886 group has two PWM output circuits, PWM0 and PWM1, with 14-bit resolution respectively. These can operate independently. When the oscillation frequency XIN is 10 MHz, the minimum resolution bit width is 200 ns and the cycle period is 3276.8 µs. The PWM timing generator supplies a PWM control signal based on a signal that is the frequency of the XIN clock.
The following explanation assumes f(XIN) = 8 MHz.

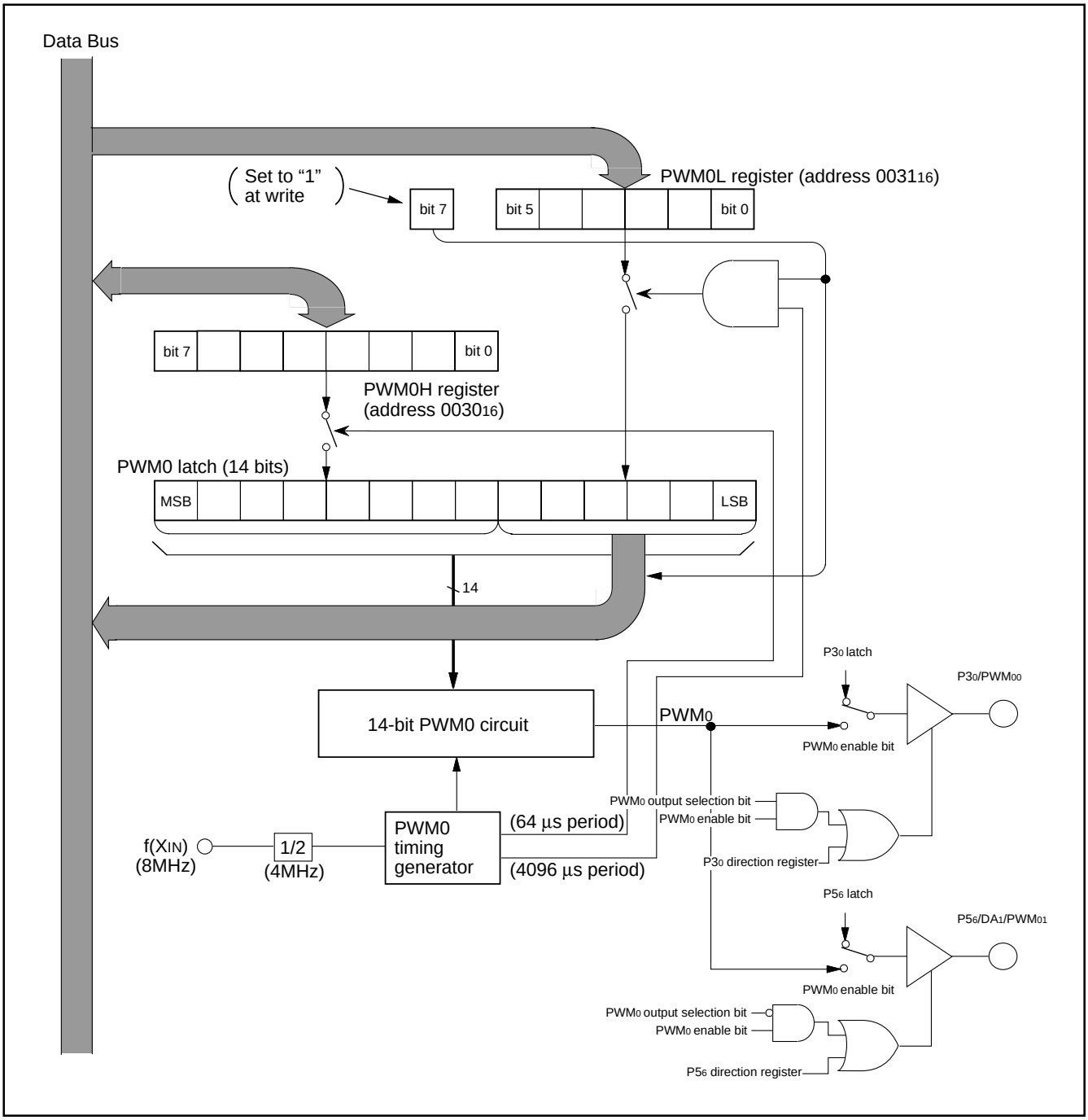


Fig. 31 PWM block diagram (PWM0)

Data Setup (PWM0)

The PWM0 output pin also functions as port $P3_0$ or $P5_6$. The PWM0 output pin is selected from either $P3_0/PWM_{00}$ or $P5_6/PWM_{01}$ by bit 4 of the AD/DA control register (address 0034_{16}).

The PWM0 output becomes enabled state by setting bit 6 of the port control register 1 (address $002E_{16}$). The high-order eight bits of output data are set in the PWM0H register (address 0030_{16}) and the low-order six bits are set in the PWM0L register (address 0031_{16}).

PWM1 is set as the same way.

PWM Operation

The 14-bit PWM data is divided into the low-order six bits and the high-order eight bits in the PWM latch.

The high-order eight bits of data determine how long an "H"-level signal is output during each sub-period. There are 64 sub-periods in each period, and each sub-period is 256 × τ (64 μs) long. The signal is "H" for a length equal to N times τ, where τ is the minimum resolution (250 ns).

"H" or "L" of the bit in the ADD part shown in Figure 33 is added to this "H" duration by the contents of the low-order 6-bit data according to the rule in Table 9.

That is, only in the sub-period tm shown by Table 9 in the PWM cycle period T = 64t, its "H" duration is lengthened to the minimum resolution τ added to the length of other periods.

For example, if the high-order eight bits of the 14-bit data are 03_{16} and the low-order six bits are 05_{16}, the length of the "H"-level output in sub-periods t_8, t_{24}, t_{32}, t_{40}, and t_{56} is 4 τ, and its length is 3 τ in all other sub-periods.

Time at the "H" level of each sub-period almost becomes equal, because the time becomes length set in the high-order 8 bits or becomes the value plus τ, and this sub-period t (= 64 μs, approximate 15.6 kHz) becomes cycle period approximately.

Table 9 Relationship between low-order 6 bits of data and period set by the ADD bit

Low-order 6 bits of data (PWML) (LSB)	Sub-periods tm Lengthened (m=0 to 63)
0 0 0 0 0 0	None
0 0 0 0 0 1	m=32
0 0 0 0 1 0	m=16, 48
0 0 0 1 0 0	m=8, 24, 40, 56
0 0 1 0 0 0	m=4, 12, 20, 28, 36, 44, 52, 60
0 1 0 0 0 0	m=2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62
1 0 0 0 0 0	m=1, 3, 5, 7, ... ,57, 59, 61, 63

Transfer From Register to Latch

Data written to the PWML register is transferred to the PWM latch at each PWM period (every 4096 μs), and data written to the PWMH register is transferred to the PWM latch at each sub-period (every 64 μs). The signal which is output to the PWM output pin is corresponding to the contents of this latch. When the PWML register is read, the latch contents are read. However, bit 7 of the PWML register indicates whether the transfer to the PWM latch is completed; the transfer is completed when bit 7 is "0" and it is not done when bit 7 is "1."

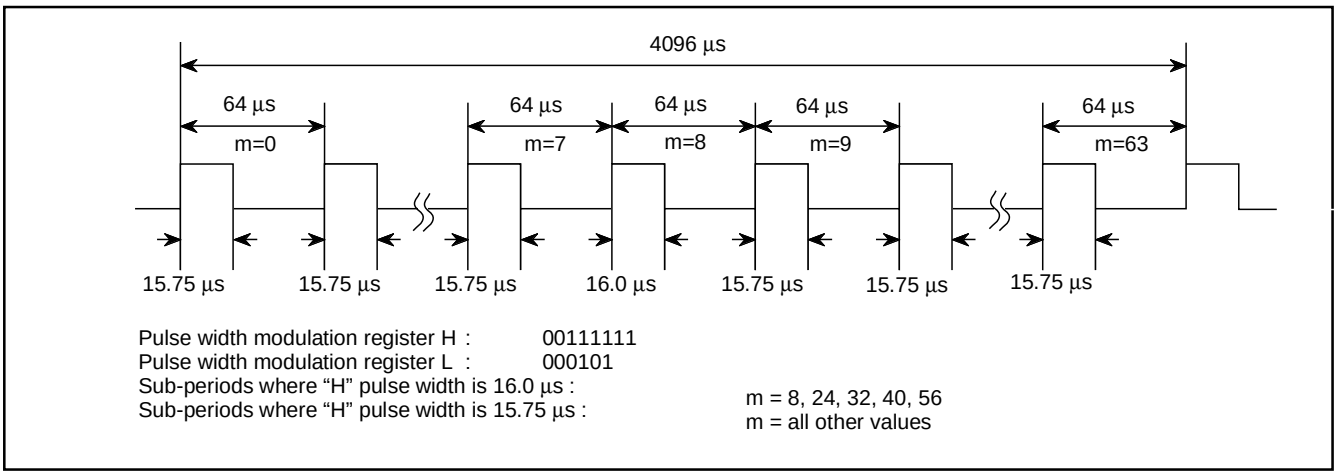


Fig. 32 PWM timing

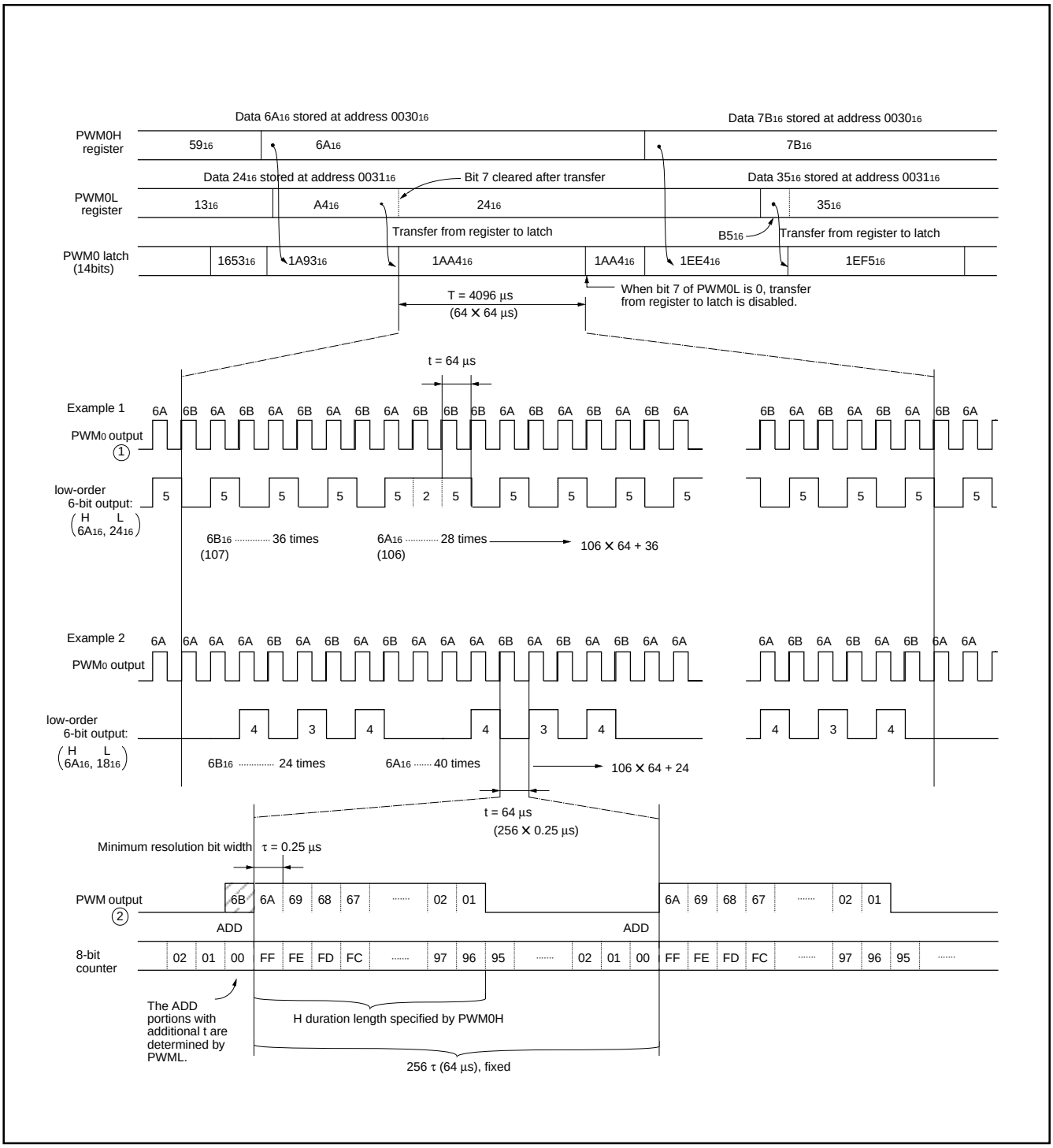


Fig. 33 14-bit PWM timing (PWM0)

BUS INTERFACE

The 3886 group has a 2-byte bus interface function which is almost functionally equal to MELPS8-41 series and the control signal from the host CPU side can operate it (slave mode).

It is possible to connect the 3886 group with the $\overline{RD}$ and $\overline{WR}$ separated CPU bus directly. Figure 36 shows the block diagram of the bus interface function.

The data bus buffer function I/O pins ($P4_2$, $P4_3$, $P4_6$, $P4_7$, $P5_0$–$P5_3$, P8) also function as the normal digital port I/O pins. When bit 0 (data bus buffer enable bit) of the data bus buffer control register (address $002A_{16}$) is "0," these pins become the normal digital port I/O pins. When it is "1," these bits become the data bus buffer function I/O pins.

The selection of either the single data bus buffer mode, which uses 1 byte: data bus buffer 0 only, or the double data bus buffer mode, which uses 2 bytes: data bus buffer 0 and data bus buffer 1, is performed by bit 1 (data bus buffer function selection bit) of the data bus buffer control register (address $002A_{16}$). Port $P4_7$ becomes $\overline{S_1}$ input in the double data bus buffer mode. When data is written from the host CPU side, an input buffer full interrupt occurs. When data is read from the host CPU, an output buffer empty interrupt occurs. This microcomputer shares two input buffer full interrupt requests and two output buffer empty interrupt requests as shown in Figure 34, respectively.

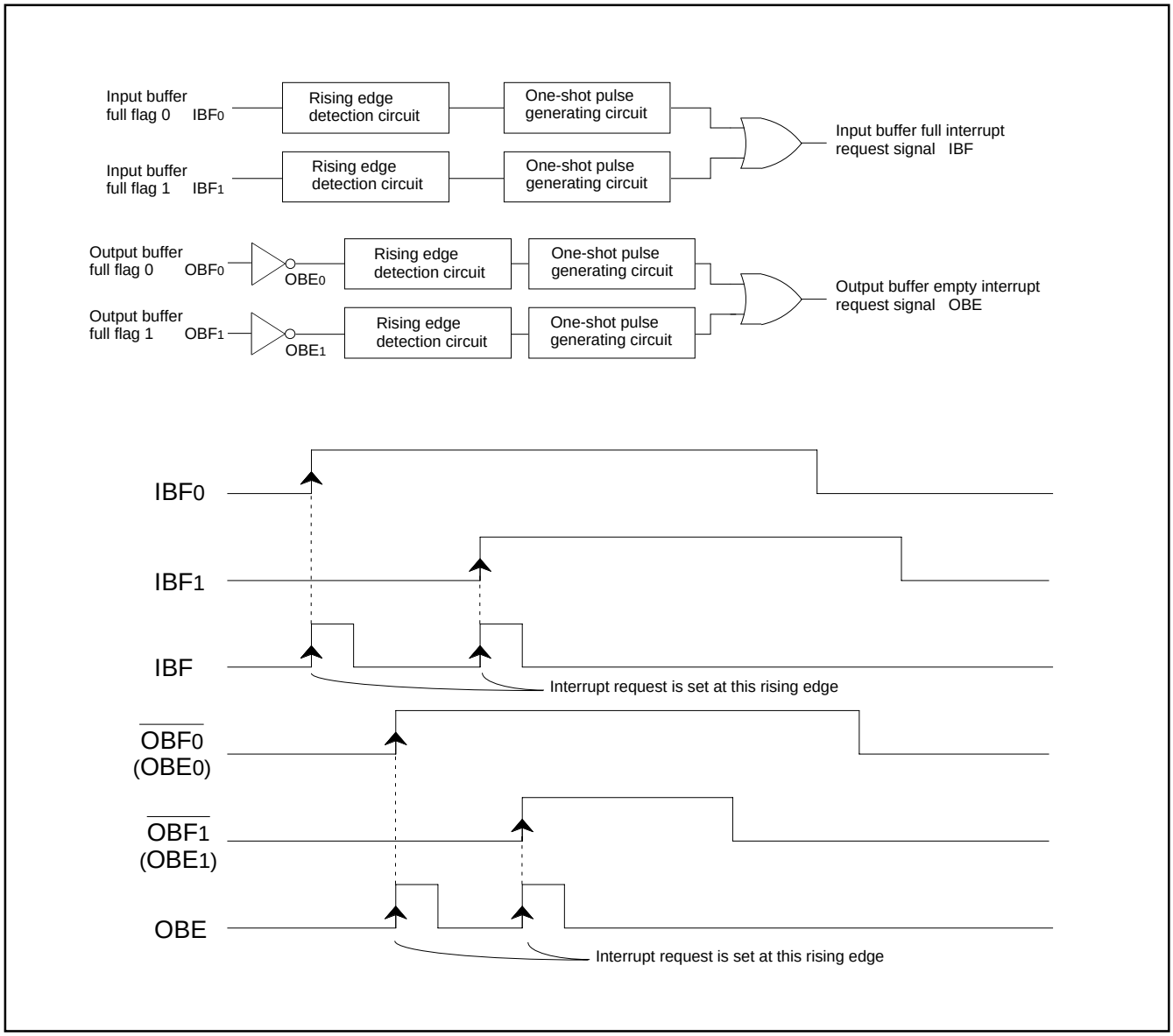
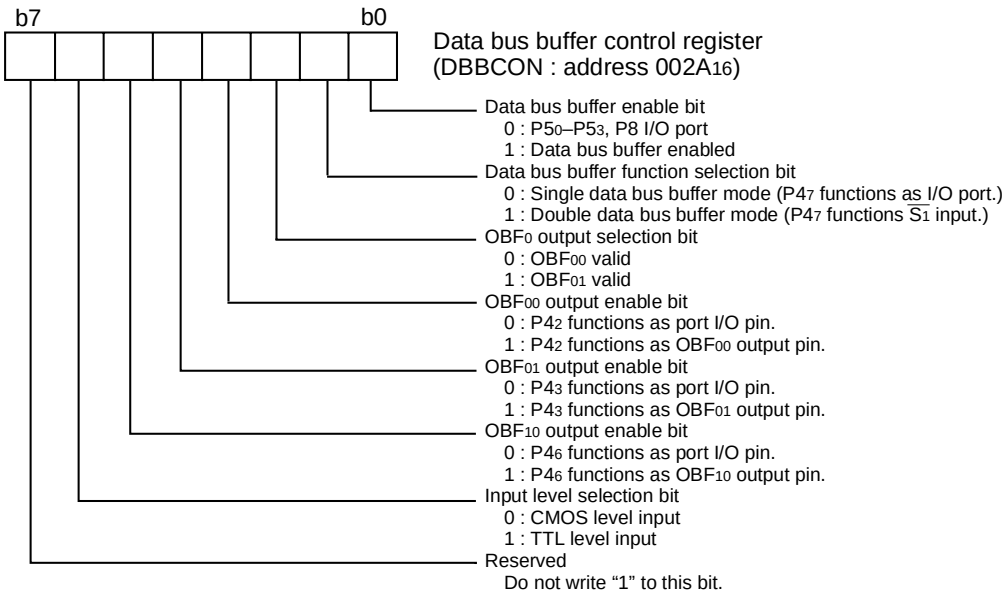
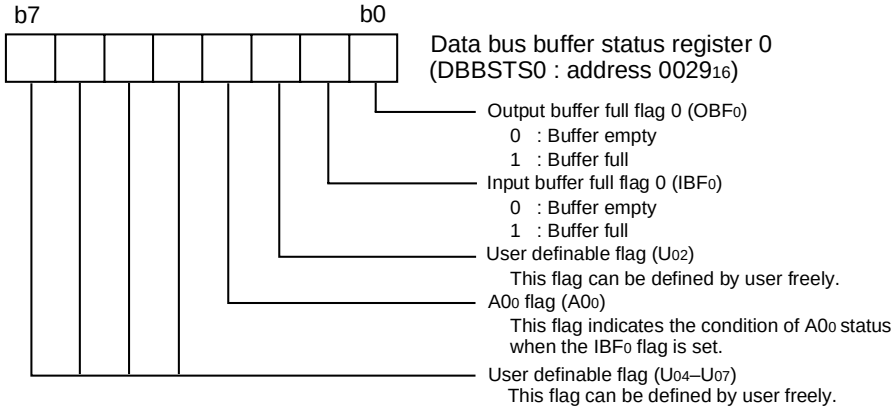


Fig. 34 Interrupt request circuit of data bus buffer





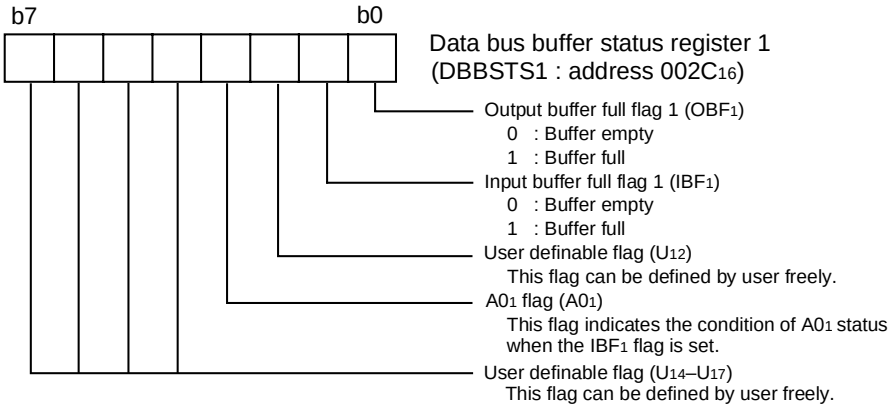


Fig. 35 Structure of bus interface related register

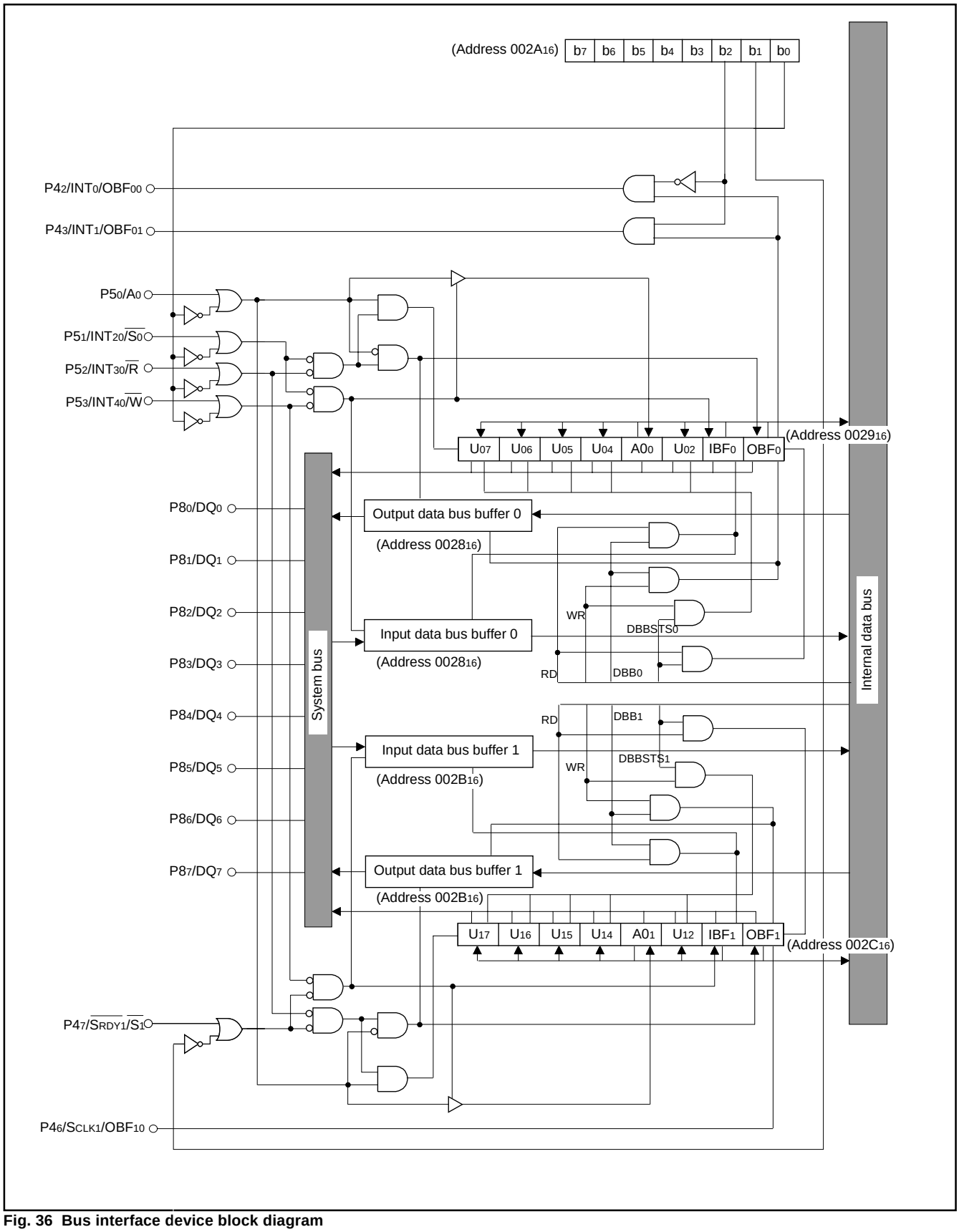


Fig. 36 Bus interface device block diagram

[Data Bus Buffer Status Register 0, 1 (DBBSTS0, DBBSTS1)] 0029_{16}, $002C_{16}$

The data bus buffer status registers 0 and 1 consist of eight bits. Bits 0, 1, and 3 are read-only bits and indicate the condition of the data bus buffer. Bits 2, 4, 5, 6, and 7 are user definable flags which can be set by program, and can be read/written. This register can be read from the host CPU when the A_0 pin is set to "H" only.

•Bit 0: Output buffer full flag OBF_0, OBF_1

When writing data to the output data bus buffer, these flags are set to "1". When reading the output data bus buffer from the host CPU, these flags are cleared to "0".

•Bit 1: Input buffer full flag IBF_0, IBF_1

When writing data from the host CPU to the input data bus buffer, these flags are set to "1". When reading the input data bus buffer from the slave CPU side, these flags are cleared to "0".

•Bit 3: A_0 flag $A0_0$, $A0_1$

When writing data from the host CPU to the input data bus buffer, the level of the A_0 pin is latched.

[Input Data Bus Buffer Register 0, 1 (DBBIN0, DBBIN1)] 0028_{16}, $002B_{16}$

Data on the data bus is latched to DBBIN by writing request from the host CPU. Data of DBBIN can be read from the data bus buffer registers (address 0028_{16} or $002B_{16}$) on SFR.

[Output Data Bus Buffer Register 0, 1 (DBBOUT0, DBBOUT1)] 0028_{16}, $002B_{16}$

When writing data to the data bus buffer registers (address 0028_{16} or $002B_{16}$) on SFR, data is set to DBBOUT. Data of DBBOUT is output from the host CPU to the data bus by performing the reading request when the A_0 pin is set to "L".

[Port control Register 2 (PCTL2)] $002F_{16}$

Even if the data bus buffer function is enabled, both $P4_2$ and $P4_3$ function as ports when the OBF_{00} output enable bit (bit 3 of address $2A_{16}$) or the OBF_{01} output enable bit (bit 4 of address $2A_{16}$) is "0". Ports $P4_2$ and $P4_3$ are cleared to "0" by changing the input buffer full flag 0 (bit 1 of address 29_{16}) from "1" to "0" under the following conditions: the port output $P4_2$/$P4_3$ clear function selection bit (bit 7) is set to "1", both ports are in the output mode of the port function, and both port latches are "1".

Table 10 Function description of control I/O pins at bus interface function selected

Pin	Name	OBF_{00} output enable bit	OBF_{01} output enable bit	OBF_{10} output enable bit	Input /Output	Functions
$P4_7/\overline{SRDY_1}$ $/\overline{S_1}$	$\overline{S_1}$	–	–	–	Input	Chip select input This is used for selecting the data bus buffer 1 and is selected at "L" level.
$P5_0/A_0$	A_0	–	–	–	Input	Address input This is used for selecting DBBSTS and DBBOUT when the host CPU is read. This is used for distinguishing command from data when writing to the host CPU.
$P5_1/\overline{INT_{20}}$ $/\overline{S_0}$	$\overline{S_0}$	–	–	–	Input	Chip select input This is used for selecting the data bus buffer 0 and is selected at "L" level.
$P5_2/\overline{INT_{30}}$ $/\overline{R}$	$\overline{R}$	–	–	–	Input	This is a timing signal for reading data from the data bus buffer to the host CPU.
$P5_3/\overline{INT_{40}}$ $/\overline{W}$	$\overline{W}$	–	–	–	Input	This is a timing signal for writing data to the data bus buffer by the host CPU.
$P4_2/INT_0$ $/OBF_{00}$	OBF_{00}	1	0	0	Output	Status output signal OBF_{00} signal is output.
$P4_3/INT_1$ $/OBF_{01}$	OBF_{01}	0	1	0	Output	Status output signal OBF_{01} signal is output.
$P4_6/SCLK_1$ $/OBF_{10}$	OBF_{10}	0	0	1	Output	Status output signal OBF_{10} signal is output.

MULTI-MASTER I²C-BUS INTERFACE

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

Figure 37 shows a block diagram of the multi-master I²C-BUS interface and Table 11 lists the multi-master I²C-BUS interface functions.

This multi-master I²C-BUS interface consists of the I²C address register, the I²C data shift register, the I²C clock control register, the I²C control register, the I²C status register, the I²C start/stop condition control register and other control circuits.

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

Table 11 Multi-master I²C-BUS interface functions

Item	Function
Format	In conformity with Philips I²C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	In conformity with Philips I²C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequency	16.1 kHz to 400 kHz (at ϕ= 4 MHz) 20.2 kHz to 312.5 kHz (at ϕ = 5 MHz)

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

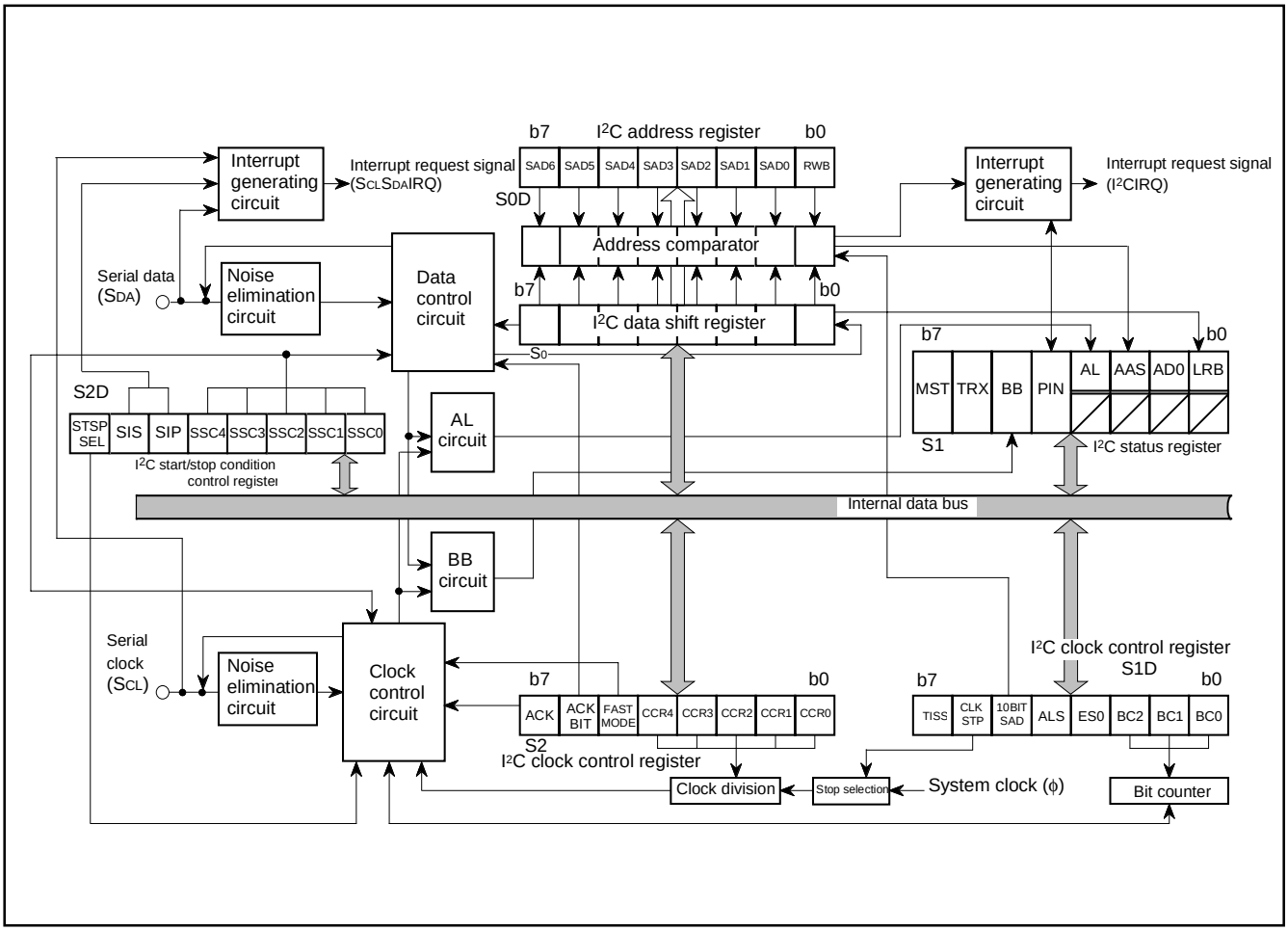


Fig. 37 Block diagram of multi-master I²C-BUS interface

✻ : Purchase of MITSUBISHI ELECTRIC CORPORATIONS I²C components conveys a license under the Philips I²C Patent Rights to use these components an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

[I²C Data Shift Register (S0)] 0012₁₆

The I²C data shift register (S0 : address 0012₁₆) is an 8-bit shift register to store receive data and write transmit data.

When transmit data is written into this register, it is transferred to the outside from bit 7 in synchronization with the SCL clock, and each time one-bit data is output, the data of this register are shifted by one bit to the left. When data is received, it is input to this register from bit 0 in synchronization with the SCL clock, and each time one-bit data is input, the data of this register are shifted by one bit to the left. The minimum 2 cycles of ϕ are required from the rising of the SCL clock until input to this register.

The I²C data shift register is in a write enable status only when the I²C-BUS interface enable bit (ES0 bit : bit 3 of address 15₁₆) of the I²C control register is "1." The bit counter is reset by a write instruction to the I²C data shift register. When both the ES0 bit and the MST bit of the I²C status register (address 0014₁₆) are "1," the SCL is output by a write instruction to the I²C data shift register. Reading data from the I²C data shift register is always enabled regardless of the ES0 bit value.

[I²C Address Register (S0D)] 0013₁₆

The I²C address register (address 0013₁₆) consists of a 7-bit slave address and a read/$\overline{\text{write}}$ bit. In the addressing mode, the slave address written in this register is compared with the address data to be received immediately after the START condition is detected.

•Bit 0: Read/$\overline{\text{write}}$ bit (RWB)

This is not used in the 7-bit addressing mode. In the 10-bit addressing mode, the first address data to be received is compared with the contents (SAD6 to SAD0 + RWB) of the I²C address register.

The RWB bit is cleared to "0" automatically when the stop condition is detected.

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

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

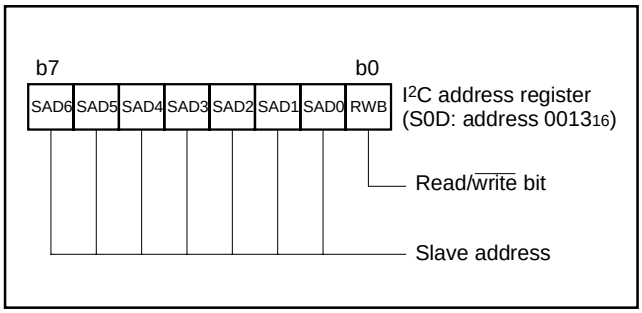


Fig. 38 Structure of I²C address register

[I²C Clock Control Register (S2)] 0016₁₆

The I²C clock control register (address 0016₁₆) is used to set ACK control, SCL mode and SCL frequency.

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

These bits control the SCL frequency. Refer to Table 12.

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

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

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

•Bit 6: ACK bit (ACK BIT)

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

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

*ACK clock: Clock for acknowledgment

•Bit 7: ACK clock bit (ACK)

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

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

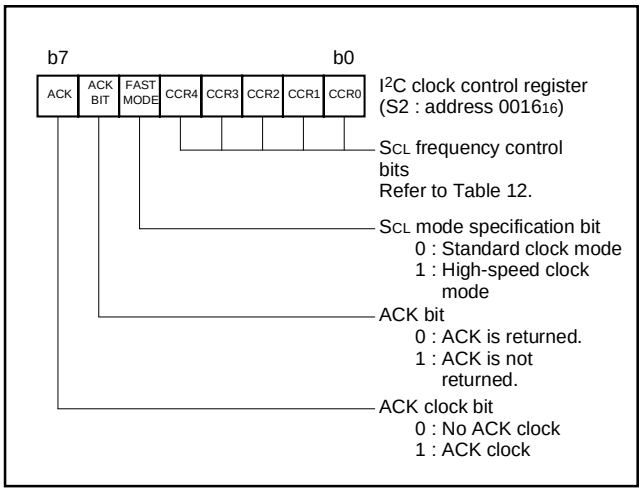


Fig. 39 Structure of I²C clock control register

Table 12 Set values of I²C clock control register and SCL frequency

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

Notes 1: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at ϕ = 4 MHz). "H" duration of the clock fluctuates from –4 to +2 cycles of ϕ in the standard clock mode, and fluctuates from –2 to +2 cycles of ϕ in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because "L" duration is extended instead of "H" duration reduction.

These are value when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2: Each value of SCL frequency exceeds the limit at ϕ = 4 MHz or more. When using these setting value, use ϕ of 4 MHz or less.

3: The data formula of SCL frequency is described below:
ϕ/(8 × CCR value) Standard clock mode
ϕ/(4 × CCR value) High-speed clock mode (CCR value ≠ 5)
ϕ/(2 × CCR value) High-speed clock mode (CCR value = 5)
Do not set 0 to 2 as CCR value regardless of ϕ frequency.
Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

[I²C Control Register (S1D)] 0015₁₆

The I²C control register (address 0015₁₆) controls data communication format.

•Bits 0 to 2: Bit counter (BC0–BC2)

These bits decide the number of bits for the next 1-byte data to be transmitted. The I²C interrupt request signal occurs immediately after the number of count specified with these bits (ACK clock is added to the number of count when ACK clock is selected by ACK bit (bit 7 of address 0016₁₆)) have been transferred, and BC0 to BC2 are returned to "000₂".

Also when a START condition is received, these bits become "000₂" and the address data is always transmitted and received in 8 bits.

•Bit 3: I²C interface enable bit (ES0)

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

When ES0 = "0," the following is performed.

- PIN = "1," BB = "0" and AL = "0" are set (which are bits of the I²C status register at address 0014₁₆).
- Writing data to the I²C data shift register (address 0012₁₆) is disabled.

•Bit 4: Data format selection bit (ALS)

This bit decides whether or not to recognize slave addresses. When this bit is set to "0," the addressing format is selected, so that address data is recognized. When a match is found between a slave address and address data as a result of comparison or when a general call (refer to "(5) I²C Status Register," bit 1) is received, transfer processing can be performed. When this bit is set to "1," the free data format is selected, so that slave addresses are not recognized.

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

This bit selects a slave address specification format. When this bit is set to "0," the 7-bit addressing format is selected. In this case, only the high-order 7 bits (slave address) of the I²C address register (address 0013₁₆) are compared with address data. When this bit is set to "1," the 10-bit addressing format is selected, and all the bits of the I²C address register are compared with address data.

•Bit 6: System clock stop selection bit (CLKSTP)

When executing the WIT or STP instruction, this bit selects the condition of system clock provided to the multi-master I²C-BUS interface. When this bit is set to "0," system clock and operation of the multi-master I²C-BUS interface stop by executing the WIT or STP instruction.

When this bit is set to "1," system clock and operation of the multi-master I²C-BUS interface do not stop even when the WIT instruction is executed.

When the system clock stop selection bit is "1," do not execute the STP instruction.

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

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

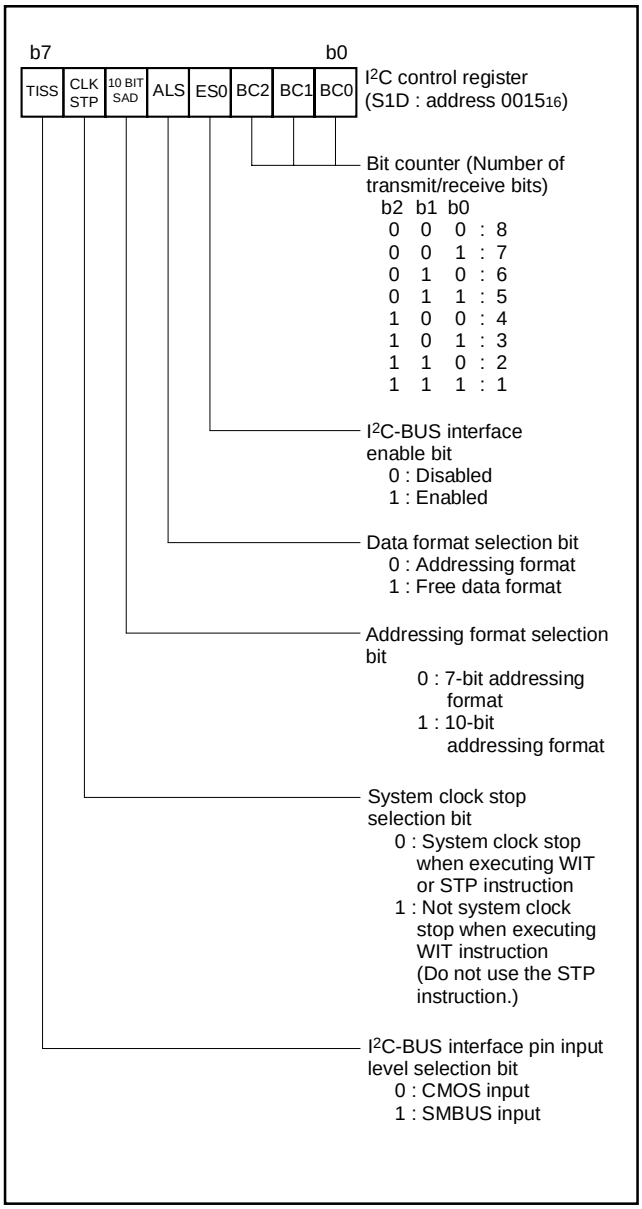


Fig. 40 Structure of I²C control register

[I²C Status Register (S1)] 0014₁₆

The I²C status register (address 0014₁₆) controls the I²C-BUS interface status. The low-order 4 bits are read-only bits and the high-order 4 bits can be read out and written to.

Set "0000₂" to the low-order 4 bits, because these bits become the reserved bits at writing.

•Bit 0: Last receive bit (LRB)

This bit stores the last bit value of received data and can also be used for ACK receive confirmation. If ACK is returned when an ACK clock occurs, the LRB bit is set to "0." If ACK is not returned, this bit is set to "1." Except in the ACK mode, the last bit value of received data is input. The state of this bit is changed from "1" to "0" by executing a write instruction to the I²C data shift register (address 0012₁₆).

•Bit 1: General call detecting flag (AD0)

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

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

•Bit 2: Slave address comparison flag (AAS)

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

① In the slave receive mode, when the 7-bit addressing format is selected, this bit is set to "1" in one of the following conditions:
- The address data immediately after occurrence of a START condition agrees with the slave address stored in the high-order 7 bits of the I²C address register (address 0013₁₆).
- A general call is received.

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

③ This bit is set to "0" by executing a write instruction to the I²C data shift register (address 0012₁₆) when ES0 is set to "1" or reset.

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

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

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

•Bit 4: SCL pin Low hold bit (PIN)

This bit generates an interrupt request signal. Each time 1-byte data is transmitted, the PIN bit changes from "1" to "0." At the same time, an interrupt request signal occurs to the CPU. The PIN bit is set to "0" in synchronization with a falling of the last clock (including the ACK clock) of an internal clock and an interrupt request signal occurs in synchronization with a falling of the PIN bit. When the PIN bit is "0," the SCL is kept in the "0" state and clock generation is disabled. Figure 42 shows an interrupt request signal generating timing chart.

The PIN bit is set to "1" in one of the following conditions:
- Executing a write instruction to the I²C data shift register (address 0012₁₆). (This is the only condition which the prohibition of the internal clock is released and data can be communicated except for the start condition detection.)
- When the ES0 bit is "0"
- At reset
- When writing "1" to the PIN bit by software

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

•Bit 5: Bus busy flag (BB)

This bit indicates the status of use of the bus system. When this bit is set to "0," this bus system is not busy and a START condition can be generated. The BB flag is set/reset by the SCL, SDA pins input signal regardless of master/slave. This flag is set to "1" by detecting the start condition, and is set to "0" by detecting the stop condition. The condition of these detecting is set by the start/stop condition setting bits (SSC4–SSC0) of the I²C start/stop condition control register (address 0017₁₆). When the ES0 bit (bit 3) of the I²C control register (address 0015₁₆) is "0" or reset, the BB flag is set to "0."

For the writing function to the BB flag, refer to the sections "START Condition Generating Method" and "STOP Condition Generating Method" described later.

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

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

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

- When ALS is "0"
- In the slave reception mode or the slave transmission mode
- When the R/$\overline{W}$ bit reception is "1"

This bit is set to "0" in one of the following conditions:

- When arbitration lost is detected.
- When a STOP condition is detected.
- When writing "1" to this bit by software is invalid by the START condition duplication preventing function **(Note)**.
- With MST = "0" and when a START condition is detected.
- With MST = "0" and when ACK non-return is detected.
- At reset

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

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

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

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

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

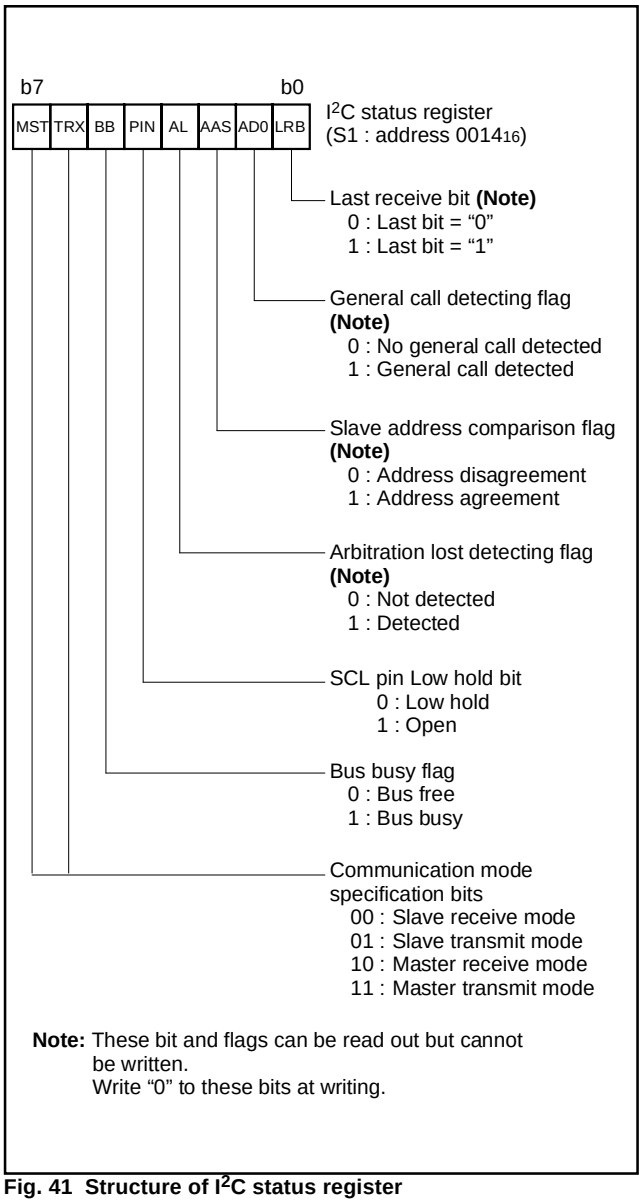


Fig. 41 Structure of I2C status register

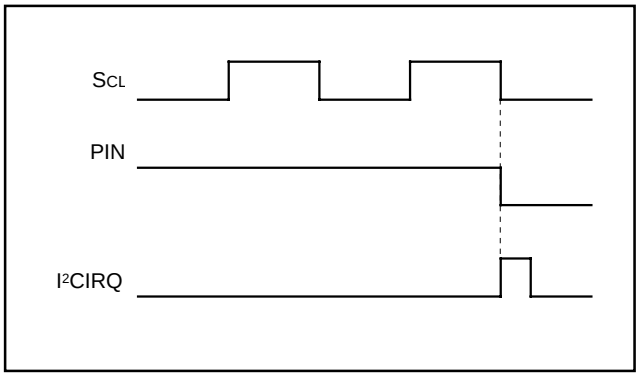


Fig. 42 Interrupt request signal generating timing

START Condition Generating Method

When writing "1" to the MST, TRX, and BB bits of the I^2C status register (address 0014_{16}) at the same time after writing the slave address to the I^2C data shift register (address 0012_{16}) with the condition in which the ES0 bit of the I^2C control register (address 0015_{16}) and the BB flag are "0", a START condition occurs. After that, the bit counter becomes "000_2" and an SCL for 1 byte is output. The START condition generating timing is different in the standard clock mode and the high-speed clock mode. Refer to Figure 43, the START condition generating timing diagram, and Table 13, the START condition generating timing table.

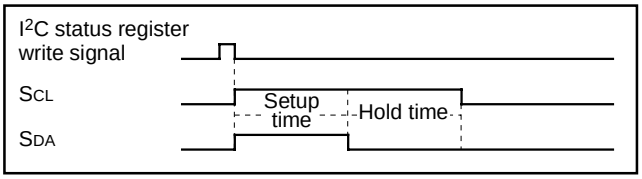


Fig. 43 START condition generating timing diagram

Table 13 START condition generating timing table

Item	START/STOP condition generating selection bit	Standard clock mode	High-speed clock mode
Setup time	"0"	5.0 µs (20 cycles)	2.5 µs (10 cycles)
	"1"	13.0 µs (52 cycles)	6.5 µs (26 cycles)
Hold time	"0"	5.0 µs (20 cycles)	2.5 µs (10 cycles)
	"1"	13.0 µs (52 cycles)	6.5 µs (26 cycles)

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

STOP Condition Generating Method

When the ES0 bit of the I^2C control register (address 0015_{16}) is "1," write "1" to the MST and TRX bits, and write "0" to the BB bit of the I^2C status register (address 0014_{16}) simultaneously. Then a STOP condition occurs. The STOP condition generating timing is different in the standard clock mode and the high-speed clock mode. Refer to Figure 44, the STOP condition generating timing diagram, and Table 14, the STOP condition generating timing table.

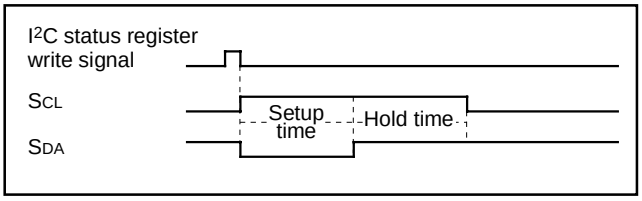


Fig. 44 STOP condition generating timing diagram

Table 14 STOP condition generating timing table

Item	START/STOP condition generating selection bit	Standard clock mode	High-speed clock mode
Setup time	"0"	5.5 µs (22 cycles)	3.0 µs (12 cycles)
	"1"	13.5 µs (54 cycles)	7.0 µs (28 cycles)
Hold time	"0"	5.5 µs (22 cycles)	3.0 µs (12 cycles)
	"1"	13.5 µs (54 cycles)	7.0 µs (28 cycles)

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

START/STOP Condition Detecting Operation

The START/STOP condition detection operations are shown in Figures 45, 46, and Table 15. The START/STOP condition is set by the START/STOP condition set bit.

The START/STOP condition can be detected only when the input signal of the SCL and SDA pins satisfy three conditions: SCL release time, setup time, and hold time (see Table 15).

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

The BB flag set/reset timing is different in the standard clock mode and the high-speed clock mode. Refer to Table 15, the BB flag set/reset time.

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

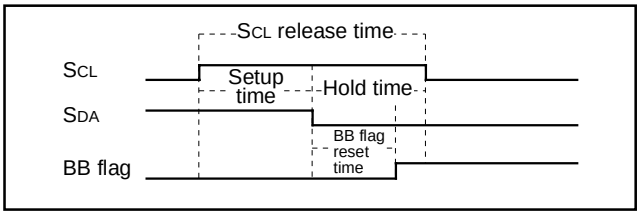


Fig. 45 START condition detecting timing diagram

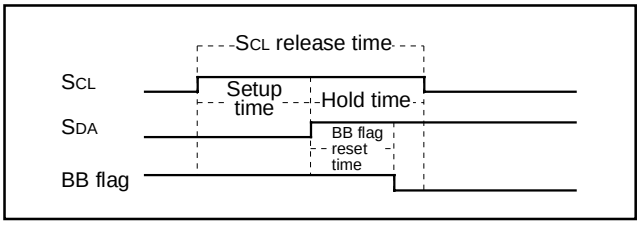


Fig. 46 STOP condition detecting timing diagram

Table 15 START condition/STOP condition detecting conditions

	Standard clock mode	High-speed clock mode
SCL release time	SSC value + 1 cycle (6.25 µs)	4 cycles (1.0 µs)
Setup time	$\frac{\text{SSC value}}{2}$ + 1 cycle < 4.0 µs (3.25 µs)	2 cycles (1.0 µs)
Hold time	$\frac{\text{SSC value}}{2}$ cycle < 4.0 µs (3.0 µs)	2 cycles (0.5 µs)
BB flag set/ reset time	$\frac{\text{SSC value} - 1}{2}$ + 2 cycles (3.375 µs)	3.5 cycles (0.875 µs)

Note: Unit : Cycle number of system clock ϕ

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

[I²C START/STOP Condition Control Register (S2D)] 0017₁₆

The I²C START/STOP condition control register (address 0017₁₆) controls START/STOP condition detection.

•Bits 0 to 4: START/STOP condition set bit (SSC4–SSC0)

SCL release time, setup time, and hold time change the detection condition by value of the main clock divide ratio selection bit and the oscillation frequency f(XIN) because these time are measured by the internal system clock. Accordingly, set the proper value to the START/STOP condition set bits (SSC4 to SSC0) in considered of the system clock frequency. Refer to Table 16.

Do not set "00000₂" or an odd number to the START/STOP condition set bit (SSC4 to SSC0).

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

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

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

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

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

•Bit 7: START/STOP condition generating selection bit (STSPSEL)

Setup/Hold time when the START/STOP condition is generated can be selected.

Cycle number of system clock becomes standard for setup/hold time. Additionally, setup/hold time is different between the START condition and the STP condition. (Refer to Tables 13 and 14.) Set "1" to this bit when the system clock frequency is 4 MHz or more.

Address Data Communication

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

① 7-bit addressing format

To adapt the 7-bit addressing format, set the 10BIT SAD bit of the I²C control register (address 0015₁₆) to "0." The first 7-bit address data transmitted from the master is compared with the high-order 7-bit slave address stored in the I²C address register (address 0013₁₆). At the time of this comparison, address comparison of the RWB bit of the I²C address register (address 0013₁₆) is not performed. For the data transmission format when the 7-bit addressing format is selected, refer to Figure 48, (1) and (2).

② 10-bit addressing format

To adapt the 10-bit addressing format, set the 10BIT SAD bit of the I²C control register (address 0015₁₆) to "1." An address comparison is performed between the first-byte address data transmitted from the master and the 8-bit slave address stored in the I²C address register (address 0013₁₆). At the time of this comparison, an address comparison between the RWB bit of the I²C address register (address 0013₁₆) and the R/W bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the RWB bit which is the last bit of the address data not only specifies the direction of communication for control data, but also is processed as an address data bit.

When the first-byte address data agree with the slave address, the AAS bit of the I²C status register (address 0014₁₆) is set to "1." After the second-byte address data is stored into the I²C data shift register (address 0012₁₆), perform an address comparison between the second-byte data and the slave address by software. When the address data of the 2 bytes agree with the slave address, set the RWB bit of the I²C address register (address 0013₁₆) to "1" by software. This processing can make the 7-bit slave address and R/$\overline{W}$ data agree, which are received after a RESTART condition is detected, with the value of the I²C address register (address 0013₁₆). For the data transmission format when the 10-bit addressing format is selected, refer to Figure 48, (3) and (4).

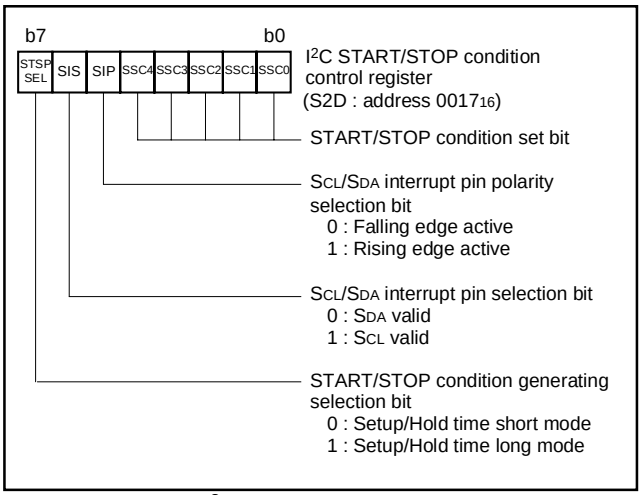


Fig. 47 Structure of I²C START/STOP condition control register

Table 16 Recommended set value to START/STOP condition set bits (SSC4–SSC0) for each oscillation frequency

Oscillation frequency f(XIN) (MHz)	Main clock divide ratio	System clock ϕ (MHz)	START/STOP condition control register	SCL release time (μs)	Setup time (μs)	Hold time (μs)
10	2	5	XXX11110	6.2 μs (31 cycles)	3.2 μs (16 cycles)	3.0 μs (15 cycles)
8	2	4	XXX11010	6.75 μs (27 cycles)	3.5 μs (14 cycles)	3.25 μs (13 cycles)
			XXX11000	6.25 μs (25 cycles)	3.25 μs (13 cycles)	3.0 μs (12 cycles)
8	8	1	XXX00100	5.0 μs (5 cycles)	3.0 μs (3 cycles)	2.0 μs (2 cycles)
4	2	2	XXX01100	6.5 μs (13 cycles)	3.5 μs (7 cycles)	3.0 μs (6 cycles)
			XXX01010	5.5 μs (11 cycles)	3.0 μs (6 cycles)	2.5 μs (5 cycles)
2	2	1	XXX00100	5.0 μs (5 cycles)	3.0 μs (3 cycles)	2.0 μs (2 cycles)

Note: Do not set "00002" or an odd number to the START/STOP condition set bit (SSC4 to SSC0).

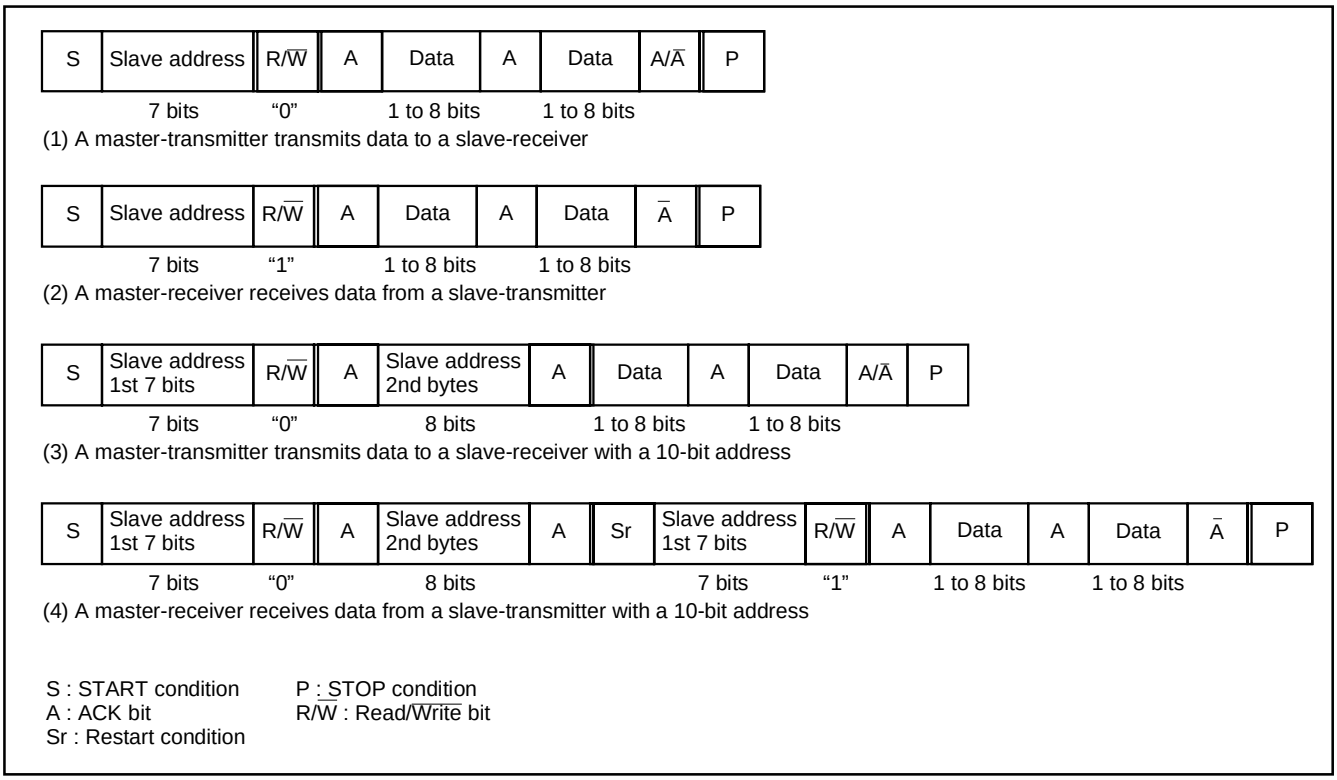


Fig. 48 Address data communication format

Example of Master Transmission

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

① Set a slave address in the high-order 7 bits of the I^2C address register (address 0013_{16}) and "0" into the RWB bit.
② Set the ACK return mode and SCL = 100 kHz by setting "85_{16}" in the I^2C clock control register (address 0016_{16}).
③ Set "00_{16}" in the I^2C status register (address 0014_{16}) so that transmission/reception mode can become initializing condition.
④ Set a communication enable status by setting "08_{16}" in the I^2C control register (address 0015_{16}).
⑤ Confirm the bus free condition by the BB flag of the I^2C status register (address 0014_{16}).
⑥ Set the address data of the destination of transmission in the high-order 7 bits of the I^2C data shift register (address 0012_{16}) and set "0" in the least significant bit.
⑦ Set "$F0_{16}$" in the I^2C status register (address 0014_{16}) to generate a START condition. At this time, an SCL for 1 byte and an ACK clock automatically occur.
⑧ Set transmit data in the I^2C data shift register (address 0012_{16}). At this time, an SCL and an ACK clock automatically occur.
⑨ When transmitting control data of more than 1 byte, repeat step ⑧.
⑩ Set "$D0_{16}$" in the I^2C status register (address 0014_{16}) to generate a STOP condition if ACK is not returned from slave reception side or transmission ends.

Example of Slave Reception

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

① Set a slave address in the high-order 7 bits of the I^2C address register (address 0013_{16}) and "0" in the RWB bit.
② Set the no ACK clock mode and SCL = 400 kHz by setting "25_{16}" in the I^2C clock control register (address 0016_{16}).
③ Set "00_{16}" in the I^2C status register (address 0014_{16}) so that transmission/reception mode can become initializing condition.
④ Set a communication enable status by setting "08_{16}" in the I^2C control register (address 0015_{16}).
⑤ When a START condition is received, an address comparison is performed.
⑥ •When all transmitted addresses are "0" (general call):
AD0 of the I^2C status register (address 0014_{16}) is set to "1" and an interrupt request signal occurs.
• When the transmitted addresses agree with the address set in ①:
AAS of the I^2C status register (address 0014_{16}) is set to "1" and an interrupt request signal occurs.
• In the cases other than the above AD0 and AAS of the I^2C status register (address 0014_{16}) are set to "0" and no interrupt request signal occurs.
⑦ Set dummy data in the I^2C data shift register (address 0012_{16}).
⑧ When receiving control data of more than 1 byte, repeat step ⑦.
⑨ When a STOP condition is detected, the communication ends.

■Precautions when using multi-master I^2C-BUS interface

(1) Read-modify-write instruction

The precautions when the read-modify-write instruction such as SEB, CLB etc. is executed for each register of the multi-master I^2C-BUS interface are described below.

- I^2C data shift register (S0: address 0012_{16})
  When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.
- I^2C address register (S0D: address 0013_{16})
  When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended. It is because H/W changes the read/write bit (RWB) at the above timing.
- I^2C status register (S1: address 0014_{16})
  Do not execute the read-modify-write instruction for this register because all bits of this register are changed by H/W.
- I^2C control register (S1D: address 0015_{16})
  When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended. Because H/W changes the bit counter (BC0-BC2) at the above timing.
- I^2C clock control register (S2: address 0016_{16})
  The read-modify-write instruction can be executed for this register.
- I^2C START/STOP condition control register (S2D: address 0017_{16})
  The read-modify-write instruction can be executed for this register.

(2) START condition generating procedure using multi-master

1. Procedure example (The necessary conditions of the generating procedure are described in Items 2 to 5 below.

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

2. Use “Branch on Bit Set” of “BBS 5, $0014, –” for the BB flag confirming and branch process.
3. Use “STA $12, STX $12” or “STY $12” of the zero page addressing instruction for writing the slave address value to the I^2C data shift register.
4. Execute the branch instruction of Item 2 and the store instruction of Item 3 continuously, as shown in the procedure example above.
5. Disable interrupts during the following three process steps:
   • BB flag confirming
   • Writing of slave address value
   • Trigger of START condition generating

   When the condition of the BB flag is bus busy, enable interrupts immediately.

(3) RESTART condition generating procedure

This procedure cannot be applied to M38867M8A and M38867E8A when the external memory is used and the bus cycle is extended by $\overline{\text{ONW}}$ function.

1. Procedure example (The necessary conditions for the procedure are described in items 2 to 4 below.)
   Execute the following procedure when the PIN bit is “0.”

```
      ⋮
  LDM #$00, S1           (Select slave receive mode)
  LDA —                  (Take out of slave address value)
  SEI                    (Disable interrupt)
  STA S0                 (Write slave address value)
  LDM #$F0, S1           (Trigger  RESTART condition generation)
  CLI                    (Enable interrupt)
      ⋮
```

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

(4) Writing to I^2C status register

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

(5) Process of after STOP condition generating

Do not write data in the I^2C data shift register S0 and the I^2C status register S1 until the bus busy flag BB becomes “0” after generating the STOP condition in the master mode. Because the STOP condition waveform might not be normally generated. Reading to the above registers do not have the problem.

(6) STOP condition input at 7th clock pulse

The SDA line may be held at LOW even if flag BB is set to “0” when all the following conditions are satisfied:

•In the slave mode
•The STOP condition is input at the 7th clock pulse while receiving a slave address or data.
•The clock pulse is continuously input.

Countermeasure:

Write dummy data to the I^2C shift register or reset the ES0 bit in the S1D register (ES0 = “L” → ES0 = “H”) during a stop condition interrupt routine with flag PIN = “1”.

Note: Do not use the read-modify-write instruction at this time. Furthermore, when the ES0 bit is set to “0”, the SDA pin becomes a general-purpose port; the port must be set to input mode or output “H”.

(7) ES0 bit switch

In standard clock mode when SSC = “00010_2” or in high-speed clock mode, flag BB may switch to “1” if ES0 bit is set to “1” when SDA is “L”.

Countermeasure:

Set ES0 to “1” when SDA is “H”.

A-D CONVERTER
[A-D Conversion Register 1,2 (AD1, AD2)] 0035_{16}, 0038_{16}

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

Bit 7 of the A-D conversion register 2 is the conversion mode selection bit. When this bit is set to "0," the A-D converter becomes the 10-bit A-D mode. When this bit is set to "1," that becomes the 8-bit A-D mode. The conversion result of the 8-bit A-D mode is stored in the A-D conversion register 1. As for 10-bit A-D mode, 10-bit reading or 8-bit reading can be performed by selecting the reading procedure of the A-D conversion register 1, 2 after A-D conversion is completed (in Figure 50).

The A-D conversion register 1 performs the 8-bit reading inclined to MSB after reset, the A-D conversion is started, or reading of the A-D converter register 1 is generated; and the register becomes the 8-bit reading inclined to LSB after the A-D converter register 2 is generated.

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

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

Comparison Voltage Generator

The comparison voltage generator divides the voltage between AV_{SS} and V_{REF} into 1024, and outputs the divided voltages in the 10-bit A-D mode (256 division in 8-bit A-D mode).

The A-D converter successively compares the comparison voltage V_{ref} in each mode, dividing the V_{REF} (see below), with the input voltage.

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

$$V_{ref} = \frac{V_{REF}}{1024} \times n \quad (n = 0–1023)$$

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

$$V_{ref} = \frac{V_{REF}}{256} \times n \quad (n = 0–255)$$

- 8-bit A-D mode

$$V_{ref} = \frac{V_{REF}}{256} \times (n–0.5) \quad (n = 1–255)$$
$$= 0 \quad (n = 0)$$

Channel Selector

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

Comparator and Control Circuit

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

Note that because the comparator consists of a capacitor coupling, set $f(X_{IN})$ to 500 kHz or more during an A-D conversion.

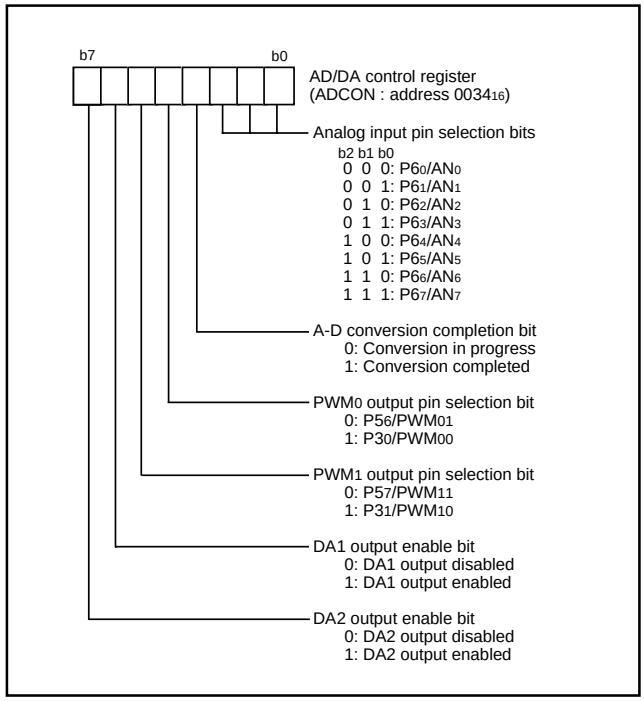


Fig. 49 Structure of AD/DA control register

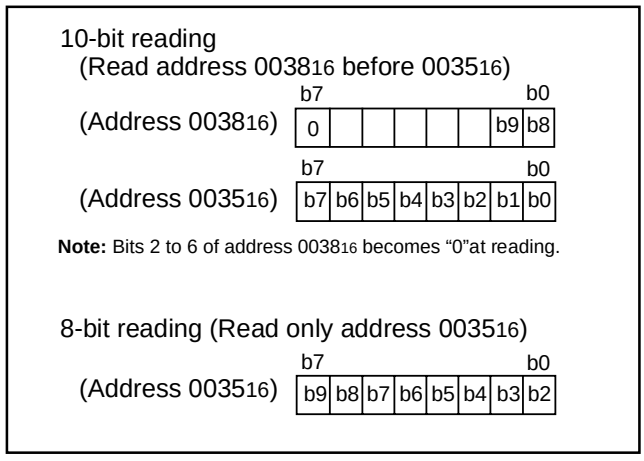


Fig. 50 Structure of 10-bit A-D mode reading

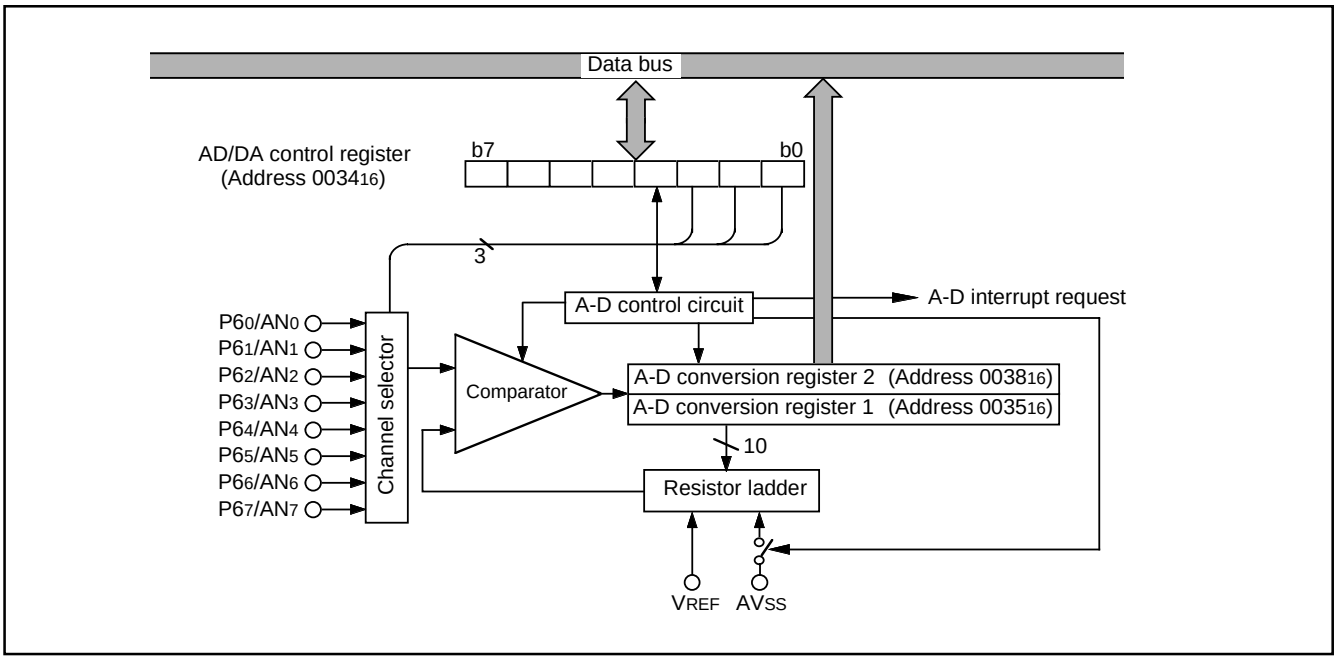


Fig. 51 Block diagram of A-D converter

D-A CONVERTER

The 3886 group has two internal D-A converters (DA_1 and DA_2) with 8-bit resolution.

The D-A converter is performed by setting the value in each D-A conversion register. The result of D-A conversion is output from the DA_1 or DA_2 pin by setting the DA output enable bit to "1".

When using the D-A converter, the corresponding port direction register bit ($P5_6/DA_1/PWM_{01}$ or $P5_7/DA_2/PWM_{11}$) must be set to "0" (input status).

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

$V = V_{REF} \times n/256$ (n = 0 to 255)

Where V_{REF} is the reference voltage.

At reset, the D-A conversion registers are cleared to "00_{16}", the DA output enable bits are cleared to "0", and the $P5_6/DA_1/PWM_{01}$ and $P5_7/DA_2/PWM_{11}$ pins become high impedance.

The DA output does not have buffers. Accordingly, connect an external buffer when driving a low-impedance load.

Set V_{CC} to 4.0 V or more when using the D-A converter.

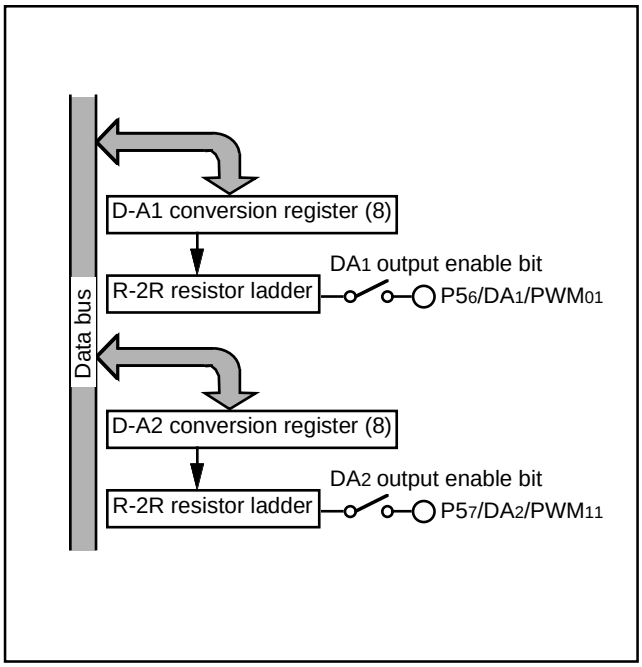


Fig. 52 Block diagram of D-A converter

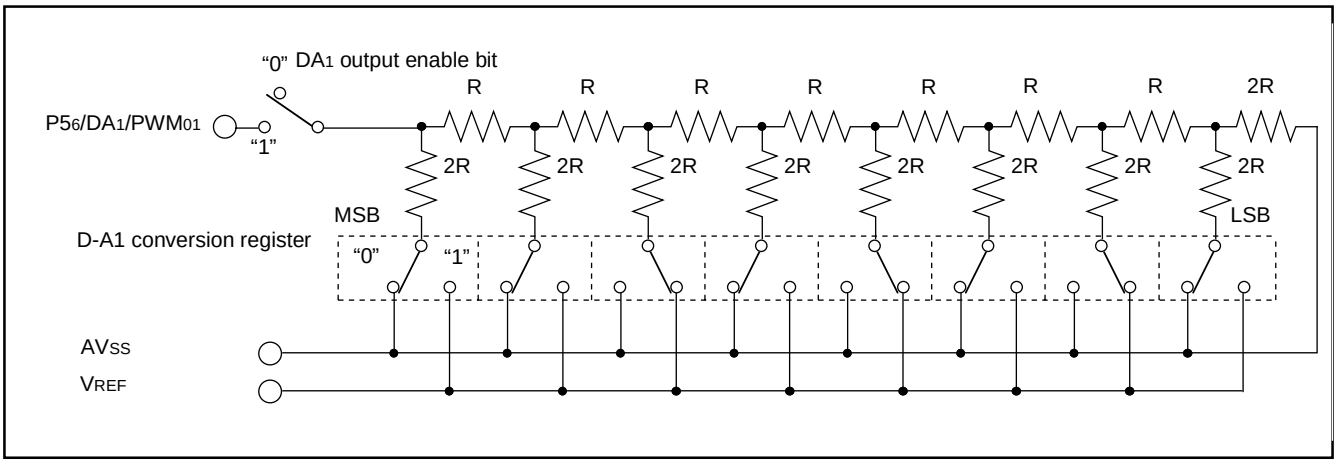


Fig. 53 Equivalent connection circuit of D-A converter (DA1)

COMPARATOR CIRCUIT
Comparator Configuration

The comparator circuit consists of resistors, comparators, a comparator control circuit, the comparator reference input selection bit (bit 7 of address $001D_{16}$), a comparator data register (address $002D_{16}$), the comparator reference power source input pin ($P0_0$/$P3_{REF}$) and analog signal input pins ($P3_0$–$P3_7$). The analog input pin ($P3_0$–$P3_7$) also functions as an ordinary digital port.

Comparator Operation

To activate the comparator, first set port P3 to input mode by setting the corresponding direction register (address 0007_{16}) to "0" to use port P3 as an analog voltage input pin. The internal fixed analog voltage (V_{CC} × 29/32) can be generated by setting "1" to the comparator reference input selection bit (bit 7) of the serial I/O2 control register (address $001D_{16}$). (The internal fixed analog voltage becomes about 4.5 V at V_{CC} = 5.0 V.) When setting "0" to the comparator reference input selection bit, the $P0_0$/$P3_{REF}$ pin becomes the comparator reference power source input pin and it is possible to input the comparator reference power source optionally from the external. The voltage comparison is immediately performed by the writing operation to the comparator data register (address $002D_{16}$). After 14 cycles of the internal system clock ϕ (the time required for the comparison), the comparison result is stored in the comparator data register (address $002D_{16}$).

If the analog input voltage is greater than the internal reference voltage, each bit of this register is "1"; if it is less than the internal reference voltage, each bit of this register is "0". To perform another comparison, the voltage comparison must be performed again by writing to the comparator data register (address $002D_{16}$). Read the result when 14 cycles of ϕ or more have passed after the comparator operation starts. The ladder resistor is turned on during 14 cycles of ϕ , which is required for the comparison, and the reference voltage is generated. An unnecessary current is not consumed because the ladder resistor is turned off while the comparator operation is not performed. Since the comparator consists of capacitor coupling, the electric charge is lost if the clock frequency is low.

Keep that the clock frequency is 1 MHz or more during the comparator operation. Do not execute the STP, WIT, or port P3 I/O instruction.

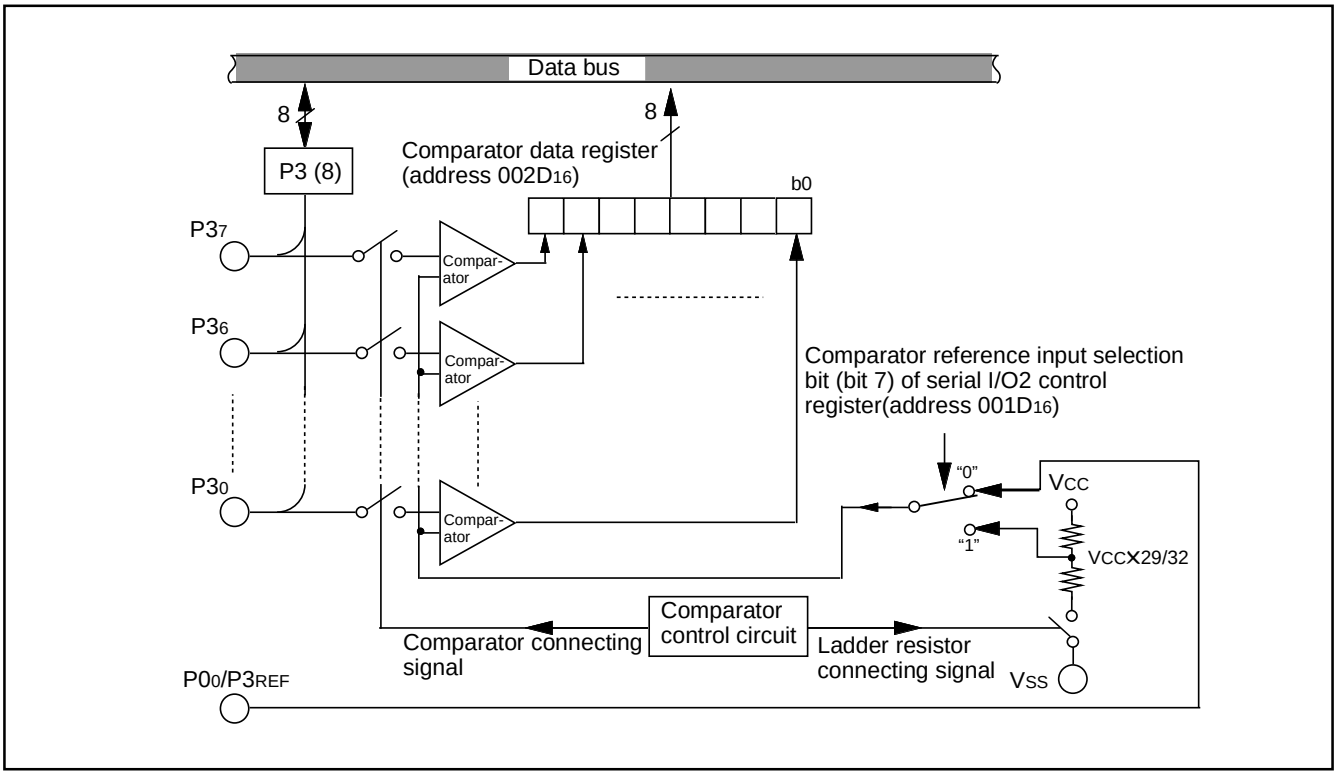


Fig. 54 Comparator circuit

WATCHDOG TIMER

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

Standard Operation of Watchdog Timer

When any data is not written into the watchdog timer control register (address 001E16) after resetting, the watchdog timer is in the stop state. The watchdog timer starts to count down by writing an optional value into the watchdog timer control register (address 001E16) and an internal reset occurs at an underflow of the watchdog timer H.

Accordingly, programming is usually performed so that writing to the watchdog timer control register (address 001E16) may be started before an underflow. When the watchdog timer control register (address 001E16) is read, the values of the high-order 6 bits of the watchdog timer H, STP instruction disable bit, and watchdog timer H count source selection bit are read.

Initial Value of Watchdog Timer

At reset or writing to the watchdog timer control register (address 001E16), each watchdog timer H and L is set to "FF16."

●Watchdog timer H count source selection bit operation

Bit 7 of the watchdog timer control register (address 001E16) permits selecting a watchdog timer H count source. When this bit is set to "0", the count source becomes the underflow signal of watchdog timer L. The detection time is set to f(XIN)=131.072 ms at 8 MHz frequency and f(XCIN)=32.768 s at 32 kHz frequency.

When this bit is set to "1", the count source becomes the signal divided by 16 for f(XIN) (or f(XCIN)). The detection time in this case is set to f(XIN)= 512 μs at 8 MHz frequency and f(XCIN)=128 ms at 32 kHz frequency. This bit is cleared to "0" after resetting.

●Operation of STP instruction disable bit

Bit 6 of the watchdog timer control register (address 001E16) permits disabling the STP instruction when the watchdog timer is in operation.

When this bit is "0", the STP instruction is enabled.

When this bit is "1", the STP instruction is disabled.

Once the STP instruction is executed, an internal reset occurs. When this bit is set to "1", it cannot be rewritten to "0" by program. This bit is cleared to "0" after resetting.

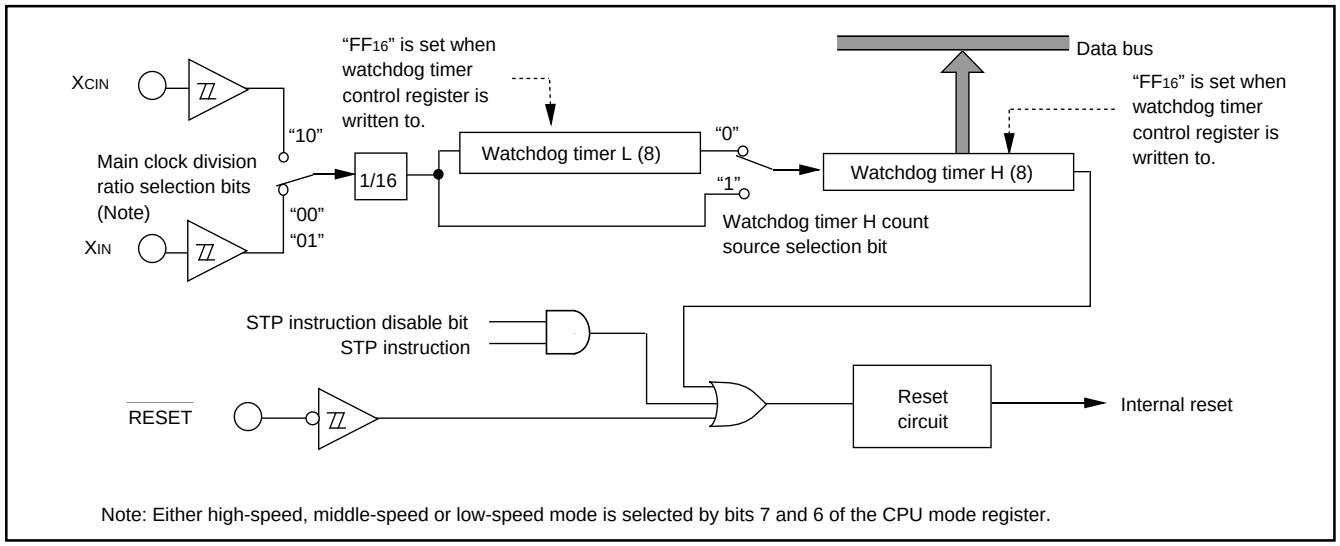


Fig. 55 Block diagram of Watchdog timer

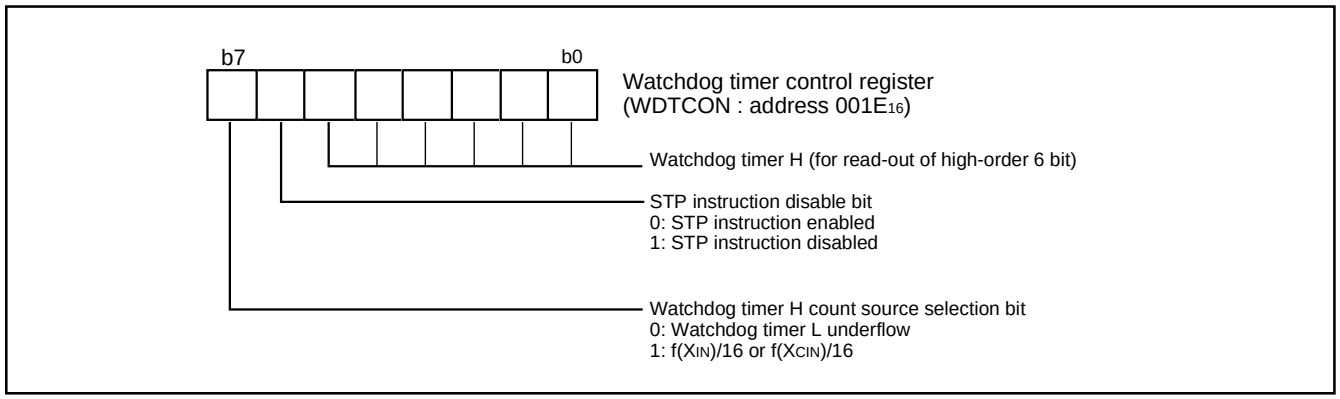


Fig. 56 Structure of Watchdog timer control register

RESET CIRCUIT

To reset the microcomputer, $\overline{\text{RESET}}$ pin should be held at an "L" level for 16 cycles or more of X_{IN}. Then the $\overline{\text{RESET}}$ pin is returned to an "H" level (the power source voltage should be between 2.7 V and 5.5 V (4.0 V to 5.5 V for flash memory version), and the oscillation should be stable), reset is released. After the reset is completed, the program starts from the address contained in address $FFFD_{16}$ (high-order byte) and address $FFFC_{16}$ (low-order byte). Make sure that the reset input voltage is less than 0.54 V for V_{CC} of 2.7 V. For flash memory version, make sure that the reset input voltage is less than 0.8 V for Vcc of 4.0 V.

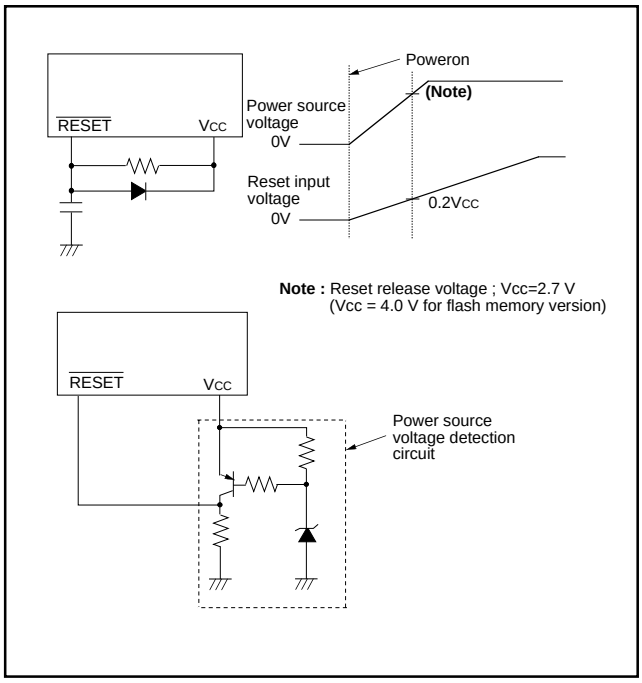


Fig. 57 Reset circuit example

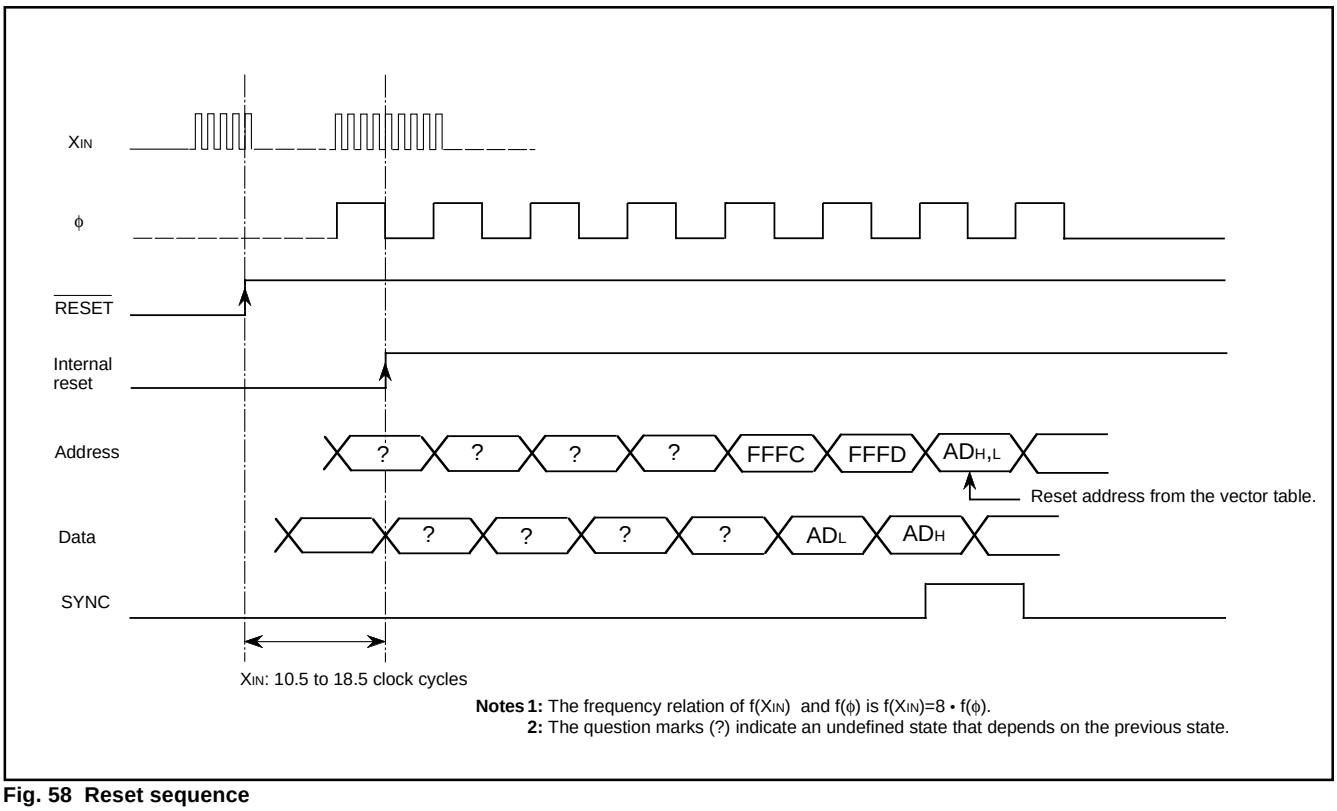


Fig. 58 Reset sequence

	Address	Register contents
(1) Port P0 (P0)	0000_{16}	00_{16}
(2) Port P0 direction register (P0D)	0001_{16}	00_{16}
(3) Port P1 (P1)	0002_{16}	00_{16}
(4) Port P1 direction register (P1D)	0003_{16}	00_{16}
(5) Port P2 (P2)	0004_{16}	00_{16}
(6) Port P2 direction register (P2D)	0005_{16}	00_{16}
(7) Port P3 (P3)	0006_{16}	00_{16}
(8) Port P3 direction register (P3D)	0007_{16}	00_{16}
(9) Port P4 (P4)	0008_{16}	00_{16}
(10) Port P4 direction register (P4D)	0009_{16}	00_{16}
(11) Port P5 (P5)	$000A_{16}$	00_{16}
(12) Port P5 direction register (P5D)	$000B_{16}$	00_{16}
(13) Port P6 (P6)	$000C_{16}$	00_{16}
(14) Port P6 direction register (P6D)	$000D_{16}$	00_{16}
(15) Port P7 (P7)	$000E_{16}$	00_{16}
(16) Port P7 direction register (P7D)	$000F_{16}$	00_{16}
(17) Port P8 (P8)	0010_{16}	00_{16}
(18) Port P8 direction register (P8D)	0011_{16}	00_{16}
(19) I^2C data shift register (S0)	0012_{16}	X X X X X X X X
(20) I^2C address register (S0D)	0013_{16}	00_{16}
(21) I^2C status register (S1)	0014_{16}	0 0 0 1 0 0 0 X
(22) I^2C control register (S1D)	0015_{16}	00_{16}
(23) I^2C clock control register (S2)	0016_{16}	00_{16}
(24) I^2C start/stop condition control register (S2D)	0017_{16}	0 0 0 1 1 0 1 0
(25) Transmit/Receive buffer register (TB/RB)	0018_{16}	X X X X X X X X
(26) Serial I/O1 status register (SIO1STS)	0019_{16}	1 0 0 0 0 0 0 0
(27) Serial I/O1 control register (SIO1CON)	$001A_{16}$	00_{16}
(28) UART control register (UARTCON)	$001B_{16}$	1 1 1 0 0 0 0 0
(29) Baud rate generator (BRG)	$001C_{16}$	X X X X X X X X
(30) Serial I/O2 control register (SIO2CON)	$001D_{16}$	00_{16}
(31) Watchdog timer control register (WDTCON)	$001E_{16}$	0 0 1 1 1 1 1 1
(32) Serial I/O2 register (SIO2)	$001F_{16}$	X X X X X X X X

	Address	Register contents
(33) Prescaler 12 (PRE12)	0020_{16}	FF_{16}
(34) Timer 1 (T1)	0021_{16}	01_{16}
(35) Timer 2 (T2)	0022_{16}	FF_{16}
(36) Timer XY mode register (TM)	0023_{16}	00_{16}
(37) Prescaler X (PREX)	0024_{16}	FF_{16}
(38) Timer X (TX)	0025_{16}	FF_{16}
(39) Prescaler Y (PREY)	0026_{16}	FF_{16}
(40) Timer Y (TY)	0027_{16}	FF_{16}
(41) Data bus buffer register 0 (DBB0)	0028_{16}	X X X X X X X X
(42) Data bus buffer status register 0 (DBBSTS0)	0029_{16}	00_{16}
(43) Data bus buffer control register (DBBCON)	$002A_{16}$	00_{16}
(44) Data bus buffer register 1 (DBB1)	$002B_{16}$	X X X X X X X X
(45) Data bus buffer status register 1 (DBBSTS1)	$002C_{16}$	00_{16}
(46) Comparator data register (CMPD)	$002D_{16}$	X X X X X X X X
(47) Port control register 1 (PCTL1)	$002E_{16}$	00_{16}
(48) Port control register 2 (PCTL2)	$002F_{16}$	00_{16}
(49) PWM0H register (PWM0H)	0030_{16}	X X X X X X X X
(50) PWM0L register (PWM0L)	0031_{16}	X 0 X X X X X X
(51) PWM1H register (PWM1H)	0032_{16}	X X X X X X X X
(52) PWM1L register (PWM1L)	0033_{16}	X 0 X X X X X X
(53) AD/DA control register (ADCON)	0034_{16}	0 0 0 0 1 0 0 0
(54) A-D conversion register 1 (AD1)	0035_{16}	X X X X X X X X
(55) D-A1 conversion register (DA1)	0036_{16}	00_{16}
(56) D-A2 conversion register (DA2)	0037_{16}	00_{16}
(57) A-D conversion register 2 (AD2)	0038_{16}	0 0 0 0 0 0 X X
(58) Interrupt source selection register (INTSEL)	0039_{16}	00_{16}
(59) Interrupt edge selection register (INTEDGE)	$003A_{16}$	00_{16}
(60) CPU mode register (CPUM)	$003B_{16}$	0 1 0 0 1 0 * 0
(61) Interrupt request register 1 (IREQ1)	$003C_{16}$	00_{16}
(62) Interrupt request register 2 (IREQ2)	$003D_{16}$	00_{16}
(63) Interrupt control register 1 (ICON1)	$003E_{16}$	00_{16}
(64) Interrupt control register 2 (ICON2)	$003F_{16}$	00_{16}
(65) Flash memory control register (FCON)	$0FFE_{16}$	00_{16}
(66) Flash command register (FCMD)	$0FFF_{16}$	00_{16}
(67) Processor status register	(PS)	X X X X X 1 X X
(68) Program counter	(PC_H)	$FFFD_{16}$ contents
	(PC_L)	$FFFC_{16}$ contents

Note: * The initial values depend on level of the CNV_{SS} pin.
X : Not fixed
Since the initial values for other than above mentioned registers and RAM contents are indefinite at reset, they must be set.

Fig. 59 Internal status at reset

CLOCK GENERATING CIRCUIT

The 3886 group has two built-in oscillation circuits. An oscillation circuit can be formed by connecting a resonator between XIN and XOUT (XCIN and XCOUT). Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between XIN and XOUT since a feed-back resistor exists on-chip. However, an external feed-back resistor is needed between XCIN and XCOUT.

Immediately after power on, only the XIN oscillation circuit starts oscillating, and XCIN and XCOUT pins function as I/O ports.

Frequency Control
(1) Middle-speed mode

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

(2) High-speed mode

The internal clock ϕ is half the frequency of XIN.

(3) Low-speed mode

The internal clock ϕ is half the frequency of XCIN.

■Note

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).

(4) Low power dissipation mode

The low power consumption operation can be realized by stopping the main clock XIN in low-speed mode. To stop the main clock, set bit 5 of the CPU mode register to "1." When the main clock XIN is restarted (by setting the main clock stop bit to "0"), set sufficient time for oscillation to stabilize.

Oscillation Control
(1) Stop mode

If the STP instruction is executed, the internal clock ϕ stops at an "H" level, and XIN and XCIN oscillators stop. When the oscillation stabilizing time set after STP instruction released bit is "0," the prescaler 12 is set to "FF16" and timer 1 is set to "0116." When the oscillation stabilizing time set after STP instruction released bit is "1," set the sufficient time for oscillation of used oscillator to stabilize since nothing is set to the prescaler 12 and timer 1.

Either XIN or XCIN divided by 16 is input to the prescaler 12 as count source, and the output of the prescaler 12 is connected to timer 1. Set the timer 1 interrupt enable bit to disabled ("0") before executing the STP instruction. Oscillator restarts when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU (remains at "H") until timer 1 underflows. The internal clock ϕ is supplied for the first time, when timer 1 underflows. Therefore make sure not to set the timer 1 interrupt request bit to "1" before the STP instruction stops the oscillator. When the oscillator is restarted by reset, apply "L" level to the $\overline{\text{RESET}}$ pin until the oscillation is stable since a wait time will not be generated.

(2) Wait mode

If the WIT instruction is executed, the internal clock ϕ stops at an "H" level, but the oscillator does not stop. The internal clock ϕ restarts at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

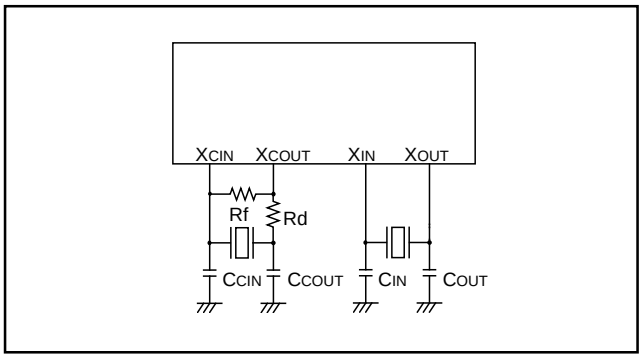


Fig. 60 Ceramic resonator circuit

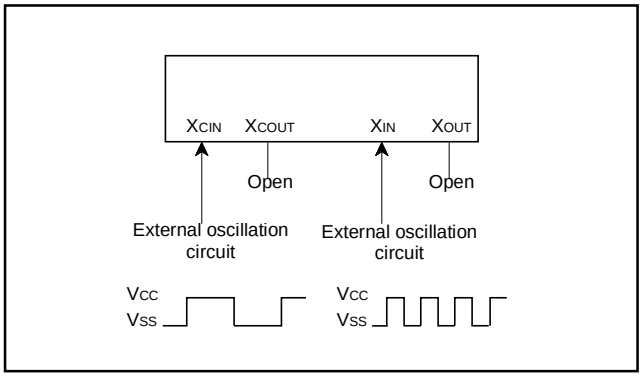
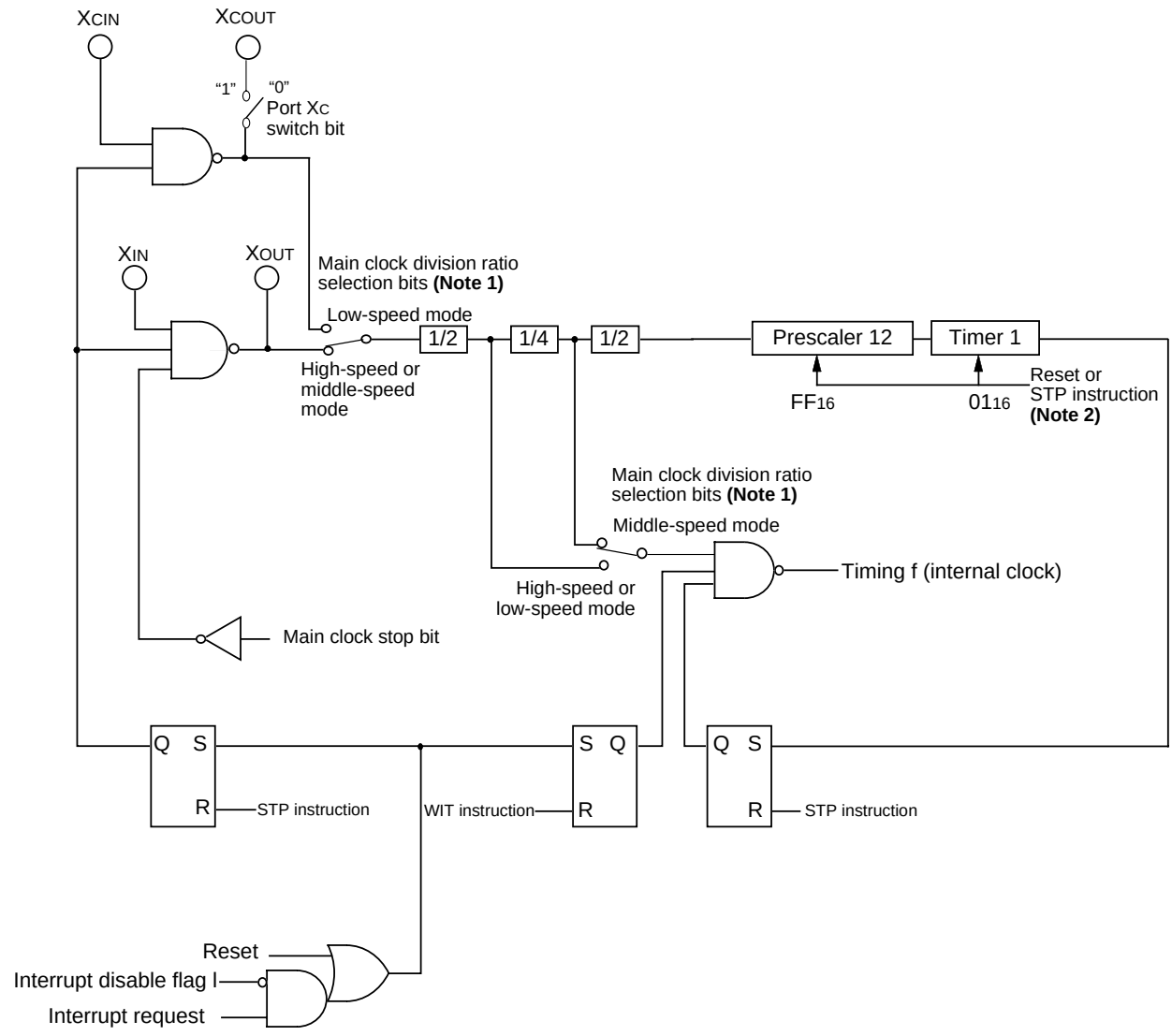
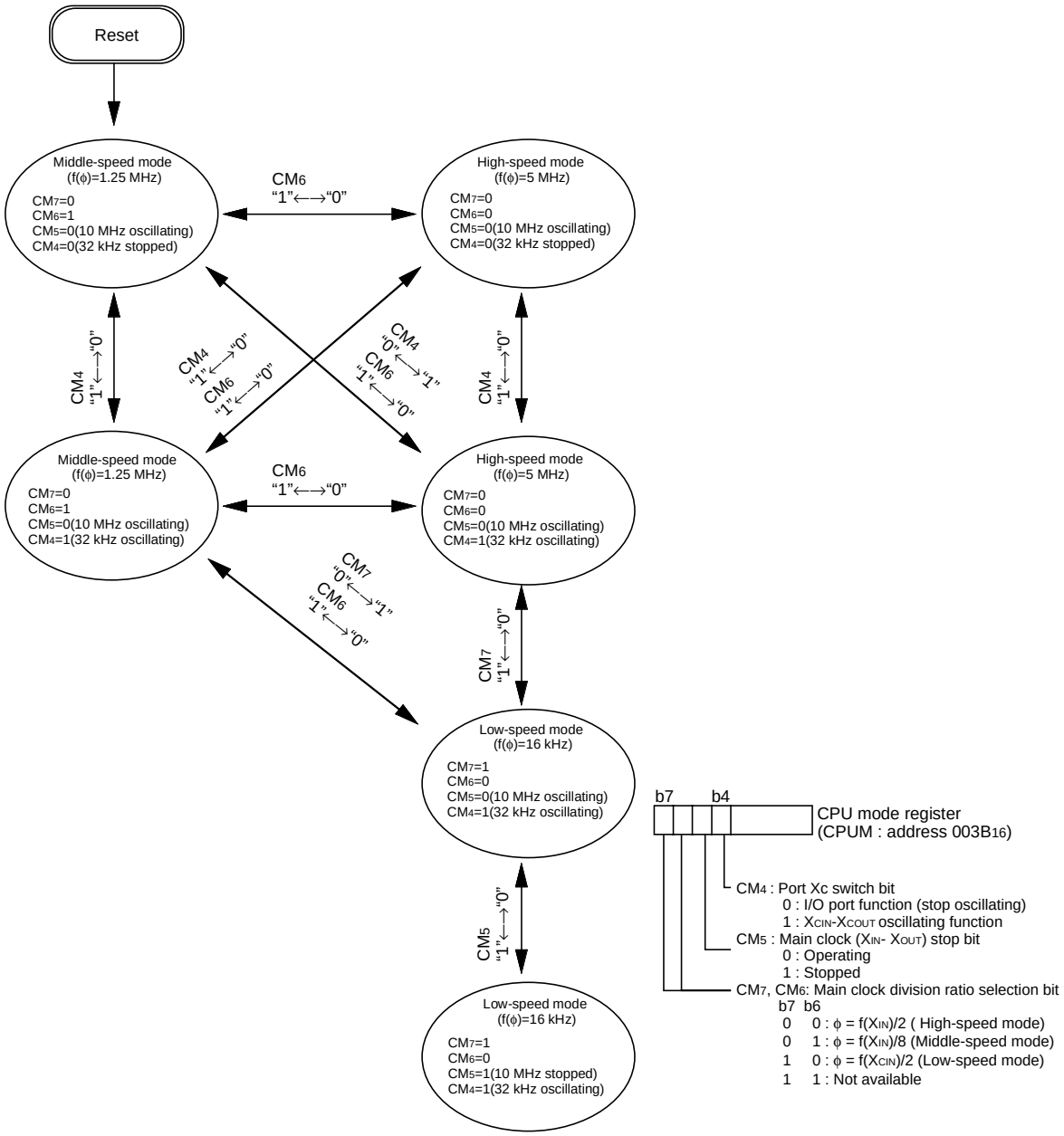


Fig. 61 External clock input circuit



Notes 1: Either high-speed, middle-speed or low-speed mode is selected by bits 7 and 6 of the CPU mode register. When low-speed mode is selected, set port Xc switch bit (b4) to "1".
2: When bit 6 of the port control register 2 is "0", the initial value is not set to the prescaler 12 and timer 1 at STP instruction execution.

Fig. 62 System clock generating circuit block diagram (Single-chip mode)



Notes 1 : Switch the mode by the allows shown between the mode blocks. (Do not switch between the modes directly without an allow.)
2 : The all modes can be switched to the stop mode or the wait mode and return to the source mode when the stop mode or the wait mode is ended.
3 : Timer operates in the wait mode.
4 : When the stop mode is ended, a delay of approximately 1 ms occurs by connecting prescaler 12 and Timer 1 in middle/high-speed mode.
5 : When the stop mode is ended, a delay of approximately 0.25 s occurs by Timer 1 and Timer 2 in low-speed mode.
6 : Wait until oscillation stabilizes after oscillating the main clock XIN before the switching from the low-speed mode to middle/high-speed mode.
7 : The example assumes that 10 MHz is being applied to the XIN pin and 32 kHz to the XCIN pin. ϕ indicates the internal clock.

Fig. 63 State transitions of system clock

PROCESSOR MODE

Single-chip mode, memory expansion mode, and microprocessor mode in the M38867M8A/E8A can be selected by changing the contents of the processor mode bits (CM_0 and CM_1 : b1 and b0 of address $003B_{16}$). In memory expansion mode and microprocessor mode, memory can be expanded externally through ports P0 to P3. In these modes, ports P0 to P3 lose their I/O port functions and become bus pins.

Table 17 Port functions in memory expansion mode and microprocessor mode

Port Name	Function
Port P0	Outputs low-order 8 bits of address.
Port P1	Outputs high-order 8 bits of address.
Port P2	Operates as I/O pins for data D_7 to D_0 (including instruction code).
Port P3	$P3_0$ and $P3_1$ function only as output pins (except that the port latch cannot be read). $P3_2$ is the $\overline{ONW}$ input pin. $P3_3$ is the $\overline{RESET_{OUT}}$ output pin. (**Note**) $P3_4$ is the ϕ output pin. $P3_5$ is the SYNC output pin. $P3_6$ is the $\overline{WR}$ output pin, and $P3_7$ is the $\overline{RD}$ output pin.

Note : If CNV_{SS} is connected to V_{SS}, the microcomputer goes to single-chip mode after a reset, so that this pin cannot be used as the $\overline{RESET_{OUT}}$ output pin.

(1) Single-chip mode

Select this mode by resetting the microcomputer with CNV_{SS} connected to V_{SS}.

(2) Memory expansion mode

Select this mode by setting the processor mode bits (b1, b0) to "01" in software with CNV_{SS} connected to V_{SS}. This mode enables external memory expansion while maintaining the validity of the internal ROM.

However, do not set this mode in the M38869M8A/MCA/MFA and the flash memory version.

(3) Microprocessor mode

Select this mode by resetting the microcomputer with CNV_{SS} connected to V_{CC}, or by setting the processor mode bits to "10" in software with CNV_{SS} connected to V_{SS}. In microprocessor mode, the internal ROM is no longer valid and external memory must be used.

Do not set this mode in the M38869M8A/MCA/MFA and the flash memory version.

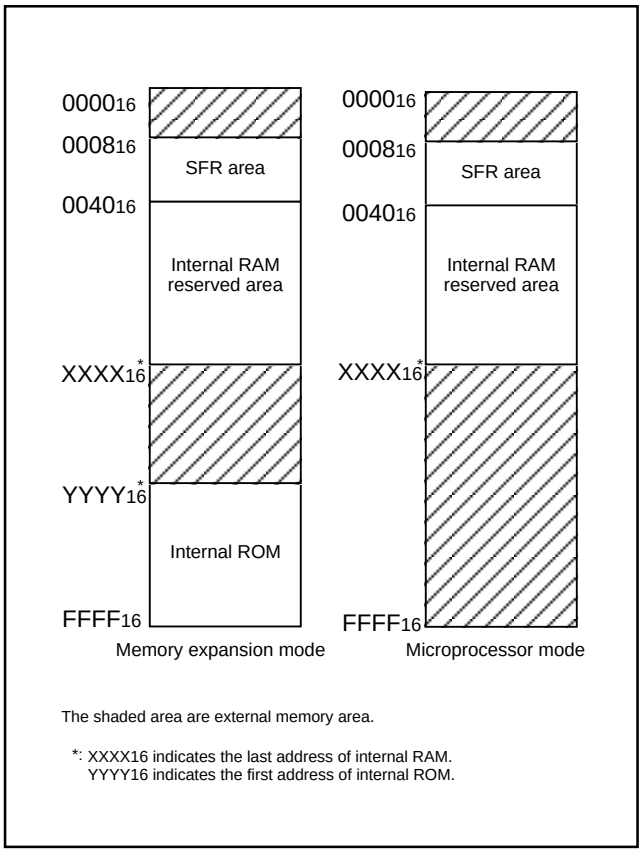


Fig. 64 Memory maps in various processor modes

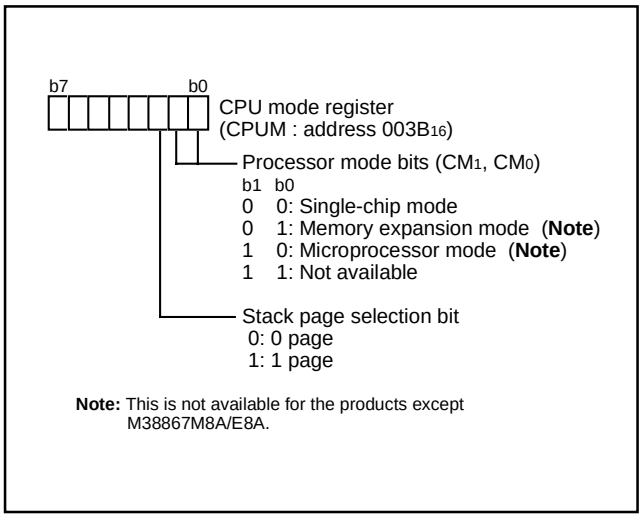
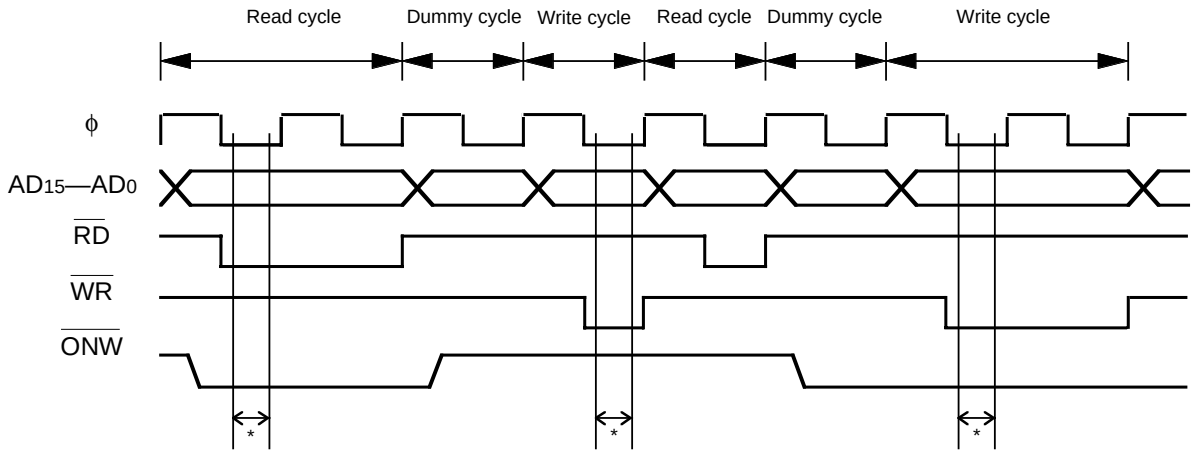


Fig. 65 Structure of CPU mode register

BUS CONTROL AT MEMORY EXPANSION

The M38867M8A/E8A have a built-in $\overline{ONW}$ function to facilitate access to an external (expanded) memory and I/O devices in memory expansion mode or microprocessor mode.

If an "L" level signal is input to the P32/$\overline{ONW}$ pin when the CPU is in a read or write state, the corresponding read or write cycle is extended by one cycle of ϕ. During this extended term, the $\overline{RD}$ and $\overline{WR}$ signals remain at "L." This extension function is valid only for writing to and reading from addresses 0000_{16} to 0007_{16} and 0440_{16} to $FFFF_{16}$, and only read and write cycles are extended.



* Term where $\overline{ONW}$ input signal is received.
During this term, the $\overline{ONW}$ signal must be fixed at either "H" or "L". At all other times, the input level of the $\overline{ONW}$ signal has no affect on operations. The bus cycles is not extended for an address in the area 0008_{16} to $043F_{16}$, because the $\overline{ONW}$ signal is not received.

Fig. 66 $\overline{ONW}$ function timing

EPROM MODE

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. The One Time PROM version and the built-in EPROM version have the function of the M5M27C101 corresponding for writing to the built-in PROM. Set the address of PROM programmer in the user ROM area.

Table 18 Programming adapter

Package	Name of Programming Adapter
80P6Q-A	PCA4738H-80A
80D0	PCA4738L-80A

Table 19 PROM programmer setup

Product name	PROM programmer setup		ROM area of microcomputer
	Corresponding device	Writing area	
M38867E8AHP	M5M27C101K byte program	8080_{16} | $FFFD_{16}$	8080_{16} | $FFFD_{16}$
M38867E8AFS			

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 67 is recommended to verify programming.

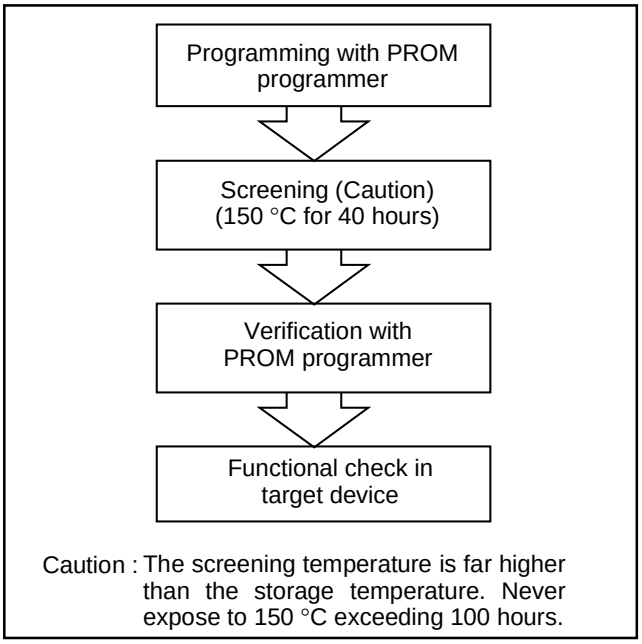


Fig. 67 Programming and testing of One Time PROM version

FLASH MEMORY MODE

The M38869FFAHP/GP has the flash memory mode in addition to the normal operation mode (microcomputer mode). The user can use this mode to perform read, program, and erase operations for the internal flash memory.

The M38869FFAHP/GP has three modes the user can choose: the parallel input/output and serial input/output mode, where the flash memory is handled by using the external programmer, and the CPU reprogramming mode, where the flash memory is handled by the central processing unit (CPU). The following explains these modes.

(1) Flash memory mode 1 (parallel I/O mode)

The parallel I/O mode can be selected by connecting wires as shown in Figures 68 and supplying power to the V_{CC} and V_{PP} pins. In this mode, the M38869FFAHP/GP operates as an equivalent of MITSUBISHI's CMOS flash memory M5M28F101. However, because the M38869FFAHP/GP's internal memory has a capacity of 60 Kbytes, programming is available for addresses 01000_{16} to $0FFFF_{16}$, and make sure that the data in addresses 00000_{16} to $00FFF_{16}$ and addresses 10000_{16} to $1FFFF_{16}$ are FF_{16}. Note also that the M38869FFAHP/GP does not contain a facility to read out a device identification code by applying a high voltage to address input (A9). Be careful not to erratically set program conditions when using a general-purpose PROM programmer.

Table 20 shows the pin assignments when operating in the parallel input/output mode.

Table 20 Pin assignments of M38869FFAHP/GP when operating in the parallel input/output mode

	M38869FFAHP/GP	M5M28F101
V_{CC}	V_{CC}	V_{CC}
V_{PP}	CNV$_{SS}$	V_{PP}
V_{SS}	V_{SS}	V_{SS}
Address input	Ports P0, P1, P3$_1$	A_0–A_{16}
Data I/O	Port P2	D_0–D_7
$\overline{CE}$	P3$_6$	$\overline{CE}$
$\overline{OE}$	P3$_7$	$\overline{OE}$
$\overline{WE}$	P3$_3$	$\overline{WE}$

Functional Outline (parallel input/output mode)

In the parallel input/output mode, the M38869FFAHP/GP allow the user to choose an operation mode between the read-only mode and the read/write mode (software command control mode) depending on the voltage applied to the V_{PP} pin. When $V_{PP} = V_{PP}L$, the read-only mode is selected, and the user can choose one of three states (e.g., read, output disable, or standby) depending on inputs to the $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$ pins. When $V_{PP} = V_{PP}H$, the read/write mode is selected, and the user can choose one of four states (e.g., read, output disable, standby, or write) depending on inputs to the $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$ pins. Table 21 shows assignment states of control input and each state.

● Read

The microcomputer enters the read state by driving the $\overline{CE}$, and $\overline{OE}$ pins low and the $\overline{WE}$ pin high; and the contents of memory corresponding to the address to be input to address input pins (A_0–A_{16}) are output to the data input/output pins (D_0–D_7).

● Output disable

The microcomputer enters the output disable state by driving the $\overline{CE}$ pin low and the $\overline{WE}$ and $\overline{OE}$ pins high; and the data input/output pins enter the floating state.

● Standby

The microcomputer enters the standby state by driving the $\overline{CE}$ pin high. The M38869FFAHP/GP is placed in a power-down state consuming only a minimal supply current. At this time, the data input/output pins enter the floating state.

● Write

The microcomputer enters the write state by driving the V_{PP} pin high ($V_{PP} = V_{PP}H$) and then the $\overline{WE}$ pin low when the $\overline{CE}$ pin is low and the $\overline{OE}$ pin is high. In this state, software commands can be input from the data input/output pins, and the user can choose program or erase operation depending on the contents of this software command.

Table 21 Assignment sates of control input and each state

Mode \ Pin / State		$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	V_{PP}	Data I/O
Read-only	Read	V_{IL}	V_{IL}	V_{IH}	$V_{PP}L$	Output
	Output disable	V_{IL}	V_{IH}	V_{IH}	$V_{PP}L$	Floating
	Standby	V_{IH}	×	×	$V_{PP}L$	Floating
Read/Write	Read	V_{IL}	V_{IL}	V_{IH}	$V_{PP}H$	Output
	Output disable	V_{IL}	V_{IH}	V_{IH}	$V_{PP}H$	Floating
	Standby	V_{IH}	×	×	$V_{PP}H$	Floating
	Write	V_{IL}	V_{IH}	V_{IL}	$V_{PP}H$	Input

Note: × can be V_{IL} or V_{IH}.

Table 22 Pin description (flash memory parallel I/O mode)

Pin	Name	Input /Output	Functions
V_{CC}, V_{SS}	Power supply	—	Supply 5 V ± 10 % to V_{CC} and 0 V to V_{SS}.
CNV$_{SS}$	V_{PP} input	Input	Connect to 5 V ± 10 % in read-only mode, connect to 11.7 to 12.6 V in read/write mode.
$\overline{\text{RESET}}$	Reset input	Input	Connect to V_{SS}.
X_{IN}	Clock input	Input	Connect a ceramic resonator between X_{IN} and X_{OUT}.
X_{OUT}	Clock output	Output	
AV$_{SS}$	Analog supply input	—	Connect to V_{SS}.
V_{REF}	Reference voltage input	Input	Connect to V_{SS}.
P0$_0$–P0$_7$	Address input (A$_0$–A$_7$)	Input	Port P0 functions as 8-bit address input (A$_0$–A$_7$).
P1$_0$–P1$_7$	Address input (A$_8$–A$_{15}$)	Input	Port P1 functions as 8-bit address input (A$_8$–A$_{15}$).
P2$_0$–P2$_7$	Data I/O (D$_0$–D$_7$)	I/O	Function as 8-bit data's I/O pins (D$_0$–D$_7$).
P3$_0$–P3$_7$	Control signal input	Input	P3$_7$, P3$_6$ and P3$_3$ function as the $\overline{\text{OE}}$, $\overline{\text{CE}}$ and $\overline{\text{WE}}$ input pins respectively. P3$_1$ functions as the A$_{16}$ input pin. Connect P3$_0$ and P3$_2$ to V_{SS}. Input "H" or "L" to P3$_4$, P3$_5$, or keep them open.
P4$_0$–P4$_7$	Input port P4	Input	Connect P4$_4$, P4$_6$ to V_{SS}. Input "H" or "L" to P4$_0$ - P4$_3$, P4$_5$, P4$_7$, or keep them open.
P5$_0$–P5$_7$	Input port P5	Input	Input "H" or "L", or keep them open.
P6$_0$–P6$_7$	Input port P6	Input	Input "H" or "L", or keep them open.
P7$_0$–P7$_7$	Input port P7	Input	Input "H" or "L", or keep them open.
P8$_0$–P8$_7$	Input port P8	Input	Input "H" or "L", or keep them open.

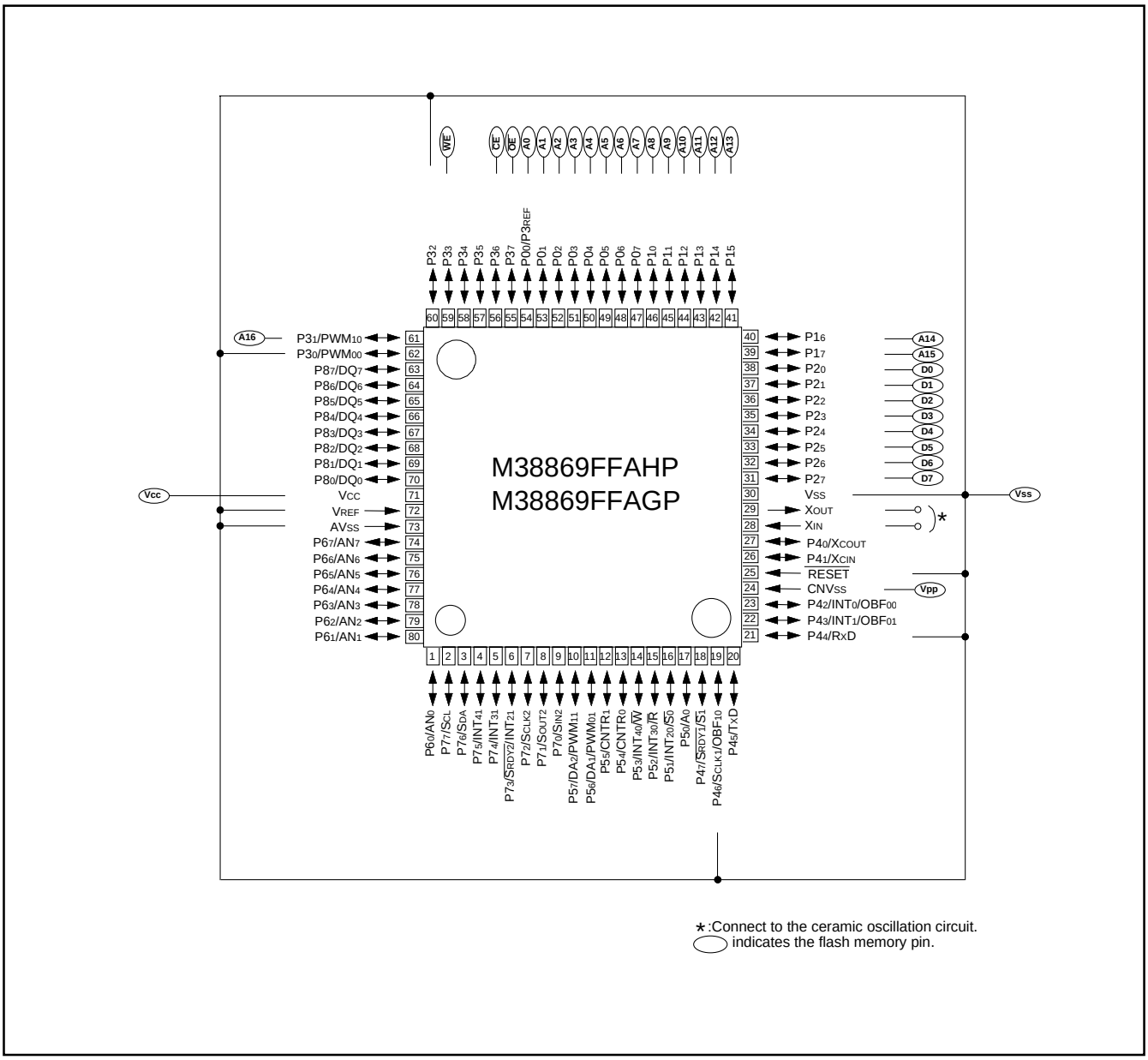


Fig. 68 Pin connection of M38869FFAHP/GP when operating in parallel input/output mode

Read-only Mode

The microcomputer enters the read-only mode by applying $V_{PP}L$ to the V_{PP} pin. In this mode, the user can input the address of a memory location to be read and the control signals at the timing shown in Figure 69, and the M38869FFAHP/GP will output the contents of the user's specified address from data I/O pin to the external. In this mode, the user cannot perform any operation other than read.

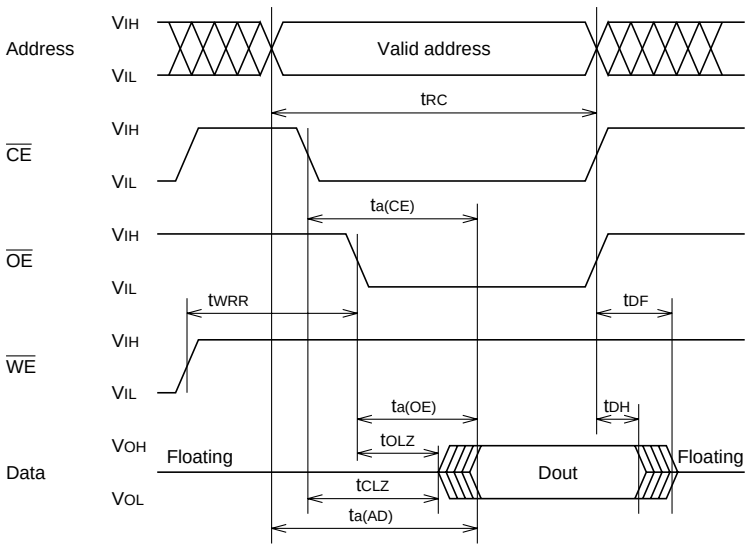


Fig. 69 Read timing

Read/Write Mode

The microcomputer enters the read/write mode by applying $V_{PP}H$ to the V_{PP} pin. In this mode, the user must first input a software command to choose the operation (e. g., read, program, or erase) to be performed on the flash memory (this is called the first cycle), and then input the information necessary for execution of the command (e.g, address and data) and control signals (this is called the second cycle). When this is done, the M38869FFAHP/GP executes the specified operation.

Table 23 shows the software commands and the input/output information in the first and the second cycles. The input address is latched internally at the falling edge of the $\overline{WE}$ input; software commands and other input data are latched internally at the rising edge of the $\overline{WE}$ input.

The following explains each software command. Refer to Figures 70 to 72 for details about the signal input/output timings.

Table 23 Software command (Parallel input/output mode)

Symbol	First cycle		Second cycle	
	Address input	Data input	Address input	Data I/O
Read	×	00_{16}	Read address	Read data (Output)
Program	×	40_{16}	Program address	Program data (Input)
Program verify	×	$C0_{16}$	×	Verify data (Output)
Erase	×	20_{16}	×	20_{16} (Input)
Erase verify	Verify address	$A0_{16}$	×	Verify data (Output)
Reset	×	FF_{16}	×	FF_{16} (Input)
Device identification	×	90_{16}	ADI	DDI (Output)

Note: ADI = Device identification address : manufacturer's code 00000_{16}, device code 00001_{16}
DDI = Device identification data : manufacturer's code $1C_{16}$, device code $D0_{16}$
X can be V_{IL} or V_{IH}.

● **Read command**

The microcomputer enters the read mode by inputting command code "00_{16}" in the first cycle. The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ input. When the address of a memory location to be read is input in the second cycle, with control signals input at the timing shown in Figure 70, the M38869FFAHP/GP outputs the contents of the specified address from the data I/O pins to the external.

The read mode is retained until any other command is latched into the command latch. Consequently, once the M38869FFAHP/GP enters the read mode, the user can read out the successive memory contents simply by changing the input address and executing the second cycle only. Any command other than the read command must be input beginning from its command code over again each time the user execute it. The contents of the command latch immediately after power-on is 00_{16}.

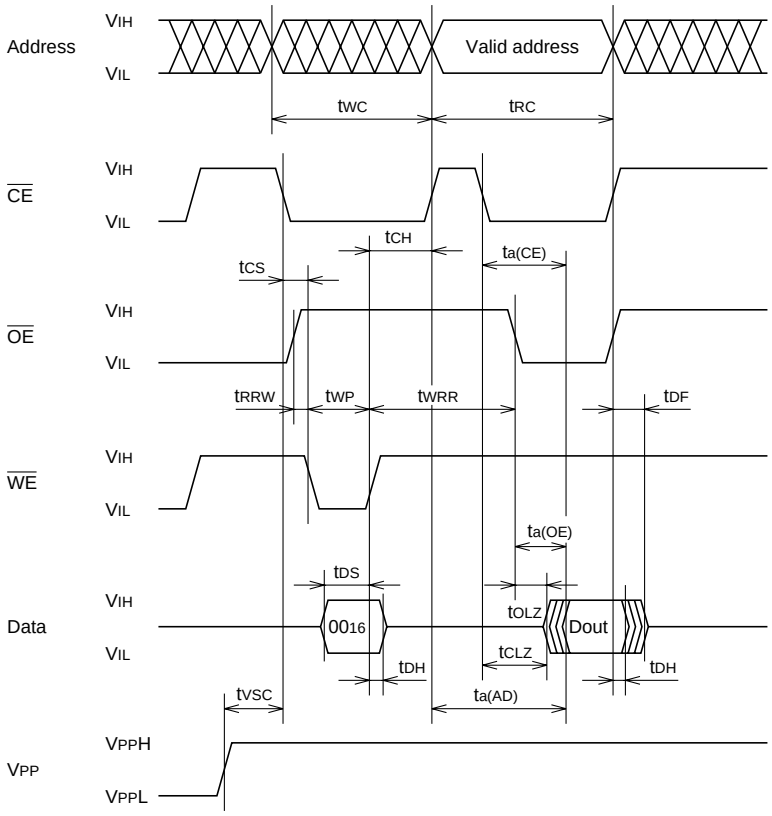


Fig. 70 Timings during reading

● **Program command**

The microcomputer enters the program mode by inputting command code "40$_{16}$" in the first cycle. The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ input. When the address which indicates a program location and data is input in the second cycle, the M38869FFAHP/GP internally latches the address at the falling edge of the $\overline{WE}$ input and the data at the rising edge of the $\overline{WE}$ input. The M38869FFAHP/GP starts programming at the rising edge of the $\overline{WE}$ input in the second cycle and finishes programming within 10 μs as measured by its internal timer. Programming is performed in units of bytes.

Note: A programming operation is not completed by executing the program command once. Always be sure to execute a program verify command after executing the program command. When the failure is found in this verification, the user must repeatedly execute the program command until the pass. Refer to Figure 73 for the programming flowchart.

● **Program verify command**

The microcomputer enters the program verify mode by inputting command code "C0$_{16}$" in the first cycle. This command is used to verify the programmed data after executing the program command. The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ input. When control signals are input in the second cycle at the timing shown in Figure 71, the M38869FFAHP/GP outputs the programmed address's contents to the external. Since the address is internally latched when the program command is executed, there is no need to input it in the second cycle.

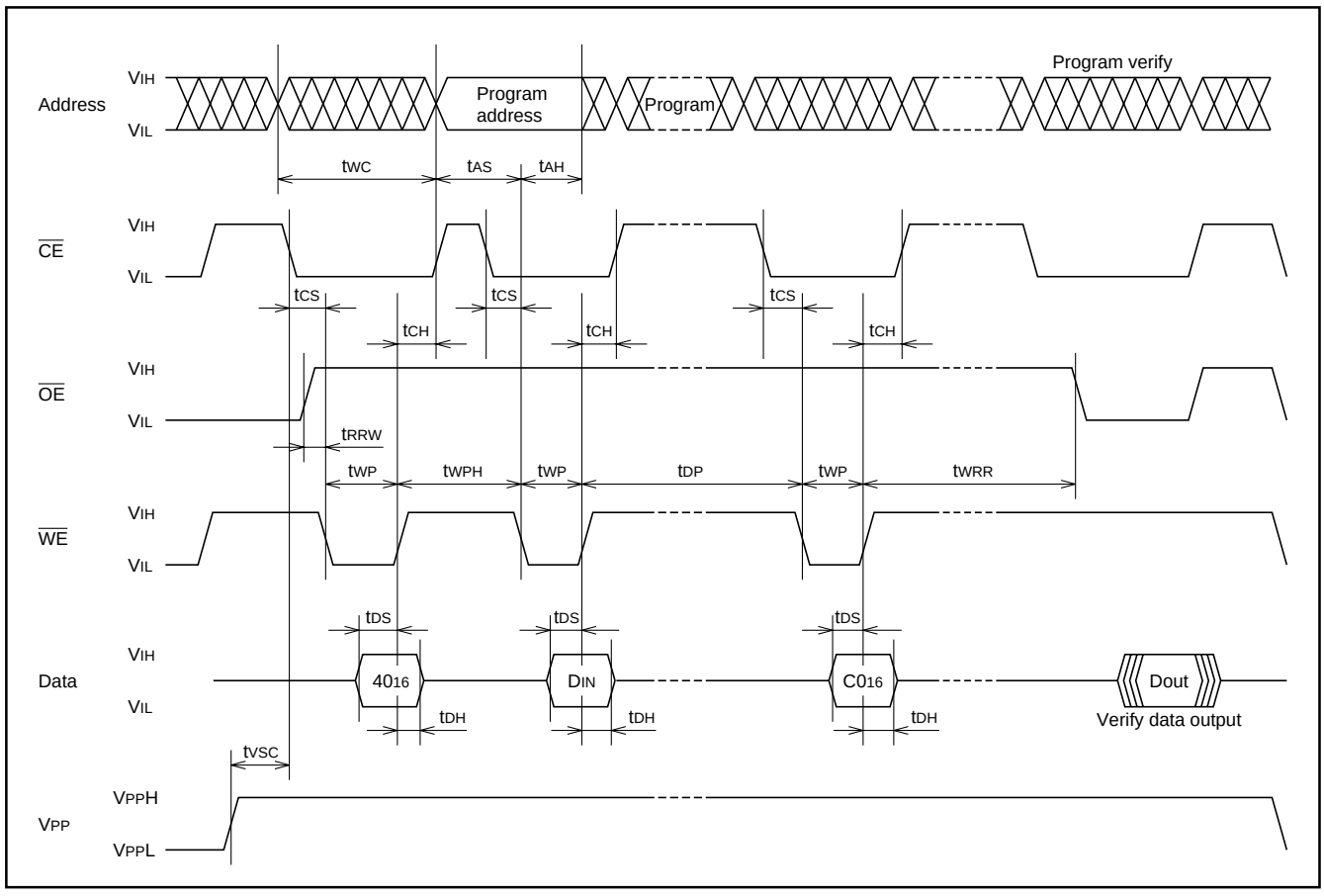


Fig. 71 Input/output timings during programming (Verify data is output at the same timing as for read.)

● **Erase command**

The erase command is executed by inputting command code 20_{16} in the first cycle and command code 20_{16} again in the second cycle. The command code is latched into the internal command latch at the rising edges of the $\overline{WE}$ input in the first cycle and in the second cycle, respectively. The erase operation is initiated at the rising edge of the $\overline{WE}$ input in the second cycle, and the memory contents are collectively erased within 9.5 ms as measured by the internal timer. Note that data 00_{16} must be written to all memory locations before executing the erase command.

Note: An erase operation is not completed by executing the erase command once. Always be sure to execute an erase verify command after executing the erase command. When the failure is found in this verification, the user must repeatedly execute the erase command until the pass. Refer to Figure 73 for the erase flowchart.

● **Erase verify command**

The user must verify the contents of all addresses after completing the erase command. The microcomputer enters the erase verify mode by inputting the verify address and command code $A0_{16}$ in the first cycle. The address is internally latched at the falling edge of the $\overline{WE}$ input, and the command code is internally latched at the rising edge of the $\overline{WE}$ input. When control signals are input in the second cycle at the timing shown in Figure 72, the M38869FFAHP/GP outputs the contents of the specified address to the external.

Note: If any memory location where the contents have not been erased is found in the erase verify operation, execute the operation of "erase → erase verify" over again. In this case, however, the user does not need to write data 00_{16} to memory locations before erasing.

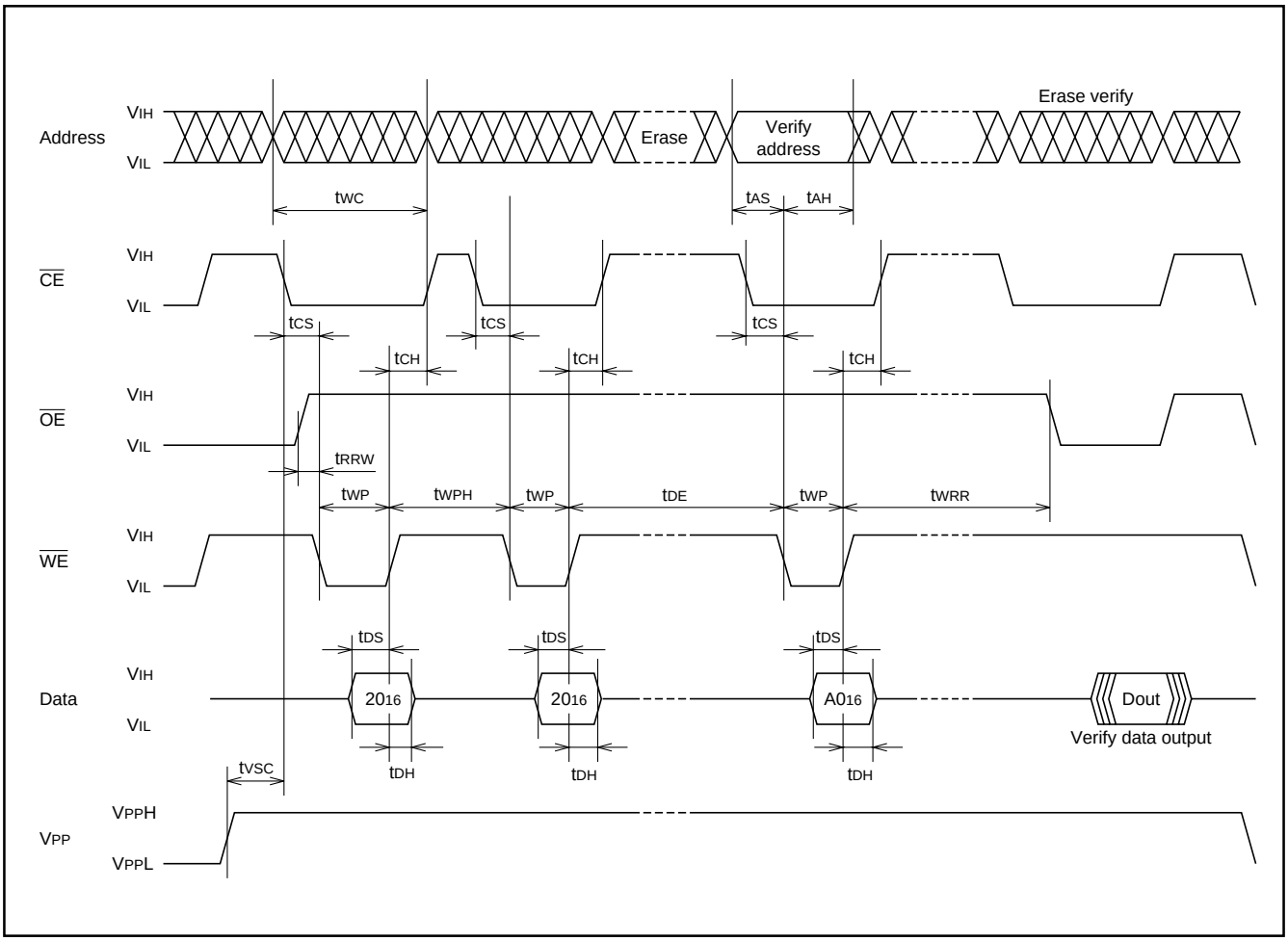


Fig. 72 Input/output timings during erasing (verify data is output at the same timing as for read.)

● **Reset command**

The reset command provides a means of stopping execution of the erase or program command safely. If the user inputs command code FF_{16} in the second cycle after inputting the erase or program command in the first cycle and again input command code FF_{16} in the third cycle, the erase or program command is disabled (i.e., reset), and the M38869FFAHP/GP is placed in the read mode. If the reset command is executed, the contents of the memory does not change.

● **Device identification code command**

By inputting command code 90_{16} in the first cycle, the user can read out the device identification code. The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ input. At this time, the user can read out manufacture's code $1C_{16}$ (i.e., MITSUBISHI) by inputting 0000_{16} to the address input pins in the second cycle; the user can read out device code $D0_{16}$ (i. e., 1M-bit flash memory) by inputting 0001_{16}.

These command and data codes are input/output at the same timing as for read.

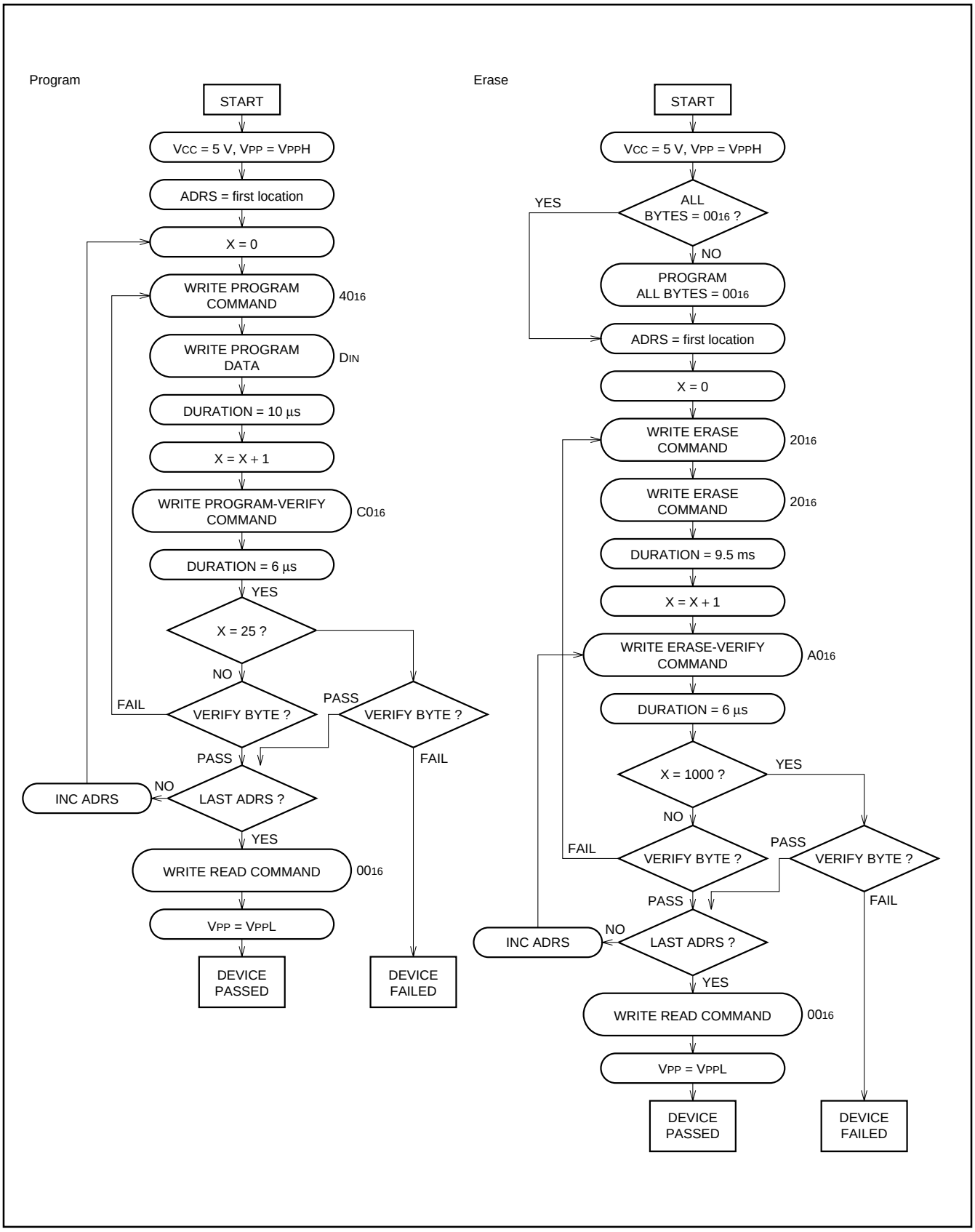


Fig. 73 Programming/Erasing algorithm flow chart

Table 24 DC ELECTRICAL CHARACTERISTICS (T_a = 25 °C, V_{CC} = 5 V ± 10 %, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
I_{SB1}	V_{CC} supply current (at standby)	V_{CC} = 5.5 V, $\overline{CE}$ = V_{IH}			1	mA
I_{SB2}		V_{CC} = 5.5 V, $\overline{CE}$ = V_{CC} ± 0.2 V			100	µA
I_{CC1}	V_{CC} supply current (at read)	V_{CC} = 5.5 V, $\overline{CE}$ = V_{IL}, t_{RC} = 150 ns, I_{OUT} = 0 mA			15	mA
I_{CC2}	V_{CC} supply current (at program)	V_{PP} = $V_{PP}H$			15	mA
I_{CC3}	V_{CC} supply current (at erase)	V_{PP} = $V_{PP}H$			15	mA
I_{PP1}	V_{PP} supply current (at read)	0≤V_{PP}≤V_{CC}			10	µA
		V_{CC}<V_{PP}≤V_{CC} + 1.0 V			100	µA
		V_{PP} = $V_{PP}H$			100	µA
I_{PP2}	V_{PP} supply current (at program)	V_{PP} = $V_{PP}H$			30	mA
I_{PP3}	V_{PP} supply current (at erase)	V_{PP} = $V_{PP}H$			30	mA
V_{IL}	"L" input voltage		0		0.8	V
V_{IH}	"H" input voltage		2.0		V_{CC}	V
V_{OL}	"L" output voltage	I_{OL} = 2.1 mA			0.45	V
V_{OH1}	"H" output voltage	I_{OH} = –400 µA	2.4			V
V_{OH2}		I_{OH} = –100 µA	V_{CC} –0.4			V
$V_{PP}L$	V_{PP} supply voltage (read only)		V_{CC}		V_{CC} + 1.0	V
$V_{PP}H$	V_{PP} supply voltage (read/write)		11.7	12.0	12.6	V

AC ELECTRICAL CHARACTERISTICS (T_a = 25 °C, V_{CC} = 5 V ± 10 %, unless otherwise noted)

Table 25 Read-only mode

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_{RC}	Read cycle time	250		ns
$t_{a(AD)}$	Address access time		250	ns
$t_{a(CE)}$	$\overline{CE}$ access time		250	ns
$t_{a(OE)}$	$\overline{OE}$ access time		100	ns
t_{CLZ}	Output enable time (after $\overline{CE}$)	0		ns
t_{OLZ}	Output enable time (after $\overline{OE}$)	0		ns
t_{DF}	Output floating time (after $\overline{OE}$)		35	ns
t_{DH}	Output valid time (after $\overline{CE}$, $\overline{OE}$, address)	0		ns
t_{WRR}	Write recovery time (before read)	6		µs

Table 26 Read/Write mode

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_{WC}	Write cycle time	150		ns
t_{AS}	Address set up time	0		ns
t_{AH}	Address hold time	60		ns
t_{DS}	Data setup time	50		ns
t_{DH}	Data hold time	10		ns
t_{WRR}	Write recovery time (before read)	6		µs
t_{RRW}	Read recovery time (before write)	0		µs
t_{CS}	$\overline{CE}$ setup time	20		ns
t_{CH}	$\overline{CE}$ hold time	0		ns
t_{WP}	Write pulse width	60		ns
t_{WPH}	Write pulse waiting time	20		ns
t_{DP}	Program time	10		µs
t_{DE}	Erase time	9.5		ms
t_{VSC}	V_{PP} setup time	1		µs

Note: Read timing of Read/Write mode is same as Read-only mode.

(2) Flash memory mode 2 (serial I/O mode)

The M38869FFAHP/GP has a function to serially input/output the software commands, addresses, and data required for operation on the internal flash memory (e. g., read, program, and erase) using only a few pins. This is called the serial I/O (input/output) mode. This mode can be selected by driving the SDA (serial data input/output), SCLK (serial clock input), and $\overline{OE}$ pins high after connecting wires as shown in Figures 74 and powering on the V_{CC} pin and then applying $V_{PP}H$ to the V_{PP} pin.

In the serial I/O mode, the user can use six types of software commands: read, program, program verify, erase, erase verify and error check.

Serial input/output is accomplished synchronously with the clock, beginning from the LSB (LSB first).

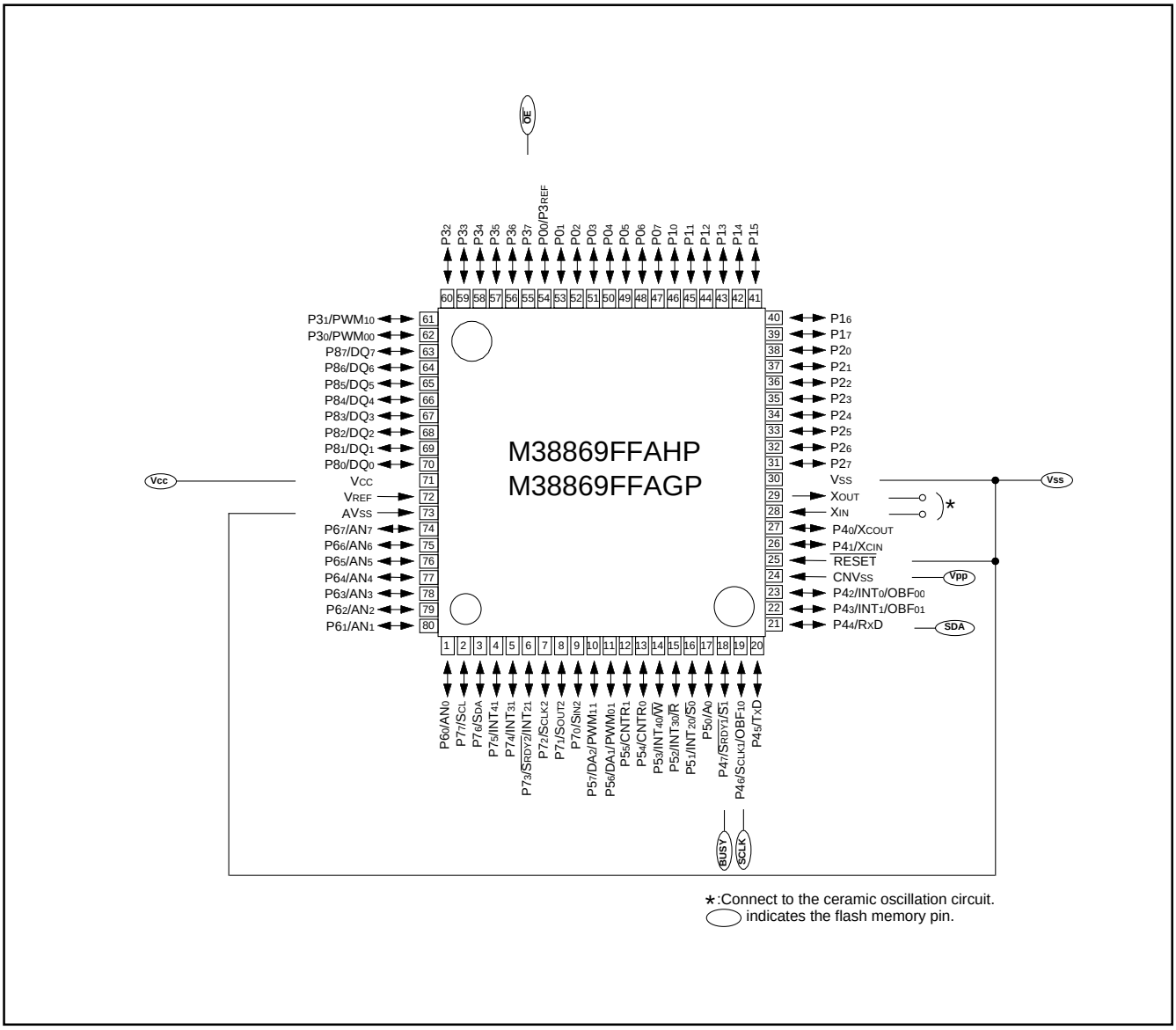


Fig. 74 Pin connection of M38869FFAHP/GP when operating in serial I/O mode

Table 27 Pin description (flash memory serial I/O mode)

Pin	Name	Input /Output	Functions
V_{CC}, V_{SS}	Power supply	—	Supply 5 V ± 10 % to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	V_{PP} input	Input	Connect to 11.7 to 12.6 V.
$\overline{RESET}$	Reset input	Input	Connect to V_{SS}.
X_{IN}	Clock input	Input	Connect a ceramic resonator between X_{IN} and X_{OUT}.
X_{OUT}	Clock output	Output	
AV_{SS}	Analog supply input	—	Connect to V_{SS}.
V_{REF}	Reference voltage input	Input	Input an arbitrary level between the range of V_{SS} and V_{CC}.
$P0_0$–$P0_7$	Input port P0	Input	Input “H” or “L”, or keep them open.
$P1_0$–$P1_7$	Input port P1	Input	Input “H” or “L”, or keep them open.
$P2_0$–$P2_7$	Input port P2	Input	Input “H” or “L”, or keep them open.
$P3_0$–$P3_6$	Input port P3	Input	Input “H” or “L”, or keep them open.
$P3_7$	Control signal input	Input	$\overline{OE}$ input pin
$P4_0$–$P4_3$, $P4_5$	Input port P4	Input	Input “H” or “L” to $P4_0$ - $P4_3$, $P4_5$, or keep them open.
$P4_4$	SDA I/O	I/O	This pin is for serial data I/O.
$P4_6$	SCLK input	Input	This pin is for serial clock input.
$P4_7$	BUSY output	Output	This pin is for BUSY signal output.
$P5_0$–$P5_7$	Input port P5	Input	Input “H” or “L”, or keep them open.
$P6_0$–$P6_7$	Input port P6	Input	Input “H” or “L”, or keep them open.
$P7_0$–$P7_7$	Input port P7	Input	Input “H” or “L”, or keep them open.
$P8_0$–$P8_7$	Input port P8	Input	Input “H” or “L”, or keep them open.

Functional Outline (serial I/O mode)

In the serial I/O mode, data is transferred synchronously with the clock using serial input/output. The input data is read from the SDA pin into the internal circuit synchronously with the rising edge of the serial clock pulse; the output data is output from the SDA pin synchronously with the falling edge of the serial clock pulse. Data is transferred in units of eight bits.

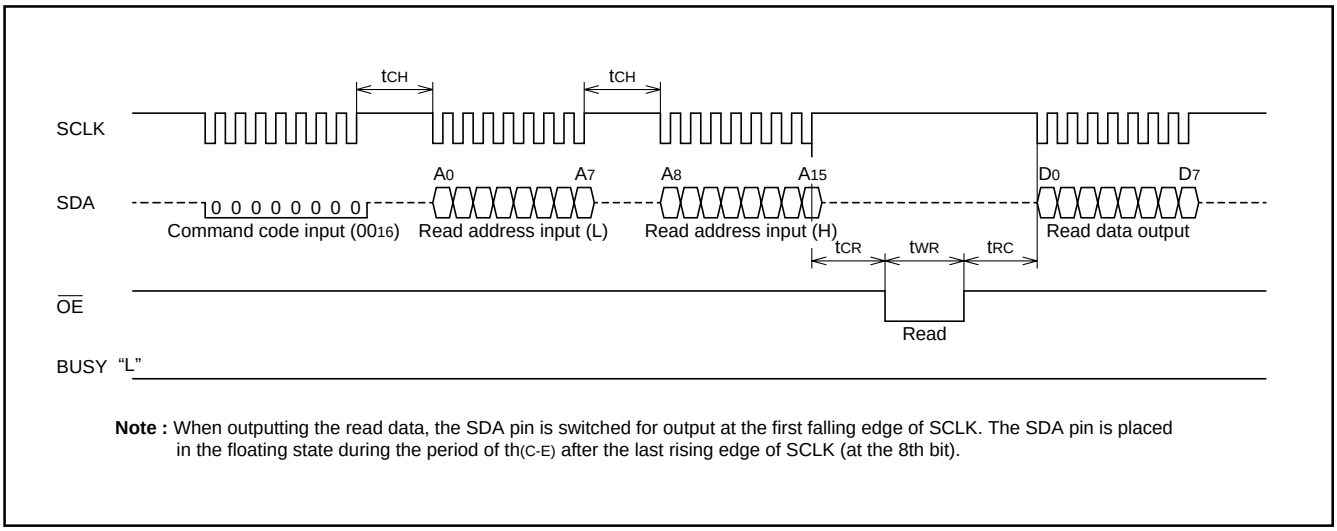
In the first transfer, the user inputs the command code. This is followed by address input and data input/output according to the contents of the command. Table 28 shows the software commands used in the serial I/O mode. The following explains each software command.

Table 28 Software command (serial I/O mode)

Number of transfers / Command	First command code input	Second	Third	Fourth
Read	00_{16}	Read address L (Input)	Read address H (Input)	Read data (Output)
Program	40_{16}	Program address L (Input)	Program address H (Input)	Program data (Input)
Program verify	$C0_{16}$	Verify data (Output)	————	————
Erase	20_{16}	20_{16} (Input)	————	————
Erase verify	$A0_{16}$	Verify address L (Input)	Verify address H (Input)	Verify data (Output)
Error check	80_{16}	Error code (Output)	————	————

● **Read command**

Input command code 00_{16} in the first transfer. Proceed and input the low-order 8 bits and the high-order 8 bits of the address and pull the $\overline{OE}$ pin low. When this is done, the M38869FFAHP/GP reads out the contents of the specified address, and then latches it into the internal data latch. When the $\overline{OE}$ pin is released back high and serial clock is input to the SCLK pin, the read data that has been latched into the data latch is serially output from the SDA pin.



Note : When outputting the read data, the SDA pin is switched for output at the first falling edge of SCLK. The SDA pin is placed in the floating state during the period of $th_{(C-E)}$ after the last rising edge of SCLK (at the 8th bit).

Fig. 75 Timings during reading

● **Program command**

Input command code 40_{16} in the first transfer. Proceed and input the low-order 8 bits and the high-order 8 bits of the address and then program data. Programming is initiated at the last rising edge of the serial clock during program data transfer. The BUSY pin is driven high during program operation. Programming is completed within 10 μs as measured by the internal timer, and the BUSY pin is pulled low.

Note : A programming operation is not completed by executing the program command once. Always be sure to execute a program verify command after executing the program command. When the failure is found in the verification, the user must repeatedly execute the program command until the pass in the verification. Refer to Figure 73 for the programming flowchart.

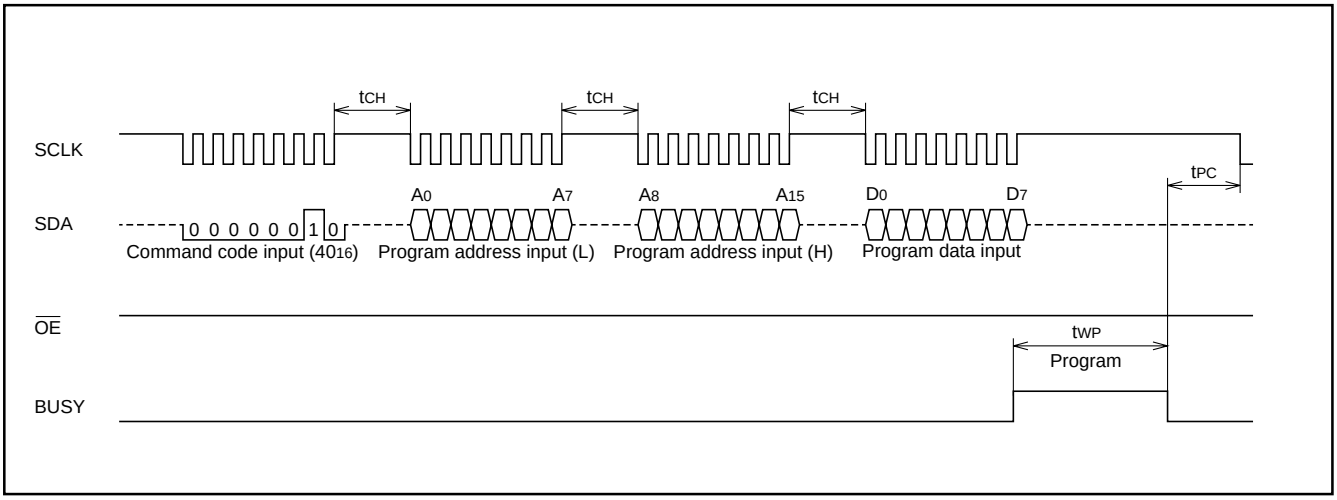
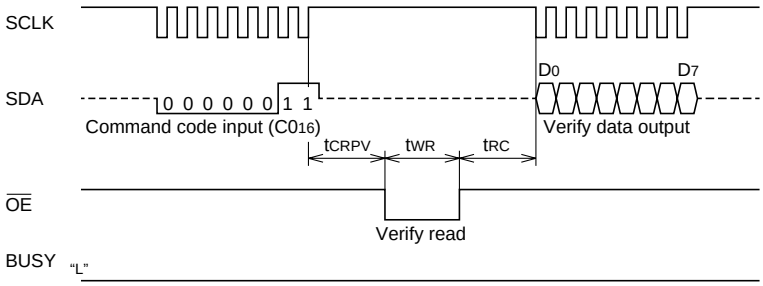


Fig. 76 Timings during programming

● **Program verify command**

Input command code $C0_{16}$ in the first transfer. Proceed and drive the $\overline{OE}$ pin low. When this is done, the M38869FFAHP/GP verify-reads the programmed address's contents, and then latches it into the internal data latch. When the $\overline{OE}$ pin is released back high and serial clock is input to the SCLK pin, the verify data that has been latched into the data latch is serially output from the SDA pin.



Note: When outputting the verify data, the SDA pin is switched for output at the first falling edge of SCLK. The SDA pin is placed in the floating state during the period of $th_{(C-E)}$ after the last rising edge of SCLK (at the 8th bit).

Fig. 77 Timings during program verify

● **Erase command**

Input command code 20_{16} in the first transfer and command code 20_{16} again in the second transfer. When this is done, the M38869FFAHP/GP executes an erase command. Erase is initiated at the last rising edge of the serial clock. The BUSY pin is driven high during the erase operation. Erase is completed within 9.5 ms as measured by the internal timer, and the BUSY pin is pulled low. Note that data 00_{16} must be written to all memory locations before executing the erase command.

Note: A erase operation is not completed by executing the erase command once. Always be sure to execute a erase verify command after executing the erase command. When the failure is found in the verification, the user must repeatedly execute the erase command until the pass in the verification. Refer to Figure 73 for the erase flowchart.

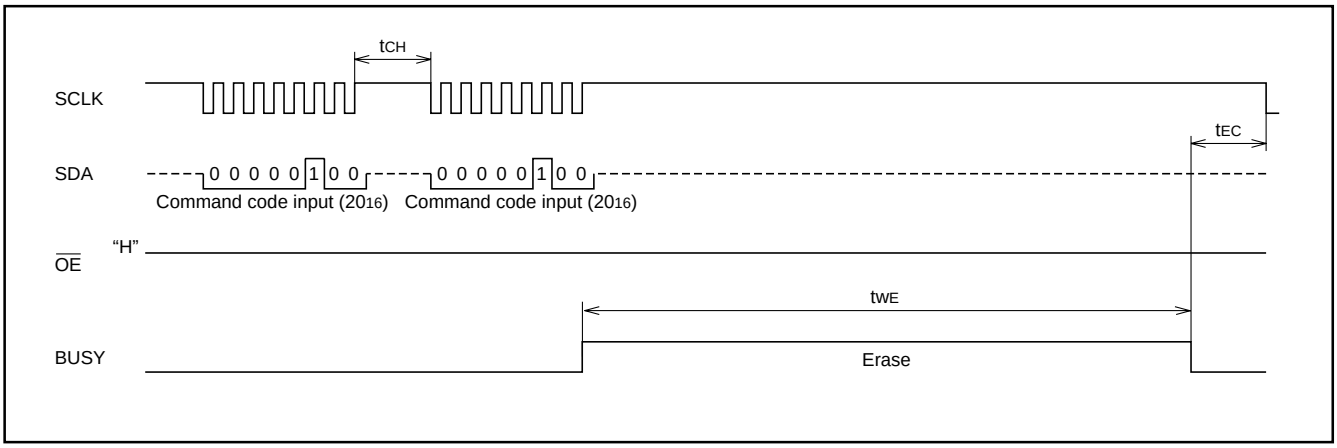


Fig. 78 Timings at erasing

● **Erase verify command**

The user must verify the contents of all addresses after completing the erase command. Input command code $A0_{16}$ in the first transfer. Proceed and input the low-order 8 bits and the high-order 8 bits of the address and pull the $\overline{OE}$ pin low. When this is done, the M38869FFAHP/GP reads out the contents of the specified address, and then latches it into the internal data latch. When the $\overline{OE}$ pin is released back high and serial clock is input to the SCLK pin, the verify data that has been latched into the data latch is serially output from the SDA pin.

Note: If any memory location where the contents have not been erased is found in the erase verify operation, execute the operation of "erase → erase verify" over again. In this case, however, the user does not need to write data 00_{16} to memory locations before erasing.

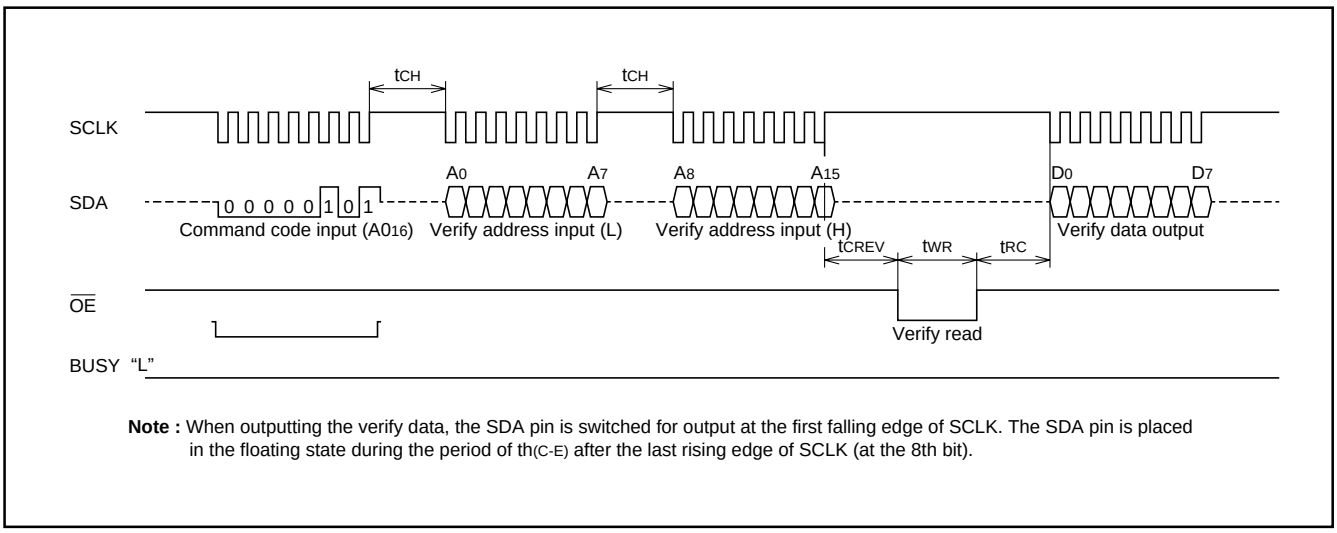
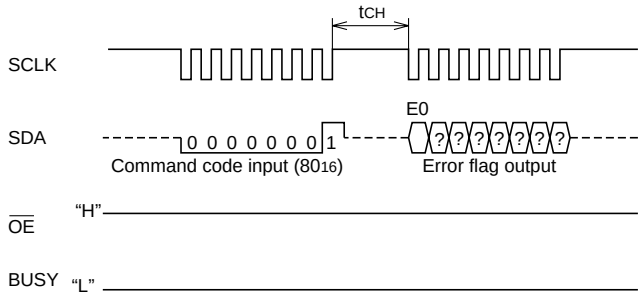


Fig. 79 Timings during erase verify

● **Error check command**

Input command code 80_{16} in the first transfer, and the M38869FFAHP/GP outputs error information from the SDA pin, beginning at the next falling edge of the serial clock. If the LSB bit of the 8-bit error information is 1, it indicates that a command error has occurred. A command error means that some invalid commands other than commands shown in Table 28 has been input.

When a command error occurs, the serial communication circuit sets the corresponding flag and stops functioning to avoid an erroneous programming or erase. When being placed in this state, the serial communication circuit does not accept the subsequent serial clock and data (even including an error check command). Therefore, if the user wants to execute an error check command, temporarily drop the V_{PP} pin input to the $V_{PP}L$ level to terminate the serial input/output mode. Then, place the M38869FFAHP/GP into the serial I/O mode back again. The serial communication circuit is reset by this operation and is ready to accept commands. The error flag alone is not cleared by this operation, so the user can examine the serial communication circuit's error conditions before reset. This examination is done by the first execution of an error check command after the reset. The error flag is cleared when the user has executed the error check command. Because the error flag is undefined immediately after power-on, always be sure to execute the error check command.



Note: When outputting the error flag, the SDA pin is switched for output at the first falling edge of the serial clock. The SDA pin is placed in the floating state during the period of $t_{h(C-E)}$ after the last rising edge of the serial clock (at the 8th bit).

Fig. 80 Timings at error checking

Note: The programming/erasing algorithm flow chart of the serial I/O mode is the same as that of the parallel I/O mode. Refer to Figure 73.

DC ELECTRICAL CHARACTERISTICS (Ta = 25 °C, VCC = 5 V ± 10 %, VPP = 11.7 to 12.6 V, unless otherwise noted)

ICC, IPP-relevant standards during read, program, and erase are the same as in the parallel input/output mode. VIH, VIL, VOH, VOL, IIH, and IIL for the SCLK, SDA, BUSY, $\overline{OE}$ pins conform to the microcomputer modes.

Table 29 AC Electrical characteristics
(Ta = 25 °C, VCC = 5 V ± 10 %, VPP = 11.7 to 12.6 V, f(XIN) = 10 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
tCH	Serial transmission interval	500(Note 1)		ns
tCR	Read waiting time after transmission	500(Note 1)		ns
tWR	Read pulse width	400(Note 2)		ns
tRC	Transfer waiting time after read	500(Note 1)		ns
tCRPV	Waiting time before program verify	6		µs
tWP	Programming time		10	µs
tPC	Transfer waiting time after programming	500(Note 1)		ns
tCREV	Waiting time before erase verify	6		ns
tWE	Erase time		9.5	ns
tEC	Transfer waiting time after erase	500(Note 1)		ns
tc(CK)	SCLK input cycle time	250		ns
tw(CKH)	SCLK high-level pulse width	100		ns
tw(CKL)	SCLK low-level pulse width	100		ns
tr(CK)	SCLK rise time	20		ns
tf(CK)	SCLK fall time	20		ns
td(C-Q)	SDA output delay time	0	90	ns
th(C-Q)	SDA output hold time	0		ns
th(C-E)	SDA output hold time (only the 8th bit)	150(Note 3)	250(Note 4)	ns
tsu(D-C)	SDA input set up time	30		ns
th(C-D)	SDA input hold time	90		ns

Notes 1: When f(XIN) = 10 MHz or less, calculate the minimum value according to formula 1.

$$\text{Formula 1} : \frac{5000}{f(X_{IN})} \times 10^6$$

2: When f(XIN) = 10 MHz or less, calculate the minimum value according to formula 2.

$$\text{Formula 2} : \frac{4000}{f(X_{IN})} \times 10^6$$

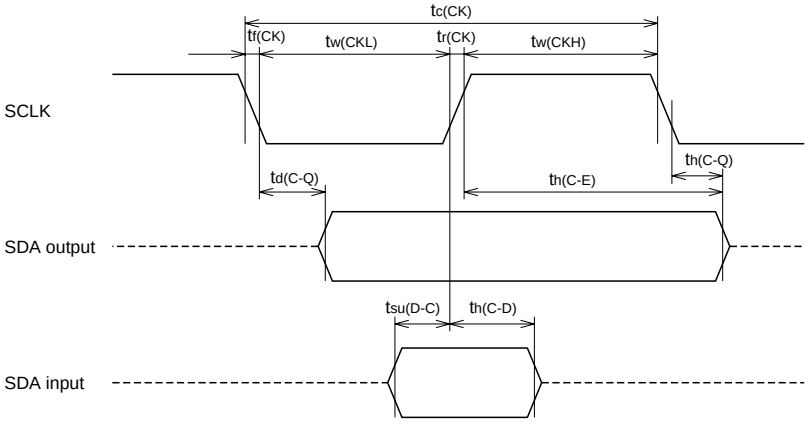
3: When f(XIN) = 10 MHz or less, calculate the minimum value according to formula 3.

$$\text{Formula 3} : \frac{1500}{f(X_{IN})} \times 10^6$$

4: When f(XIN) = 10 MHz or less, calculate the minimum value according to formula 4

$$\text{Formula 4} : \frac{2500}{f(X_{IN})} \times 10^6$$

AC waveforms



Test conditions for AC characteristics

- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V
- Input timing voltage : VIL = 0.2 VCC, VIH = 0.8 VCC

(3) Flash memory mode 3 (CPU reprogramming mode)

The M38869FFAHP/GP has the CPU reprogramming mode where a built-in flash memory is handled by the central processing unit (CPU).

In CPU reprogramming mode, the flash memory is handled by writing and reading to/from the flash memory control register (see Figure 81) and the flash command register (see Figure 82).

The CNVSS pin is used as the VPP power supply pin in CPU reprogramming mode. It is necessary to apply the power-supply voltage of VPPH from the external to this pin.

Functional Outline (CPU reprogramming mode)

Figure 81 shows the flash memory control register bit configuration. Figure 82 shows the flash command register bit configuration.

Bit 0 of the flash memory control register is the CPU reprogramming mode select bit. When this bit is set to "1" and VPPH is applied to the CNVss/VPP pin, the CPU reprogramming mode is selected. Whether the CPU reprogramming mode is realized or not is judged by reading the CPU reprogramming mode monitor flag (bit 2 of the flash memory control register).

Bit 1 is a busy flag which becomes "1" during erase and program execution.

Whether each operation has been completed or not is judged by checking this flag after execution of each erase or program command.

Bits 4, 5 of the flash memory control register are the erase/program area select bits. These bits specify an area where erase and program is operated. When the erase command is executed after an area is specified by these bits, only the specified area is erased. Programming is enabled only for the specified area: programming is disabled for all other areas.

When CPU reprogramming mode is valid, the area not specified by the erase/program area select bits cannot be read out.

Transfer the CPU reprogramming mode control program to internal RAM before entering the CPU reprogramming mode, and then execute this program on internal RAM.

If an interrupt occurs while this program is being executed, the flash memory area is accessed, but normal operations cannot be performed because the flash memory area cannot be read out. Execute processes such as interrupt disable during the CPU reprogramming mode control program.

Figure 83 shows the CPU mode register bit configuration in the CPU reprogramming mode. Set bits 1 and 0 to "00" (single-chip mode) in the CPU reprogramming mode.

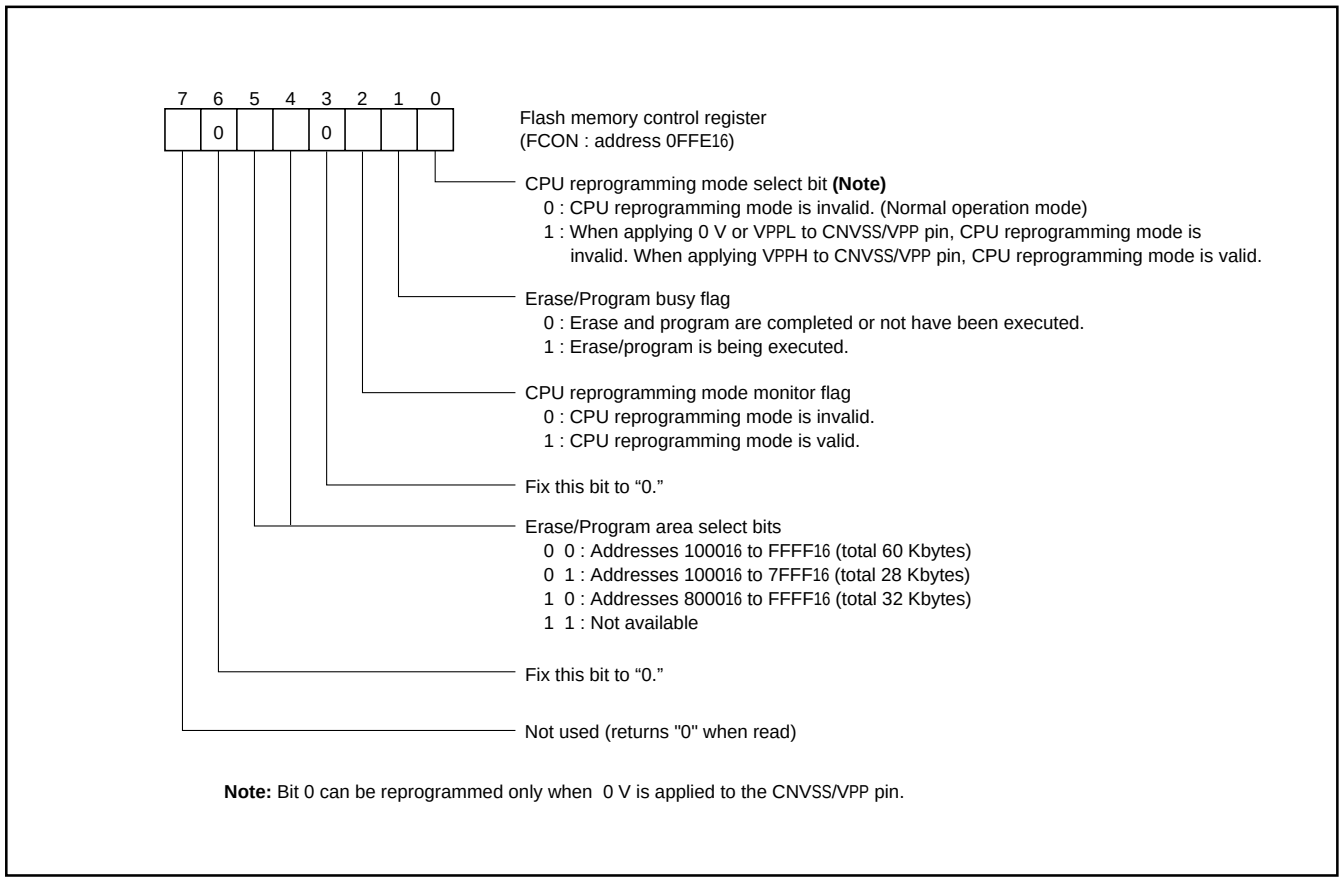


Fig. 81 Flash memory control register bit configuration

● CPU reprogramming mode operation procedure

The operation procedure in CPU reprogramming mode is described below.

< Beginning procedure >

① Apply 0 V to the CNVss/VPP pin for reset release.
② After CPU reprogramming mode control program is transferred to internal RAM, jump to this control program on RAM. (The following operations are controlled by this control program).
③ Set "1" to the CPU reprogramming mode select bit.
④ Apply VPPH to the CNVSS/VPP pin.
⑤ Wait till CNVSS/VPP pin becomes 12 V.
⑥ Read the CPU reprogramming mode monitor flag to confirm whether the CPU reprogramming mode is valid.
⑦ The operation of the flash memory is executed by software-command-writing to the flash command register .

Note: The following are necessary other than this:
- •Control for data which is input from the external (serial I/O etc.) and to be programmed to the flash memory
- •Initial setting for ports etc.
- •Writing to the watchdog timer

< Release procedure >

① Apply 0V to the CNVSS/VPP pin.
② Wait till CNVSS/VPP pin becomes 0V.
③ Set the CPU reprogramming mode select bit to "0."

Each software command is explained as follows.

● Read command

When "00_{16}" is written to the flash command register, the M38869FFAHP/GP enters the read mode. The contents of the corresponding address can be read by reading the flash memory (For instance, with the LDA instruction etc.) under this condition.
The read mode is maintained until another command code is written to the flash command register. Accordingly, after setting the read mode once, the contents of the flash memory can continuously be read.
After reset and after the reset command is executed, the read mode is set.

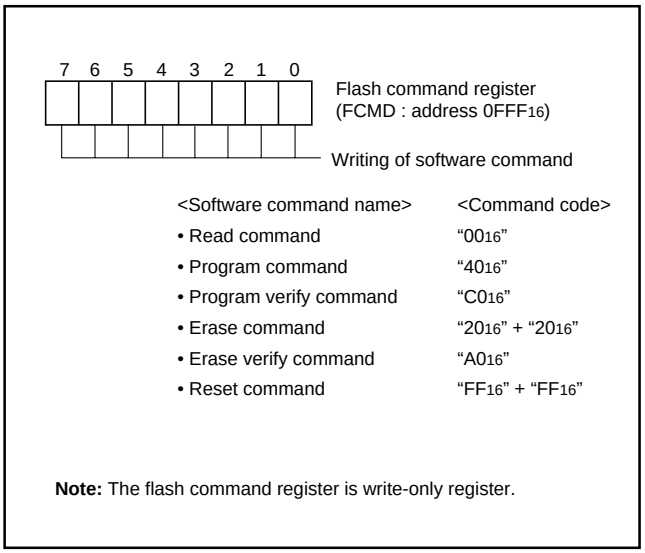


Fig. 82 Flash command register bit configuration

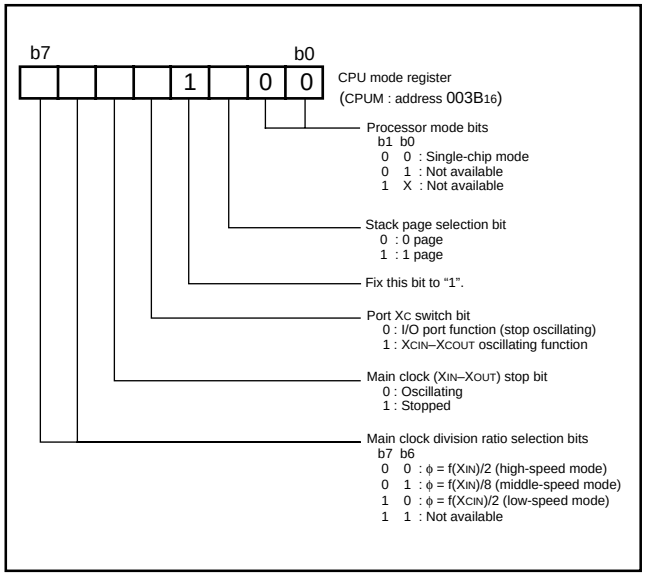


Fig. 83 CPU mode register bit configuration in CPU rewriting mode

● **Program command**

When "40_{16}" is written to the flash command register, the M38869FFAHP/GP enters the program mode.

Subsequently to this, if the instruction (for instance, STA instruction) for writing byte data in the address to be programmed is executed, the control circuit of the flash memory executes the program. The erase/program busy flag of the flash memory control register is set to "1" when the program starts, and becomes "0" when the program is completed. Accordingly, after the write instruction is executed, CPU can recognize the completion of the program by polling this bit.

The programmed area must be specified beforehand by the erase/program area select bits.

During programming, watchdog timer stops with "$FFFF_{16}$" set.

Note: A programming operation is not completed by executing the program command once. Always be sure to execute a program verify command after executing the program command. When the failure is found in this verification, the user must repeatedly execute the program command until the pass. Refer to Figure 84 for the flow chart of the programming.

● **Program verify command**

When "$C0_{16}$" is written to the flash command register, the M38869FFAHP/GP enters the program verify mode. Subsequently to this, if the instruction (for instance, LDA instruction) for reading byte data from the address to be verified (i.e., previously programmed address), the contents which has been written to the address actually is read.

CPU compares this read data with data which has been written by the previous program command. In consequence of the comparison, if not agreeing, the operation of "program → program verify" must be executed again.

● **Erase command**

When writing "20_{16}" twice continuously to the flash command register, the flash memory control circuit performs erase to the area specified beforehand by the erase/program area select bits.

Erase/program busy flag of the flash memory control register becomes "1" when erase begins, and it becomes "0" when erase completes. Accordingly, CPU can recognize the completion of erase by polling this bit.

Data "00_{16}" must be written to all areas to be erased by the program and the program verify commands before the erase command is executed.

During erasing, watchdog timer stops with "$FFFF_{16}$" set.

Note: The erasing operation is not completed by executing the erase command once. Always be sure to execute an erase verify command after executing the erase command. When the failure is found in this verification, the user must repeatedly execute the erase command until the pass. Refer to Figure 84 for the erasing flowchart.

● **Erase verify command**

When "$A0_{16}$" is written to the flash command register, the M38869FFAHP/GP enters the erase verify mode. Subsequently to this, if the instruction (for instance, LDA instruction) for reading byte data from the address to be verified, the contents of the address is read.

CPU must erase and verify to all erased areas in a unit of address.

If the address of which data is not "FF_{16}" (i.e., data is not erased) is found, it is necessary to discontinue erasure verification there, and execute the operation of "erase → erase verify" again.

Note: By executing the operation of "erase → erase verify" again when the memory not erased is found. It is unnecessary to write data "00_{16}" before erasing in this case.

● **Reset command**

The reset command is a command to discontinue the program or erase command on the way. When "FF_{16}" is written to the command register two times continuously after "40_{16}" or "20_{16}" is written to the flash command register, the program, or erase command becomes invalid (reset), and the M38869FFAHP/GP enters the reset mode.

The contents of the memory does not change even if the reset command is executed.

DC Electric Characteristics

Note: The characteristic concerning the flash memory part are the same as the characteristic of the parallel I/O mode.

AC Electric Characteristics

Note: The characteristics are the same as the characteristic of the microcomputer mode.

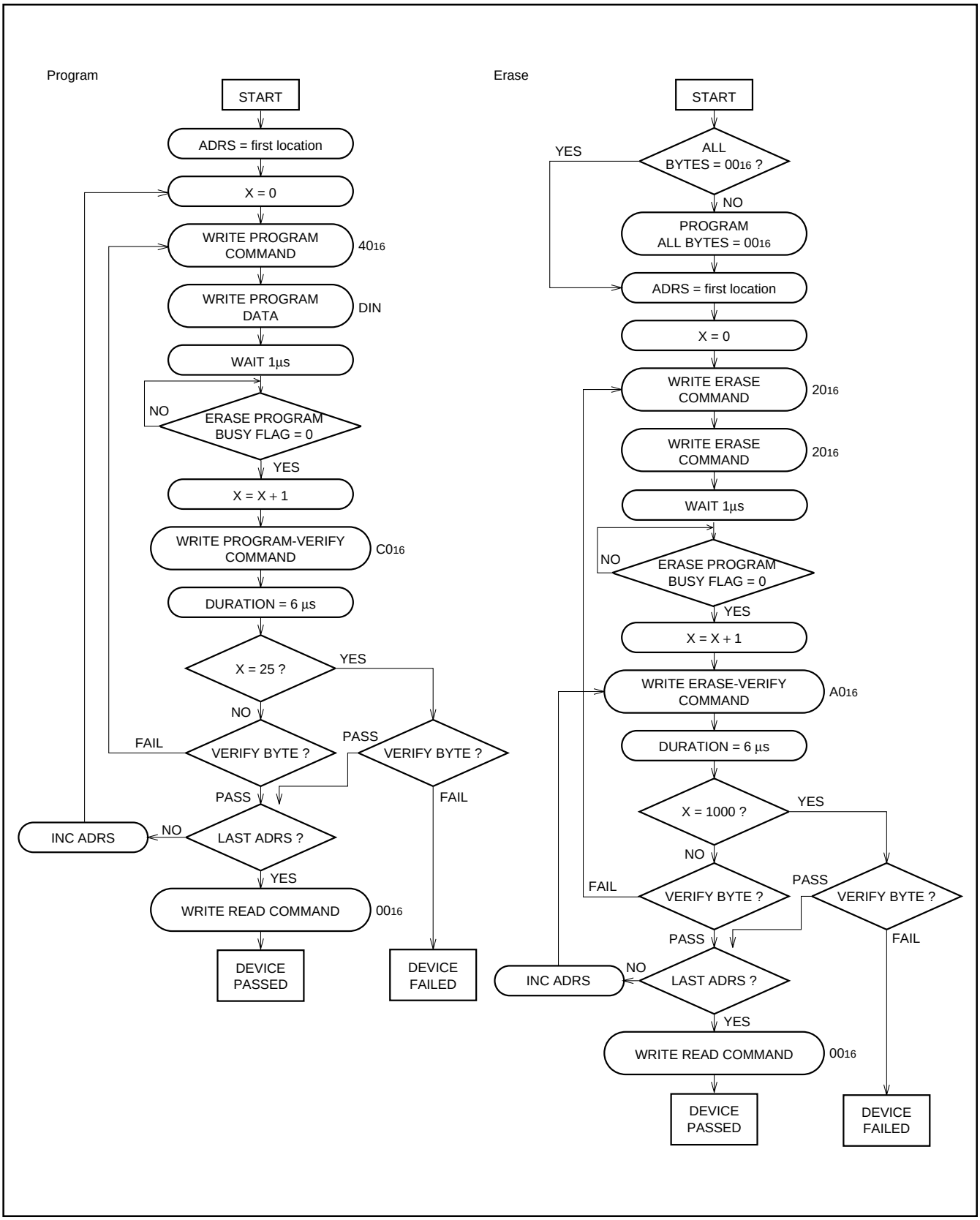


Fig. 84 Flowchart of program/erase operation at CPU reprogramming mode

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 instruction with the addressing mode which uses the value of a direction register as an index
- The bit-test instruction (BBC or BBS, etc.) to a direction register
- The read-modify-write instructions (ROR, CLB, or SEB, etc.) to a direction register.

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

Serial I/O

In clock synchronous serial I/O, if the receive side is using an external clock and it is to output the $\overline{\text{SRDY1}}$ signal, set the transmit enable bit, the receive enable bit, and the $\overline{\text{SRDY1}}$ output enable bit to "1."

Serial I/O1 continues to output the final bit from the TxD pin after transmission is completed. SOUT2 pin for serial I/O2 goes to high impedance after transfer is completed.

When in serial I/O1 (clock-synchronous mode) or in serial I/O2, an external clock is used as synchronous clock, write transmission data to the transmit buffer register or serial I/O2 register, during transfer clock is "H."

A-D Converter

The comparator uses capacitive coupling amplifier whose charge will be lost if the clock frequency is too low.

Therefore, make sure that f(XIN) is at least on 500 kHz during an A-D conversion.

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

D-A Converter

The accuracy of the D-A converter becomes rapidly poor under the V_{CC} = 4.0 V or less condition; a supply voltage of $V_{CC} \geq 4.0$ V is recommended. When a D-A converter is not used, set all values of D-Ai conversion registers (i=1, 2) to "00_{16}."

Instruction Execution Time

The instruction execution time is obtained by multiplying the period of the internal clock ϕ by the number of cycles needed to execute an instruction.

The number of cycles required to execute an instruction is shown in the list of machine instructions.

The period of the internal clock ϕ is half of the XIN period in high-speed mode.

When the $\overline{\text{ONW}}$ function is used in modes other than single-chip mode, the period of the internal clock ϕ may be four times that of the XIN.

NOTES ON USAGE

Handling of Power Source Pins

In order to avoid a latch-up occurrence, connect a capacitor suitable for high frequencies as bypass capacitor between power source pin (V_{CC} pin) and GND pin (V_{SS} pin), between power source pin (V_{CC} pin) and analog power source input pin (AV_{SS} pin). Connect the same kind of capacitor between program power source pin (CNVss/VPP) and GND pin when executing on-board reprogramming of flash memory version. Make sure the connection between each pin is as short as possible. We recommend using a ceramic capacitor of 0.01 μF to 0.1 μF.

EPROM version/One Time PROM version/ Flash memory version

The CNV_{SS} pin is connected to the internal memory circuit block by a low-ohmic resistance, since it has the multiplexed function to be a programmable power source pin (V_{PP} pin) as well.
To improve the noise reduction, connect a track between CNV_{SS} pin and V_{SS} pin or V_{CC} pin with 1 to 10 kΩ resistance.
The mask ROM version track of CNV_{SS} pin has no operational interference even if it is connected to Vss pin or Vcc pin via a resistor.

Erasing of Flash memory version

For the parallel I/O mode and the serial I/O mode, set addresses 01000_{16} to $0FFFF_{16}$ as the memory area to be erased. If an incorrect address is set as the memory area to be erased, the product may be permanently damaged.

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

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

DATA REQUIRED FOR One Time PROM PROGRAMMING ORDERS

The following are necessary when ordering a PROM programming service:

1. ROM Programming Confirmation Form*1
2. Mark Specification Form*2
3. Data to be programmed to PROM, in EPROM form (three identical copies)

For the mask ROM confirmation, the ROM programming confirmation, and the mark specifications, refer to the “Mitsubishi MCU Technical Information” Homepage.

✻1 Mask ROM Confirmation Forms
http://www.infomicom.mesc.co.jp/38000/38ordere.htm
✻2 Mark Specification Forms
http://www.infomicom.mesc.co.jp/mela/markform.htm

FUNCTIONAL DESCRIPTION SUPPLEMENT
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 30 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 31 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 85 shows the A-D conversion equivalent circuit, and Figure 86 shows the A-D conversion timing chart.

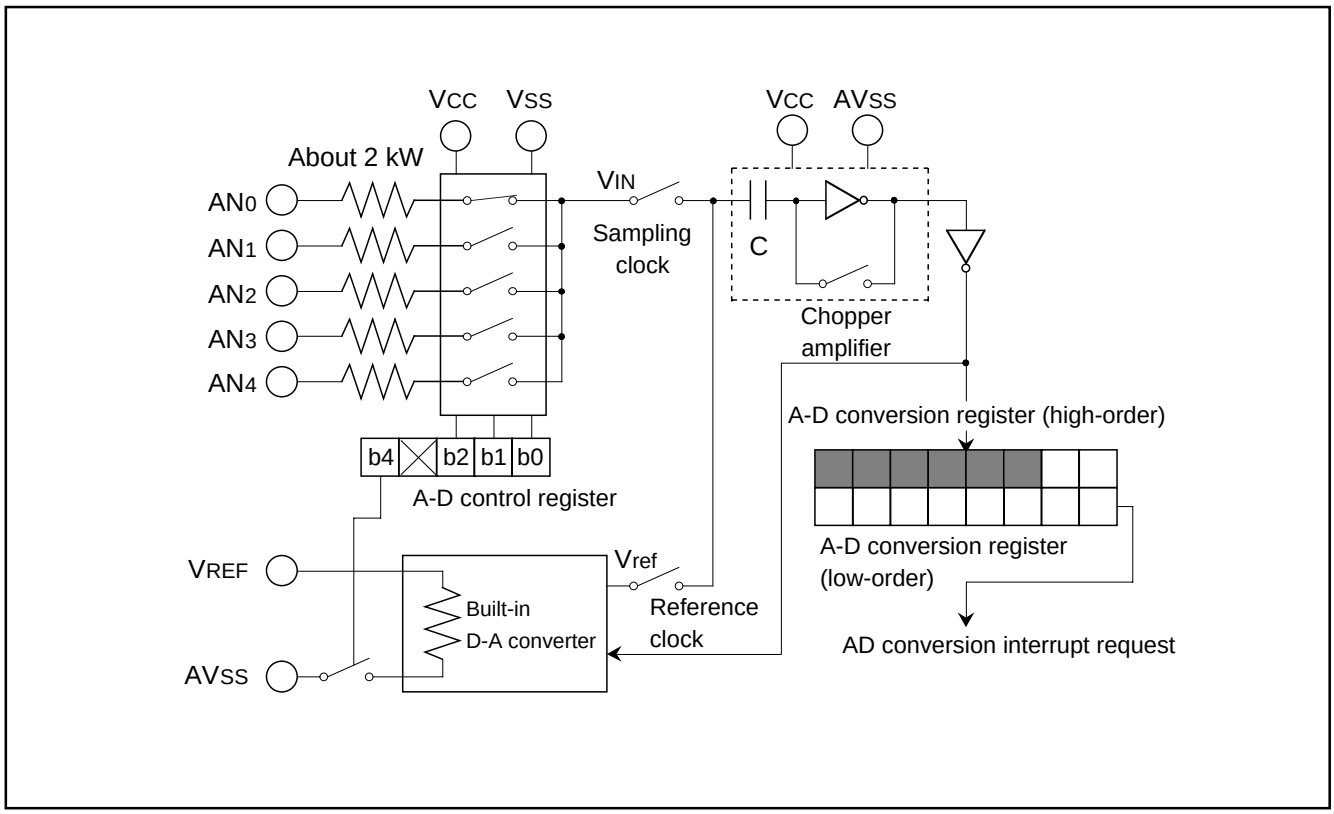


Fig. 85 A-D conversion equivalent circuit

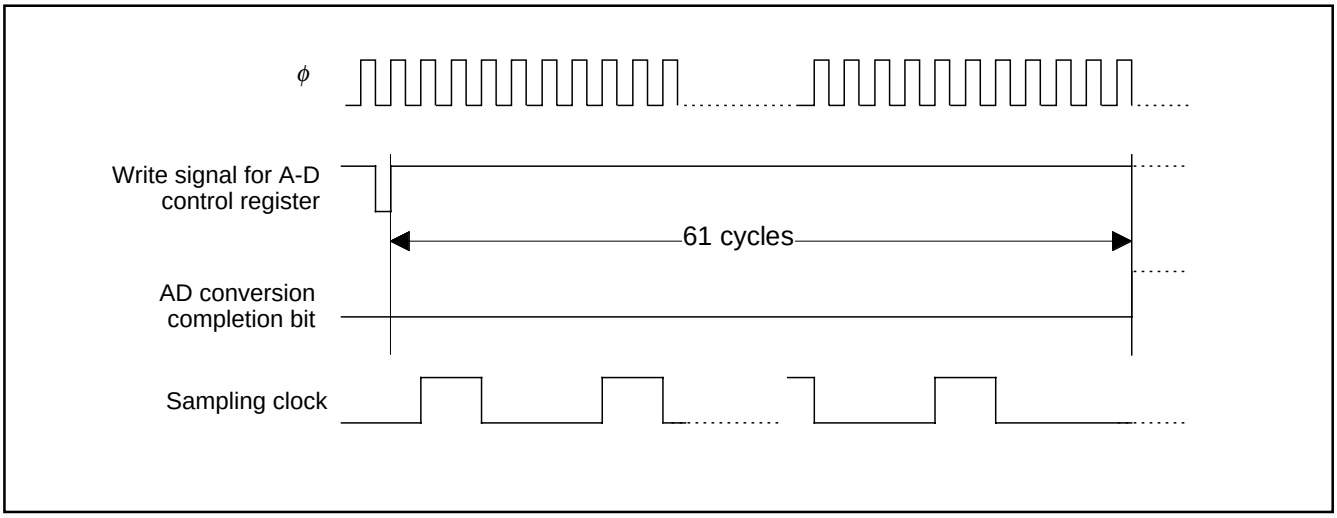


Fig. 86 A-D conversion timing chart

CHAPTER 2

APPLICATION

2.1 I/O port

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

2.1.1 Memory map

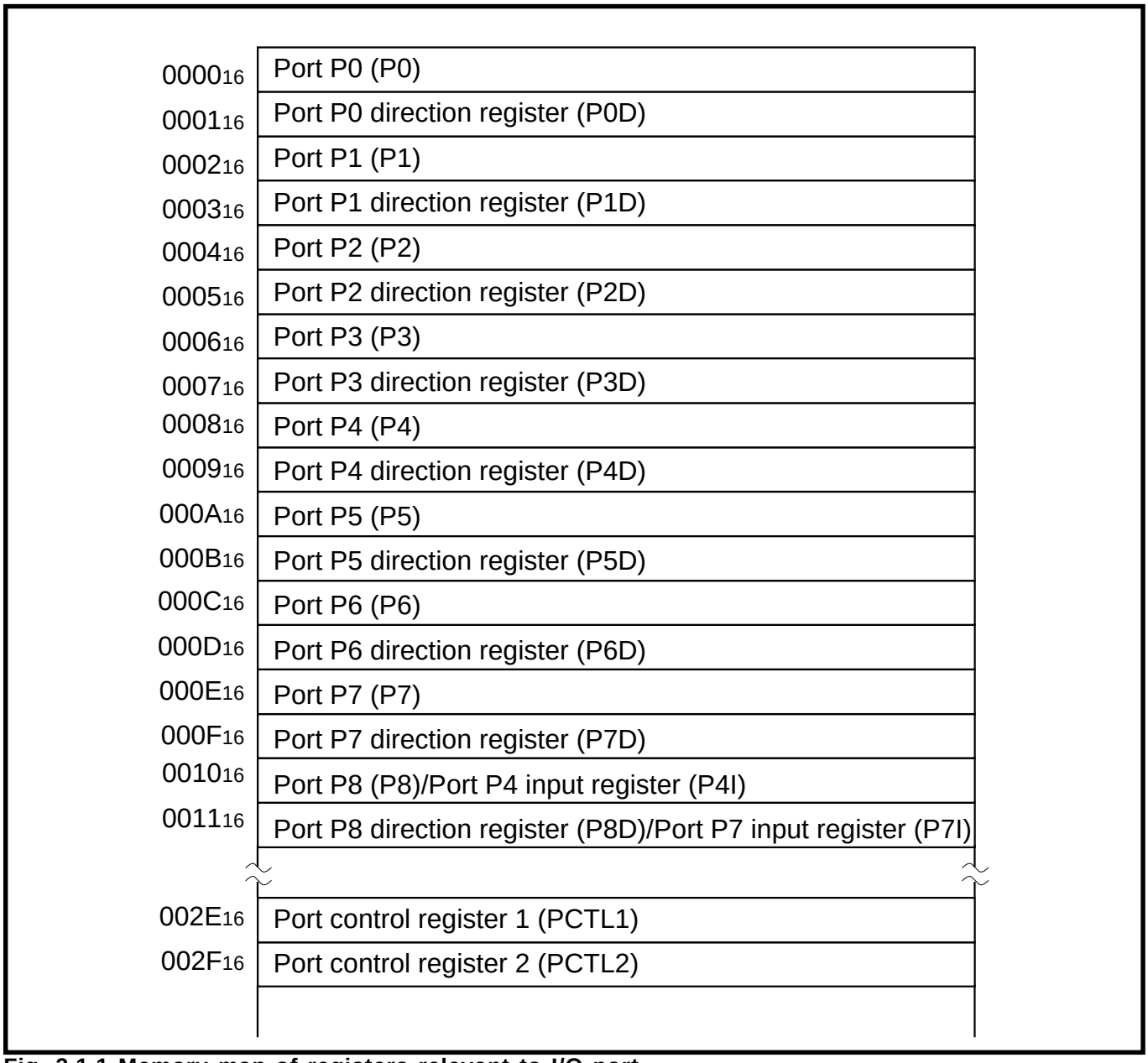


Fig. 2.1.1 Memory map of registers relevant to I/O port

2.1.2 Relevant registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (Pi) (i = 0, 1, 2, 3, 4, 5, 6, 7, 8)
[Address : 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}, $0A_{16}$, $0C_{16}$, $0E_{16}$, 10_{16}]

B	Name	Function	At reset	R	W
0	Port Pi_0	• In output mode Write } Port latch Read } • In input mode Write : Port latch Read : Value of pins	?	○	○
1	Port Pi_1		?	○	○
2	Port Pi_2		?	○	○
3	Port Pi_3		?	○	○
4	Port Pi_4		?	○	○
5	Port Pi_5		?	○	○
6	Port Pi_6		?	○	○
7	Port Pi_7		?	○	○

Fig. 2.1.2 Structure of Port Pi (i = 0 to 8)

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (PiD) (i = 0, 1, 2, 3, 4, 5, 6, 7, 8)
[Address : 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, $0F_{16}$, 11_{16}]

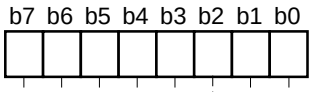
B	Name	Function	At reset	R	W
0	Port Pi direction register	0 : Port Pi_0 input mode 1 : Port Pi_0 output mode	0	×	○
1		0 : Port Pi_1 input mode 1 : Port Pi_1 output mode	0	×	○
2		0 : Port Pi_2 input mode 1 : Port Pi_2 output mode	0	×	○
3		0 : Port Pi_3 input mode 1 : Port Pi_3 output mode	0	×	○
4		0 :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

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

APPLICATION

2.1 I/O port

Port control register 1

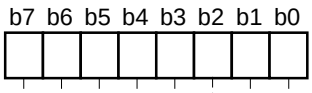


Port control register 1 (PCTL1) [Address 2E 16]

B	Name	Function	At reset	R	W
0	$P0_0$–$P0_3$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	○	○
1	$P0_4$–$P0_7$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	○	○
2	$P1_0$–$P1_3$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	○	○
3	$P1_4$–$P1_7$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	○	○
4	$P3_0$–$P3_3$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
5	$P3_4$–$P3_7$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
6	PWM_0 enable bit	0: PWM_0 output disabled 1: PWM_0 output enabled	0	○	○
7	PWM_1 enable bit	0: PWM_1 output disabled 1: PWM_1 output enabled	0	○	○

Fig. 2.1.4 Structure of Port control register 1

Port control register 2



Port control register 2 (PCTL2) [Address 2F 16]

B	Name	Function	At reset	R	W
0	P4 input level selection bit ($P4_2$–$P4_6$)	0: CMOS level input 1: TTL level input	0	○	○
1	P7 input level selection bit ($P7_0$–$P7_5$)	0: CMOS level input 1: TTL level input	0	○	○
2	P4 output structure selection bit ($P4_2$, $P4_3$, $P4_4$, $P4_6$)	0: CMOS 1: N-channel open-drain	0	○	○
3	P8 function selection bit	0: Port P8/Port P8 direction register 1: Port P4 input register/Port P7 input register	0	○	○
4	INT_2, INT_3, INT_4 interrupt switch bit	0: INT_{20}, INT_{30}, INT_{40} interrupt 1: INT_{21}, INT_{31}, INT_{41} interrupt	0	○	○
5	Timer Y count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ in low-speed mode) 1: $f(X_{CIN})$	0	○	○
6	Oscillation stabilizing time set after STP instruction released bit	0: Automatic set "01 16" to timer 1 and "FF 16" to prescaler 12 1: No automatic set	0	○	○
7	Port output $P4_2$/$P4_3$ clear function selection bit	0: Only software clear 1: Software clear and output data bus buffer 0 reading (system bus side)	0	○	○

Fig. 2.1.5 Structure of Port control register 2

2.1.3 Port P4/P7 input register

Port P4 input register/port P7 input register is selected by setting the port P8 function selection bit of port control register 2 to "1". By reading port P4/P7 input register, the contents of pins can be read out even if the pins are set as output pins. That is, the port state can be read out when the output "H" voltage is falling or the output "L" voltage is rising.

N-channel open-drain output structure is selected by setting the P4 output structure selection bit of port control register 2 to "1". TTL level input is selected by setting the P4 input level selection bit and the P7 input level selection bit of port control register 2 to "1". Pull-up is selected by setting the $P3_0$–$P3_3$ pull-up control bit and the $P3_4$–$P3_7$ pull-up control bit of port control register 1 to "1".

2.1.4 Handling of unused pins

Table 2.1.1 Handling of unused pins (in single-chip mode)

Pins/Ports name	Handling
P0, P1, P2, P3, P4, P5, P6, P7, P8	•Set to the input mode and connect each to Vcc or Vss through a resistor of 1 kΩ to 10 kΩ. •Set to the output mode and open at "L" or "H" level.
V_{REF}	•Connect to Vss (GND).
AV_{SS}	•Connect to Vss (GND).
X_{OUT}	•Open, only when using an external clock.

Table 2.1.2 Handling of unused pins (in memory expansion mode, microprocessor mode)

Pins/Ports name	Handling
$P3_0$, $P3_1$	•Open.
P4, P5, P6, P7, P8	•Set to the input mode and connect each to Vcc or Vss through a resistor of 1 kΩ to 10 kΩ. •Set to the output mode and open at "L" or "H" level.
V_{REF}	•Connect to Vss (GND).
$\overline{ONW}$	•Connect to Vcc through a resistor of 1 kΩ to 10 kΩ.
$\overline{RESET}_{OUT}$	•Open.
ϕ	•Open.
SYNC	•Open.
AV_{SS}	•Connect to Vss (GND).
X_{OUT}	•Open, only when using an external clock.

2.1 I/O port

2.1.5 Notes on input and output pins

(1) Notes in stand-by state

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

Pull-up (connect the port to 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 or pull-down 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**

In I/O ports of the N-channel open-drain, in spite of setting as an output port with its direction register, when the content of the port latch is "1", the transistor becomes the OFF state, which causes the ports to be the high-impedance state. Note that the level becomes "undefined" depending on external circuits.

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

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

(2) Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*2, the value of the unspecified bit may be changed.

● **Reason**

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

- As for a bit which is set for an input port :
  The pin state is read in the CPU, and is written to this bit after bit managing.
- As for a bit which is set for an output port :
  The bit value of the port latch is read in the CPU, and is written to this bit after bit managing.

Note the following :

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

*2 bit managing instructions : **SEB**, and **CLB** instructions

2.1.6 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. With regard to ports which can select the built-in pull-up resistor, the built-in pull-up resistor can be used. As for pins whose potential affects to operation modes such as CNVSS pin 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. With regard to ports which can select the built-in pull-up resistor, the built-in pull-up resistor can be used.
  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.

④ The AVss pin when not using the A-D/D-A converter :
- When not using the A-D/D-A converter, handle a power source pin for the A-D/D-A converter, AVss pin as follows:
- AVss: Connect to the Vss pin

(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 Interrupt

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

2.2.1 Memory map

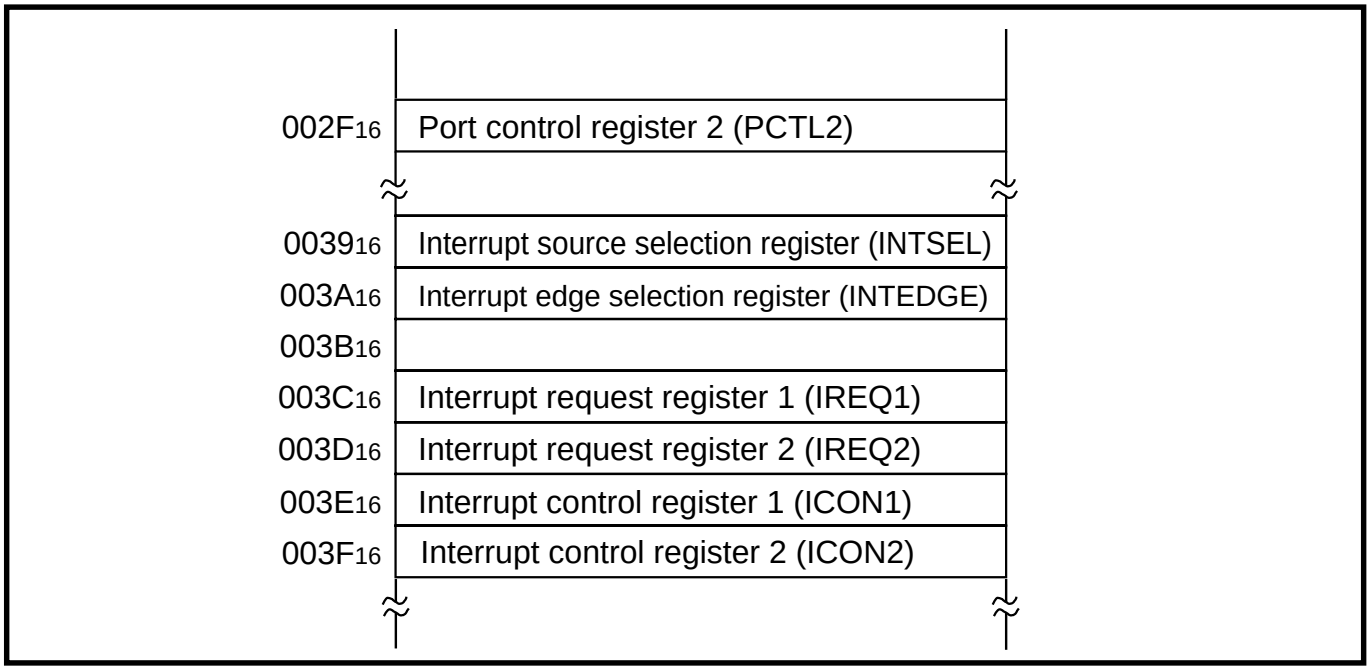


Fig. 2.2.1 Memory map of registers relevant to interrupt

2.2.2 Relevant registers

Port control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Port control register 2 (PCTL2) [Address 2F 16]

B	Name	Function	At reset	R	W
0	P4 input level selection bit ($P4_2$–$P4_6$)	0: CMOS level input 1: TTL level input	0	○	○
1	P7 input level selection bit ($P7_0$–$P7_5$)	0: CMOS level input 1: TTL level input	0	○	○
2	P4 output structure selection bit ($P4_2$, $P4_3$, $P4_4$, $P4_6$)	0: CMOS 1: N-channel open-drain	0	○	○
3	P8 function selection bit	0: Port P8/Port P8 direction register 1: Port P4 input register/Port P7 input register	0	○	○
4	INT_2, INT_3, INT_4 interrupt switch bit	0: INT_{20}, INT_{30}, INT_{40} interrupt 1: INT_{21}, INT_{31}, INT_{41} interrupt	0	○	○
5	Timer Y count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ in low-speed mode) 1: $f(X_{CIN})$	0	○	○
6	Oscillation stabilizing time set after STP instruction released bit	0: Automatic set "01_{16}" to timer 1 and "FF_{16}" to prescaler 12 1: No automatic set	0	○	○
7	Port output $P4_2$/$P4_3$ clear function selection bit	0: Only software clear 1: Software clear and output data bus buffer 0 reading (system bus side)	0	○	○

Fig. 2.2.2 Structure of Port control register 2

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register [INTSEL: address 0039_{16}]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt source selection bit	0 : INT_0 interrupt 1 : Input buffer full interrupt	0	○	○
1	INT_1/output buffer empty interrupt source selection bit	0 : INT_1 interrupt 1 : Output buffer empty interrupt	0	○	○
2	Serial I/O1 transmit/S_{CL},S_{DA} interrupt source selection bit	0 : Serial I/O1 transmit interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
3	$CNTR_0$/S_{CL},S_{DA} interrupt source selection bit	0 : $CNTR_0$ interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
4	Serial I/O2/I^2C interrupt source selection bit	0 : Serial I/O2 interrupt ✻2 1 : I^2C interrupt	0	○	○
5	INT_2/I^2C interrupt source selection bit	0 : INT_2 interrupt ✻2 1 : I^2C interrupt	0	○	○
6	$CNTR_1$/key-on wake-up interrupt source selection bit	0 : $CNTR_1$ interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○
7	AD converter/key-on wake-up interrupt source selection bit	0 : A-D converter interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○

✻1: Do not write "1" to these bits simultaneously.
✻2: Do not write "1" to these bits simultaneously.
✻3: Do not write "1" to these bits simultaneously.

Fig. 2.2.3 Structure of Interrupt source selection register

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register (INTEDGE) [Address : $3A_{16}$]

B	Name	Function	At reset	R	W
0	INT_0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT_1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×
3	INT_2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4	INT_3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
5	INT_4 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
6	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
7			0	○	×

Fig. 2.2.4 Structure of Interrupt edge selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT_1/output buffer empty interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.2.5 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2 (IREQ2) [Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	$CNTR_1$/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT_4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.2.6 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	Function	At reset	R	W
0	INT_0/input buffer full interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1/output buffer empty interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.2.7 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	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	$CNTR_1$/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	INT_4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.2.8 Structure of Interrupt control register 2

2.2 Interrupt

2.2.3 Interrupt source

The 3886 group permits interrupts of 16 sources among 21 sources. These are 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 a variety of 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 2.2.1.

Table 2.2.1 Interrupt sources, vector addresses and priority of 3886 group

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)
Input buffer full (IBF)				At input data bus buffer writing	
INT1	3	FFF916	FFF816	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
Output buffer empty (OBE)				At output data bus buffer reading	
Serial I/O1 reception	4	FFF716	FFF616	At completion of serial I/O1 data reception	Valid when serial I/O1 is selected
Serial I/O1 transmission	5	FFF516	FFF416	At completion of serial I/O1 transfer shift or when transmission buffer is empty	Valid when serial I/O1 is selected
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
Timer X	6	FFF316	FFF216	At timer X underflow	
Timer Y	7	FFF116	FFF016	At timer Y underflow	
Timer 1	8	FFEF16	FFEE16	At timer 1 underflow	STP release timer underflow
Timer 2	9	FFED16	FFEC16	At timer 2 underflow	
CNTR0	10	FFEB16	FFEA16	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
CNTR1	11	FFE916	FFE816	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
Key-on wake-up				At falling of port P3 (at input) input logical level AND	External interrupt (falling edge valid)
Serial I/O2	12	FFE716	FFE616	At completion of serial I/O2 data transfer	Valid when serial I/O2 is selected
I^2C				At completion of data transfer	
INT2	13	FFE516	FFE416	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
I^2C				At completion of data transfer	
INT3	14	FFE316	FFE216	At detection of either rising or falling edge of INT3 input	External interrupt (active edge selectable)
INT4	15	FFE116	FFE016	At detection of either rising or falling edge of INT4 input	External interrupt (active edge selectable)
A-D converter	16	FFDF16	FFDE16	At completion of A-D conversion	
Key-on wake-up				At falling of port P3 (at input) input logical level AND	External interrupt (falling edge valid)
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.

2.2.4 Interrupt operation

When an interrupt request is accepted, the contents of the following registers just before acceptance of the interrupt requests is automatically pushed onto the stack area in the order of ①, ② and ③.

①High-order contents of program counter (PC_H)
②Low-order contents of program counter (PC_L)
③Contents of processor status register (PS)

After the contents of the above registers are pushed onto the stack area, the accepted interrupt vector address enters the program counter and consequently the interrupt processing routine is executed.
When the RTI instruction is executed at the end of the interrupt processing routine, the contents of the above registers pushed onto the stack area are restored to the respective registers in the order of ③, ② and ①; and the microcomputer resumes the processing executed just before acceptance of the interrupts. Figure 2.2.9 shows an interrupt operation diagram.

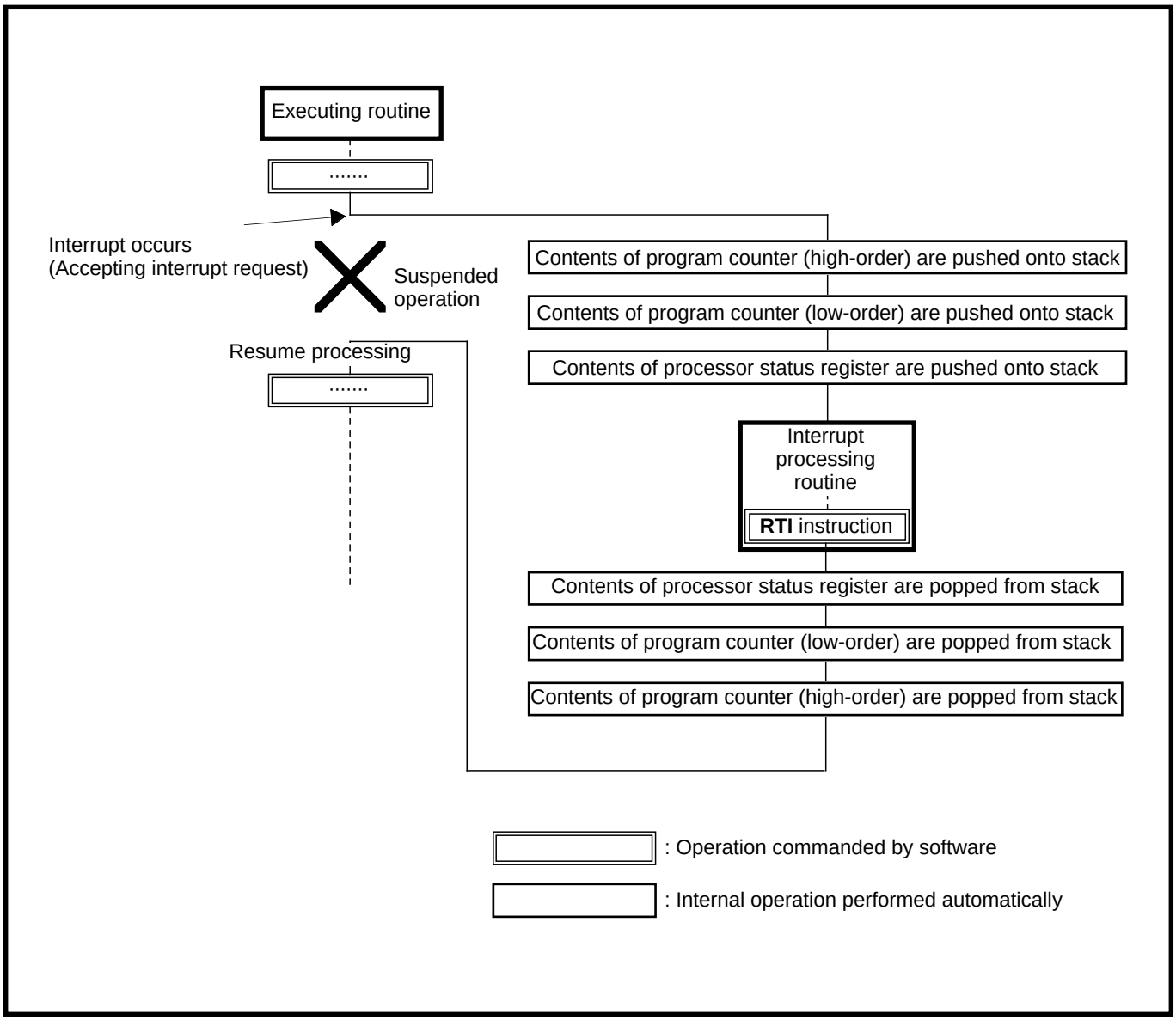


Fig. 2.2.9 Interrupt operation diagram

2.2 Interrupt

(1) Processing upon acceptance of interrupt request

Upon acceptance of an interrupt request, the following operations are automatically performed.

①The processing being executed is stopped.

②The contents of the program counter and the processor status register are pushed onto the stack area. Figure 2.2.10 shows the changes of the stack pointer and the program counter upon acceptance of an interrupt request.

③Concurrently with the push operation, the jump destination address (the beginning address of the interrupt processing routine) of the occurring interrupt stored in the vector address is set in the program counter, then the interrupt processing routine is executed.

④After the interrupt processing routine is started, the corresponding interrupt request bit is automatically cleared to "0". The interrupt disable flag is set to "1" so that multiple interrupts are disabled.

Accordingly, for executing the interrupt processing routine, it is necessary to set the jump destination address in the vector area corresponding to each interrupt.

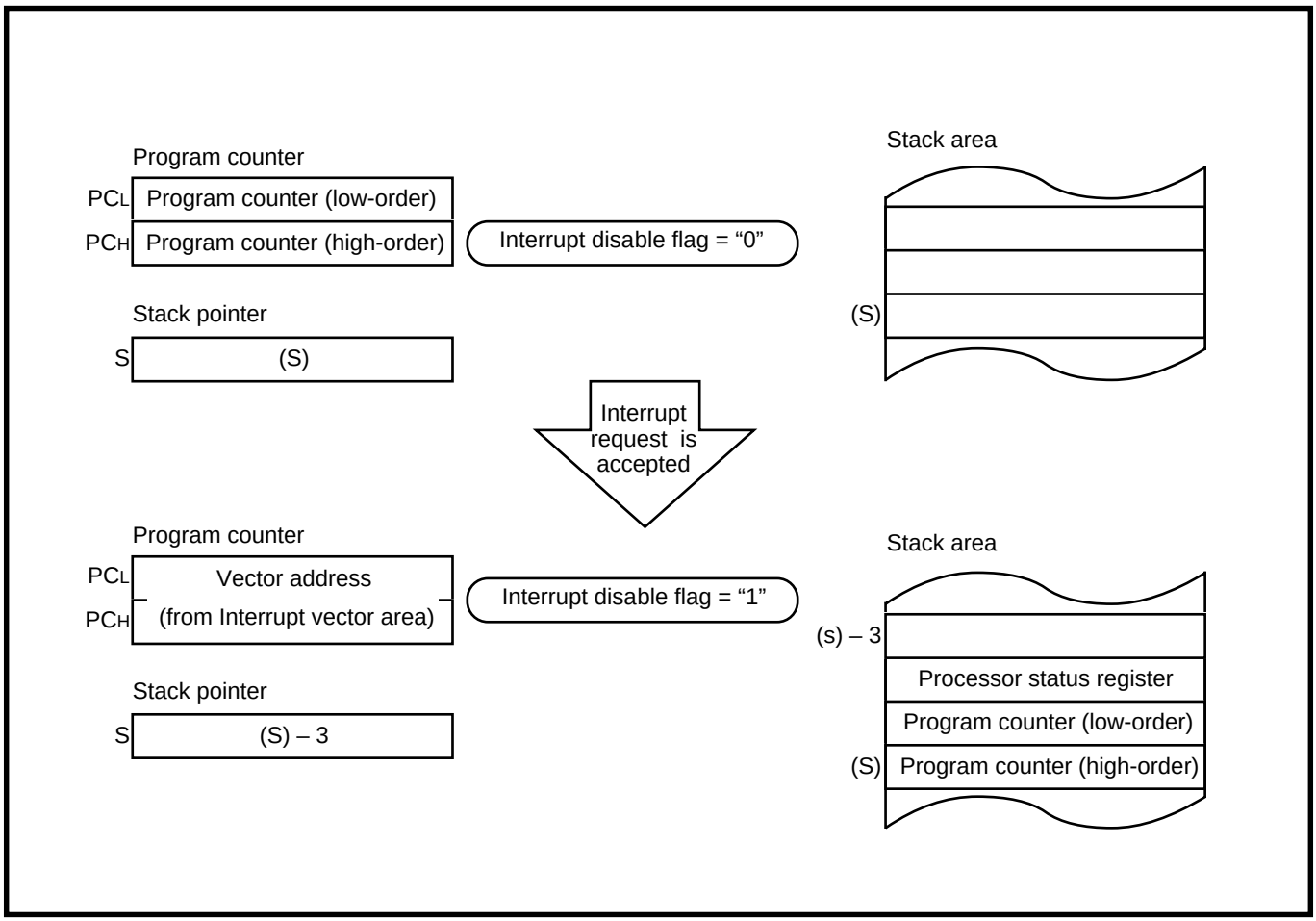


Fig. 2.2.10 Changes of stack pointer and program counter upon acceptance of interrupt request

(2) Timing after acceptance of interrupt request

The interrupt processing routine begins with the machine cycle following the completion of the instruction that is currently being executed.

Figure 2.2.11 shows the time up to execution of interrupt processing routine and Figure 2.2.12 shows the timing chart after acceptance of interrupt request.

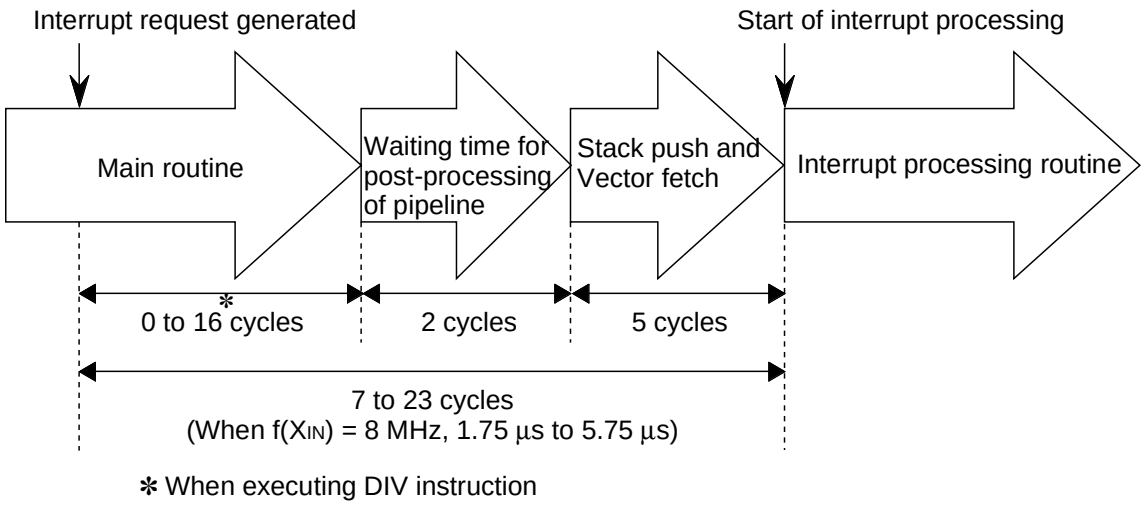
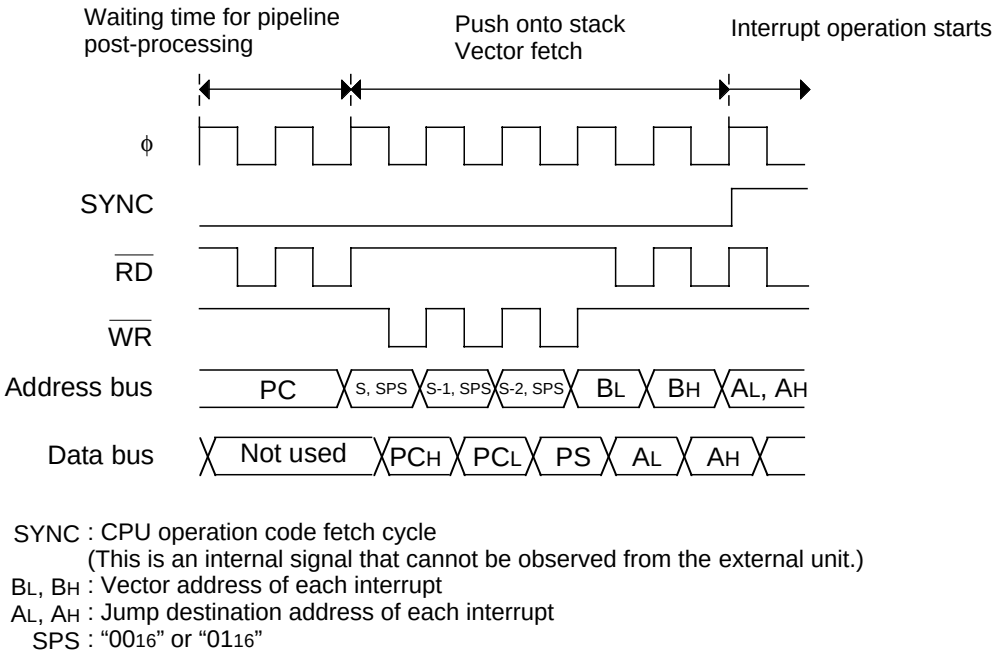


Fig. 2.2.11 Time up to execution of interrupt processing routine



SYNC : CPU operation code fetch cycle
(This is an internal signal that 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. 2.2.12 Timing chart after acceptance of interrupt request

2.2.5 Interrupt control

The acceptance of all interrupts, excluding the BRK instruction interrupt, can be controlled by the interrupt request bit, interrupt enable bit, and an interrupt disable flag, as described in detail below. Figure 2.2.13 shows an interrupt control diagram.

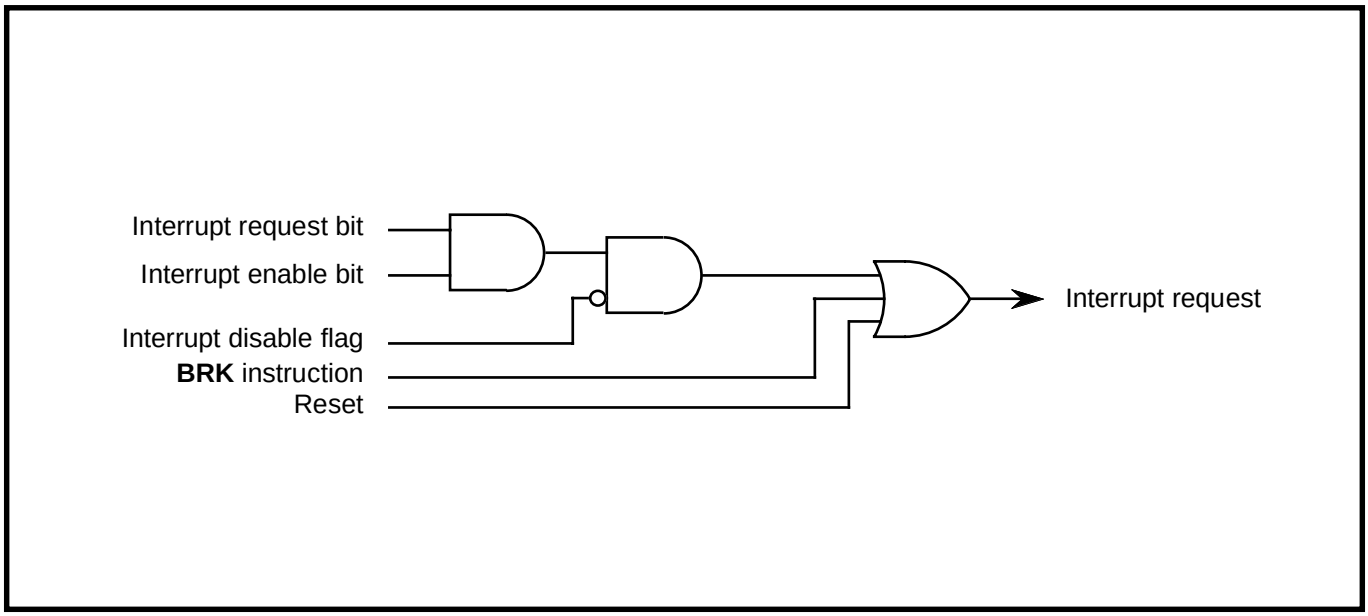


Fig. 2.2.13 Interrupt control diagram

The interrupt request bit, interrupt enable bit and interrupt disable flag function independently and do not affect each other. An interrupt is accepted when all the following conditions are satisfied.

- Interrupt request bit ..... "1"
- Interrupt enable bit ..... "1"
- Interrupt disable flag ..... "0"

Though the interrupt priority is determined by hardware, a variety of priority processing can be performed by software using the above bits and flag. Table 2.2.2 shows a list of interrupt control bits according to the interrupt source.

(1) Interrupt request bits

The interrupt request bits are allocated to the interrupt request register 1 (address $3C_{16}$) and interrupt request register 2 (address $3D_{16}$).

The occurrence of an interrupt request causes the corresponding interrupt request bit to be set to "1". The interrupt request bit is held in the "1" state until the interrupt is accepted. When the interrupt is accepted, this bit is automatically cleared to "0".

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

(2) Interrupt enable bits

The interrupt enable bits are allocated to the interrupt control register 1 (address $003E_{16}$) and the interrupt control register 2 (address $3F_{16}$).

The interrupt enable bits control the acceptance of the corresponding interrupt request.

When an interrupt enable bit is "0", the corresponding interrupt request is disabled. If an interrupt request occurs when this bit is "0", the corresponding interrupt request bit is set to "1" but the interrupt is not accepted. In this case, unless the interrupt request bit is set to "0" by software, the interrupt request bit remains in the "1" state.

When an interrupt enable bit is "1", the corresponding interrupt is enabled. If an interrupt request occurs when this bit is "1", the interrupt is accepted (when interrupt disable flag = "0").

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

(3) Interrupt disable flag

The interrupt disable flag is allocated to bit 2 of the processor status register. The interrupt disable flag controls the acceptance of interrupt request except BRK instruction.

When this flag is "1", the acceptance of interrupt requests is disabled. When the flag is "0", the acceptance of interrupt requests is enabled. This flag is set to "1" with the SEI instruction and is set to "0" with the CLI instruction.

When a main routine branches to an interrupt processing routine, this flag is automatically set to "1", so that multiple interrupts are disabled. To use multiple interrupts, set this flag to "0" with the CLI instruction within the interrupt processing routine. Figure 2.2.14 shows an example of multiple interrupts.

Table 2.2.2 List of interrupt bits according to interrupt source

Interrupt source	Interrupt enable bit		Interrupt request bit	
	Address	Bit	Address	Bit
INT_0/Input buffer full	$003E_{16}$	b0	$003C_{16}$	b0
INT_1/Output buffer empty	$003E_{16}$	b1	$003C_{16}$	b1
Serial I/O1 reception	$003E_{16}$	b2	$003C_{16}$	b2
Serial I/O1 transmission/S_{CL}, S_{DA}	$003E_{16}$	b3	$003C_{16}$	b3
Timer X	$003E_{16}$	b4	$003C_{16}$	b4
Timer Y	$003E_{16}$	b5	$003C_{16}$	b5
Timer 1	$003E_{16}$	b6	$003C_{16}$	b6
Timer 2	$003E_{16}$	b7	$003C_{16}$	b7
$CNTR_0$/S_{CL}, S_{DA}	$003F_{16}$	b0	$003D_{16}$	b0
CNTR1/Key-on wake-up	$003F_{16}$	b1	$003D_{16}$	b1
Serial I/O2/I^2C	$003F_{16}$	b2	$003D_{16}$	b2
INT_2/I^2C	$003F_{16}$	b3	$003D_{16}$	b3
INT_3	$003F_{16}$	b4	$003D_{16}$	b4
INT_4	$003F_{16}$	b5	$003D_{16}$	b5
A-D converter/Key-on wake-up	$003F_{16}$	b6	$003D_{16}$	b6

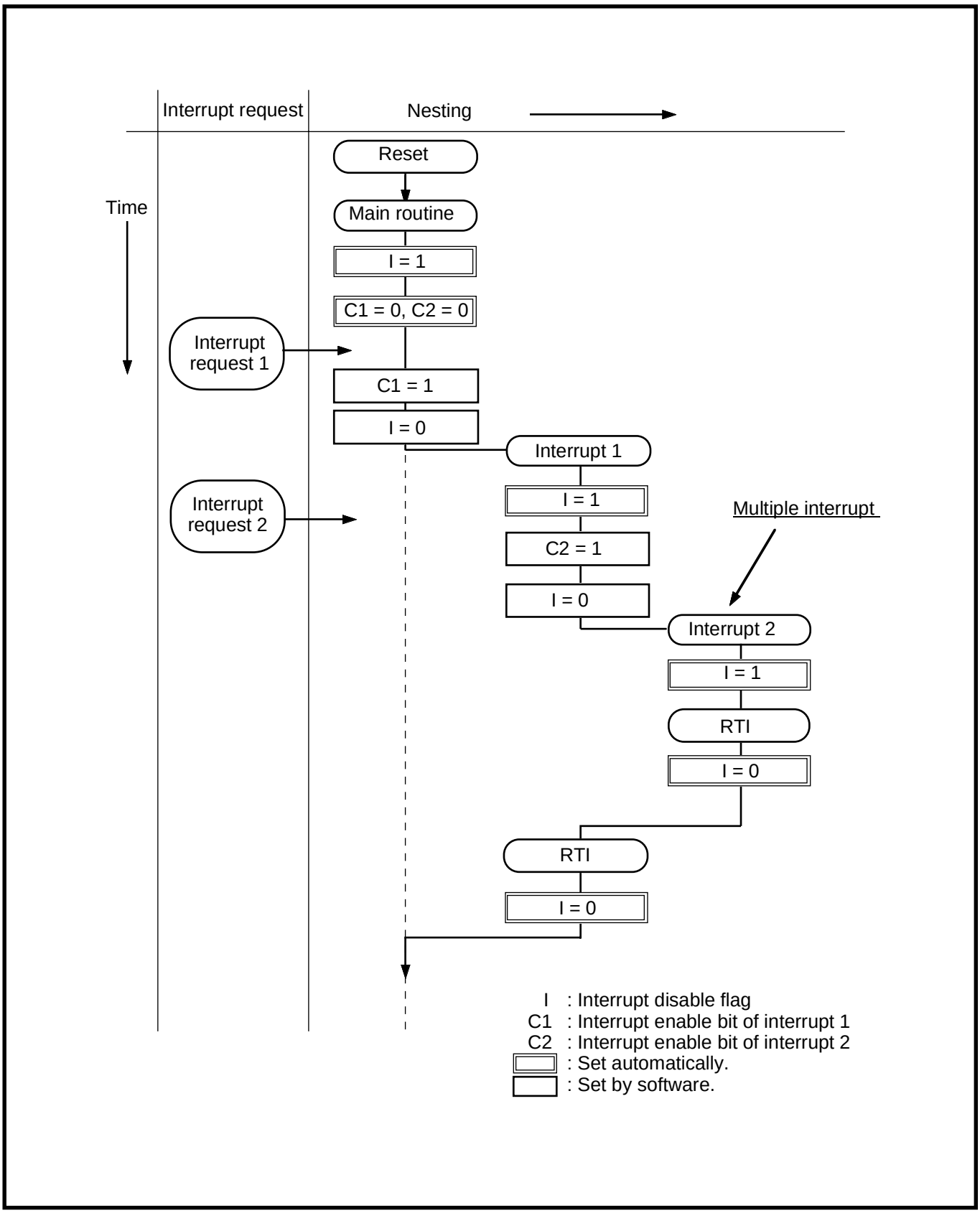


Fig. 2.2.14 Example of multiple interrupts

2.2.6 INT interrupt

The INT interrupt requests is generated when the microcomputer detects a level change of each INT pin (INT_0–INT_4).

(1) Active edge selection

INT_0–INT_4 can be selected from either a falling edge or rising edge detection as an active edge by the interrupt edge selection register. In the "0" state, the falling edge of the corresponding pin is detected. In the "1" state, the rising edge of the corresponding pin is detected.

(2) INT_0–INT_2 interrupt sources selection

Which of interrupt source of the following interrupt sources can be selected by the interrupt source selection register (address 39_{16}). (Set each bit to "0" when using INT.)

- INT_0 or input buffer full (bit 0)
- INT_1 or output buffer empty (bit 1)
- INT_2 or I^2C (bit 5)

(3) INT_2–INT_4 input pins selection

The occurrence sources of the external interrupt INT_2 to INT_4 can be selected from which of the following by the INT_2, INT_3, INT_4 interrupt switch bit of the port control register 2 (address $2F_{16}$).

- INT_{20}, INT_{30}, INT_{40} or
- INT_{21}, INT_{31}, INT_{41}

2.2.7 Key input interrupt

A key input interrupt request is generated by applying “L” level to any port P3 pin that has been set to the input mode. In other words, it is generated when AND of the input level goes from “1” to “0”.

(1) Connection example when Key input interrupt is used

When using the Key input interrupt, compose an active-low key matrix which inputs to port P3. Figure 2.2.15 shows a connection example and the port P3 block diagram when using a key input interrupt. In the connection example in Figure 2.2.15, a key input interrupt request is generated by pressing one of the keys corresponding to ports $P3_0$ to $P3_3$.

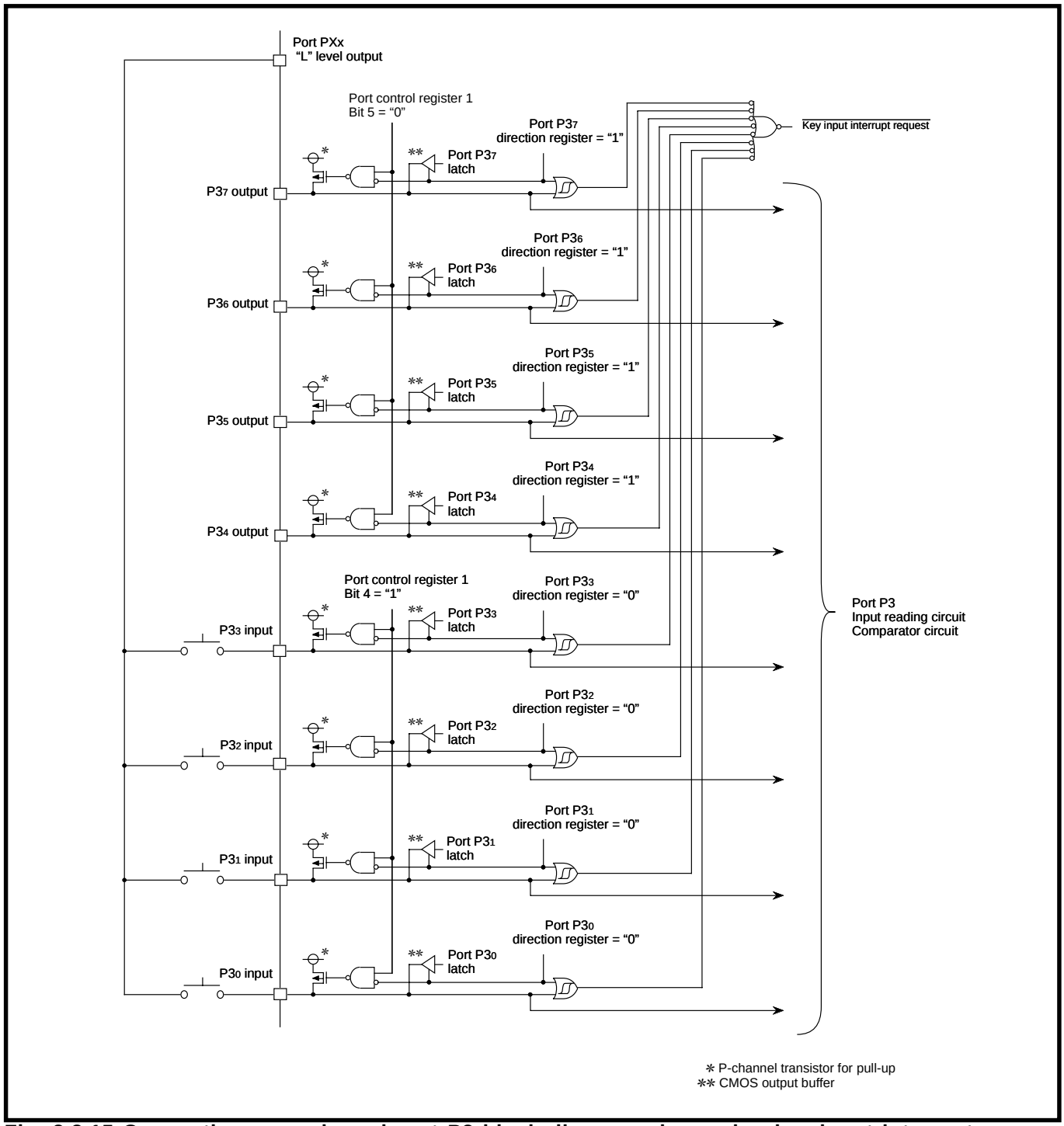
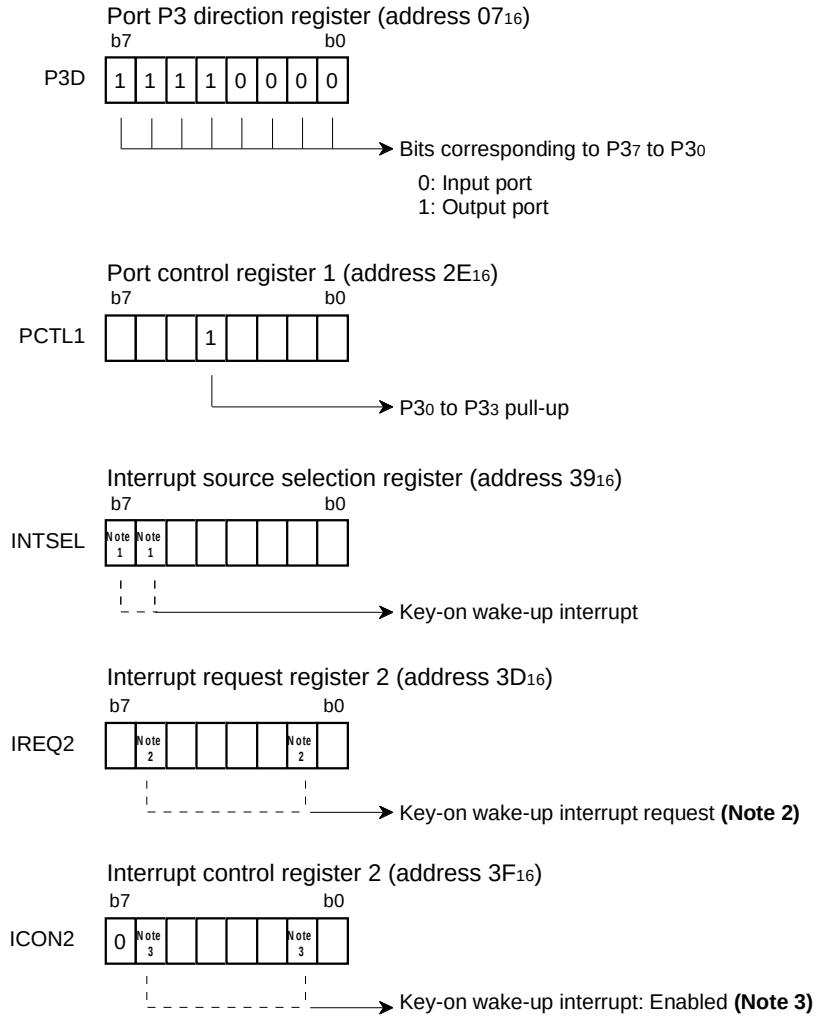


Fig. 2.2.15 Connection example and port P3 block diagram when using key input interrupt

(2) Relevant registers setting

Figure 2.2.16 shows the relevant registers setting (corresponding to Figure 2.2.15).



Notes 1: Set "1" to the bit which is selected as the key-on wake-up interrupt source from either b7 or b6 of the interrupt source selection register. Do not set "1" to these bits simultaneously. (When setting "1" to b6, set "0" to b7. When setting "0" to b6, set "1" to b7.)

2: When setting "1" to b7 of the interrupt source selection register, set "0" to b6 of the interrupt request register 2. When setting "1" to b6 of the interrupt source selection register, set "0" to b1 of the interrupt request register 2.

3: When setting "1" to b7 of the interrupt source selection register, set "1" to b6 of the interrupt control register 2. When setting "1" to b6 of the interrupt source selection register, set "1" to b1 of the interrupt control register 2.

Fig. 2.2.16 Registers setting relevant to key input interrupt (corresponding to Figure 2.2.15)

(3) Key input interrupt source selection

When using a key input interrupt source, select which of the following by the interrupt source selection register (address 39₁₆).

•$CNTR_0$ or key-on wake-up (bit 6)

•A-D converter or key-on wake-up (bit 7)

2.2.8 Notes on interrupts

(1) Switching external interrupt detection edge

When switching the external interrupt detection edge, switch it in the following sequence.

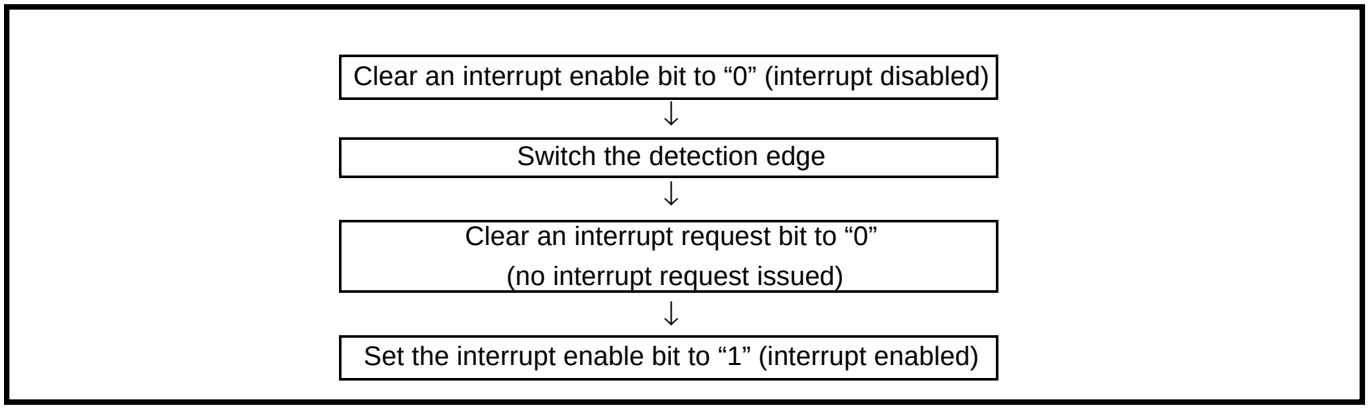


Fig. 2.2.17 Sequence of switching detection edge

■ **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.

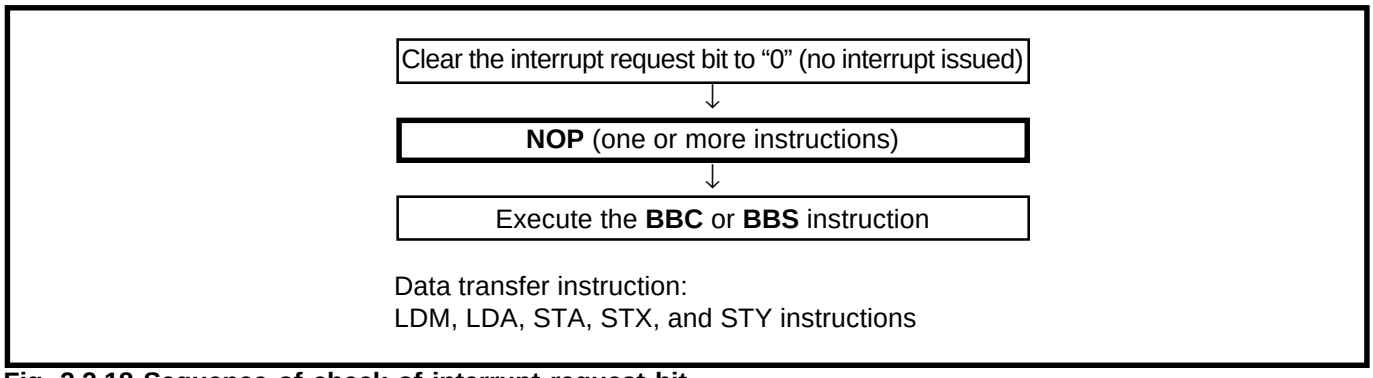


Fig. 2.2.18 Sequence of check of interrupt request bit

■ **Reason**

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

(3) Change of relevant register settings

When the setting of the following registers or bits is changed, the interrupt request bit may be set to "1".

•Interrupt edge selection register (address $3A_{16}$)
•Interrupt source selection register (address 39_{16})
•INT2, INT3, INT4 interrupt switch bit of port control register 2 (bit 4 of address $2F_{16}$)

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

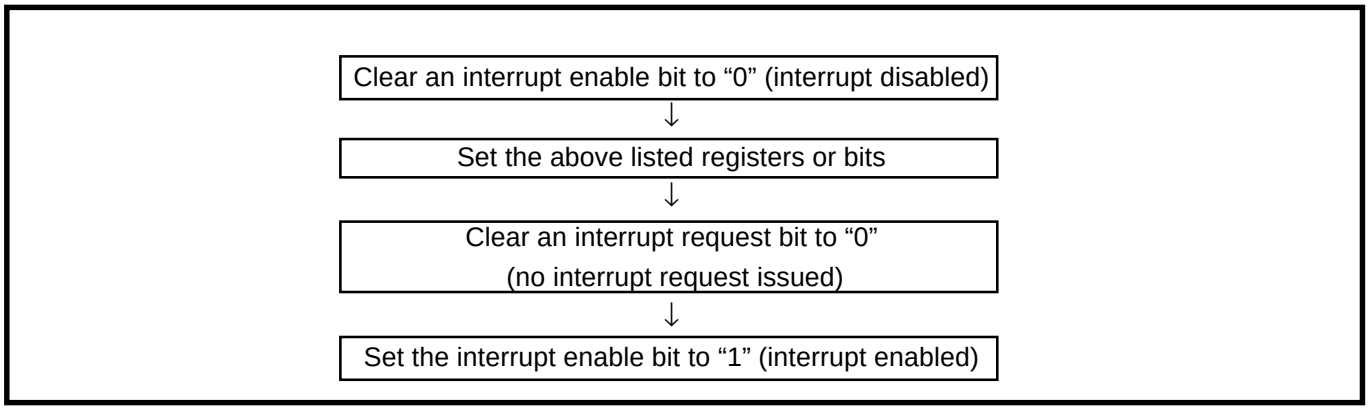


Fig. 2.2.19 Sequence of changing relevant register

2.3 Timer

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

2.3.1 Memory map

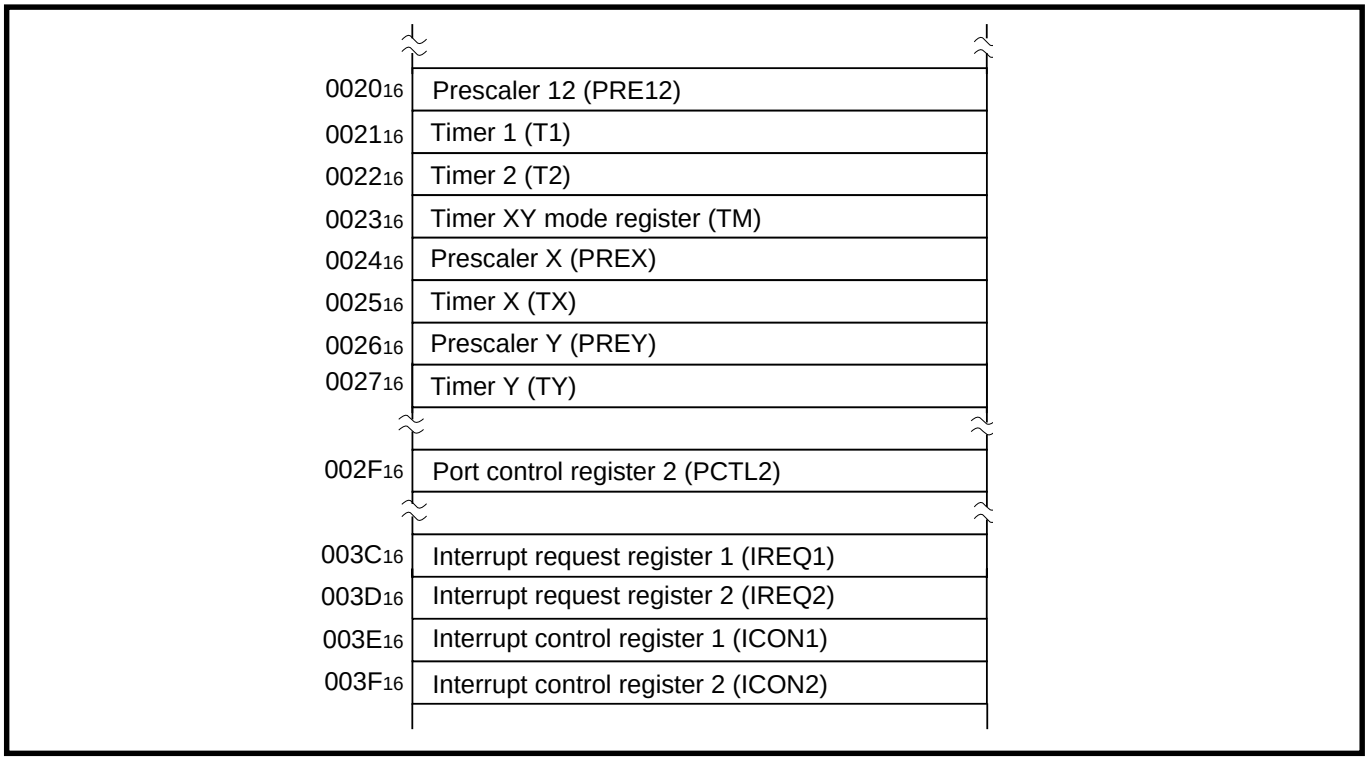
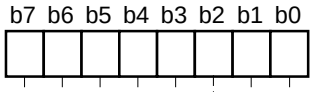


Fig. 2.3.1 Memory map of registers relevant to timers

2.3.2 Relevant registers

Prescaler 12, Prescaler X, Prescaler Y



Prescaler 12 (PRE12) [Address : 20 16]
Prescaler X (PREX) [Address : 24 16]
Prescaler Y (PREY) [Address : 26 16]

B	Name	Function	At reset	R	W
0		•Set a count value of each prescaler. •The value set in this register is written to both each prescaler and the corresponding prescaler latch at the same time. •When this register is read out, the count value of the corresponding prescaler is read out.	1	○	○
1			1	○	○
2			1	○	○
3			1	○	○
4			1	○	○
5			1	○	○
6			1	○	○
7			1	○	○

Fig. 2.3.2 Structure of Prescaler 12, Prescaler X, Prescaler Y

Timer 1

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1 (T1) [Address : 21_{16}]

B	Name	Function	At reset	R	W
0	•Set a count value of timer 1. •The value set in this register is written to both timer 1 and timer 1 latch at the same time. •When this register is read out, the timer 1's count value is read out.		1	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Fig. 2.3.3 Structure of Timer 1

Timer 2, Timer X, Timer Y

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2 (T2) [Address : 22_{16}]
Timer X (TX) [Address : 25_{16}]
Timer Y (TY) [Address : 27_{16}]

B	Name	Function	At reset	R	W
0	•Set a count value of each timer. •The value set in this register is written to both each timer and each timer latch at the same time. •When this register is read out, each timer's count value is read out.		1	○	○
1			1	○	○
2			1	○	○
3			1	○	○
4			1	○	○
5			1	○	○
6			1	○	○
7			1	○	○

Fig. 2.3.4 Structure of Timer 2, Timer X, Timer Y

Timer XY mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer XY mode register (TM) [Address : 23_{16}]

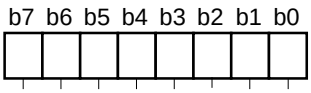
B	Name	Function	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	$CNTR_0$ active edge selection bit	The function depends on the operating mode of Timer X. (Refer to Table 2.2.1)	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Timer Y operating mode bits	b5 b4 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
5			0	○	○
6	$CNTR_1$ active edge selection bit	The function depends on the operating mode of Timer Y. (Refer to Table 2.2.1)	0	○	○
7	Timer Y count stop bit	0 : Count start 1 : Count stop	0	○	○

Fig. 2.3.5 Structure of Timer XY mode register

Table 2.3.1 $CNTR_0$ /$CNTR_1$ active edge selection bit function

Timer X /Timer Y operation modes	$CNTR_0$ / $CNTR_1$ active edge selection bit (23_{16}) contents	
Timer mode	"0"	$CNTR_0$ / $CNTR_1$ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	$CNTR_0$ / $CNTR_1$ interrupt request occurrence: Rising edge ; No influence to timer count
Pulse output mode	"0"	Pulse output start: Beginning at "H" level $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Rising edge
Event counter mode	"0"	Timer X / Timer Y: Rising edge count $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: Falling edge count $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer X / Timer Y: "H" level width measurement $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: "L" level width measurement $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Rising edge

Port control register 2



Port control register 2 (PCTL2) [Address 2F 16]

B	Name	Function	At reset	R	W
0	P4 input level selection bit (P4 2–P4 6)	0: CMOS level input 1: TTL level input	0	○	○
1	P7 input level selection bit (P7 0–P7 5)	0: CMOS level input 1: TTL level input	0	○	○
2	P4 output structure selection bit (P4 2, P4 3, P4 4, P4 6)	0: CMOS 1: N-channel open-drain	0	○	○
3	P8 function selection bit	0: Port P8/Port P8 direction register 1: Port P4 input register/Port P7 input register	0	○	○
4	INT2, INT3, INT4 interrupt switch bit	0: INT20, INT30, INT40 interrupt 1: INT21, INT31, INT41 interrupt	0	○	○
5	Timer Y count source selection bit	0: f(XIN)/16 (f(XCIN)/16 in low-speed mode) 1: f(XCIN)	0	○	○
6	Oscillation stabilizing time set after STP instruction released bit	0: Automatic set “01 16” to timer 1 and “FF 16” to prescaler 12 1: No automatic set	0	○	○
7	Port output P42/P43 clear function selection bit	0: Only software clear 1: Software clear and output data bus buffer 0 reading (system bus side)	0	○	○

Fig. 2.3.6 Structure of Port control register 2

2.3 Timer

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : 3C 16]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT_1/output buffer empty interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.3.7 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2 (IREQ2) [Address : 3D 16]

B	Name	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	$CNTR_1$/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT_4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.3.8 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	Function	At reset	R	W
0	INT_0/input buffer full interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1/output buffer empty interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.3.9 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	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	$CNTR_1$/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	INT_4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.3.10 Structure of Interrupt control register 2

2.3.3 Timer application examples

(1) Basic functions and uses

[Function 1] Control of Event interval (Timer X, Timer Y, Timer 1, Timer 2)

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

<Use>

- •Generation of an output signal timing
- •Generation of a wait time

[Function 2] Control of Cyclic operation (Timer X, Timer Y, Timer 1, Timer 2)

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>

- •Generation of cyclic interrupts
- •Clock function (measurement of 250 ms); see Application example 1
- •Control of a main routine cycle

[Function 3] Output of Rectangular waveform (Timer X, Timer Y)

The output level of the $CNTR_0$ pin or $CNTR_1$ pin is inverted each time the timer underflows (in the pulse output mode).

<Use>

- •Piezoelectric buzzer output; see Application example 2
- •Generation of the remote control carrier waveforms

[Function 4] Count of External pulses (Timer X, Timer Y)

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

<Use>

- •Frequency measurement; see Application example 3
- •Division of external pulses
- •Generation of interrupts due to a cycle using external pulses as the count source; count of a reel pulse

[Function 5] Measurement of External pulse width (Timer X, Timer Y)

The “H” or “L” level width of external pulses input to $CNTR_0$ pin or $CNTR_1$ pin is measured (in the pulse width measurement mode).

<Use>

- •Measurement of external pulse frequency (measurement of pulse width of FG pulse* for a motor); see Application example 4
- •Measurement of external pulse duty (when the frequency is fixed)

FG pulse*: Pulse used for detecting the motor speed to control the motor speed.

(2) Timer application example 1: Clock function (measurement of 250 ms)

Outline: The input clock is divided by the timer so that the clock can count up at 250 ms intervals.

Specifications: •The clock f(X_{IN}) = 4.19 MHz (2^{22} Hz) is divided by the timer.
•The clock is counted up in the process routine of the timer X interrupt which occurs at 250 ms intervals.

Figure 2.3.11 shows the timers connection and setting of division ratios; Figure 2.3.12 shows the relevant registers setting; Figure 2.3.13 shows the control procedure.

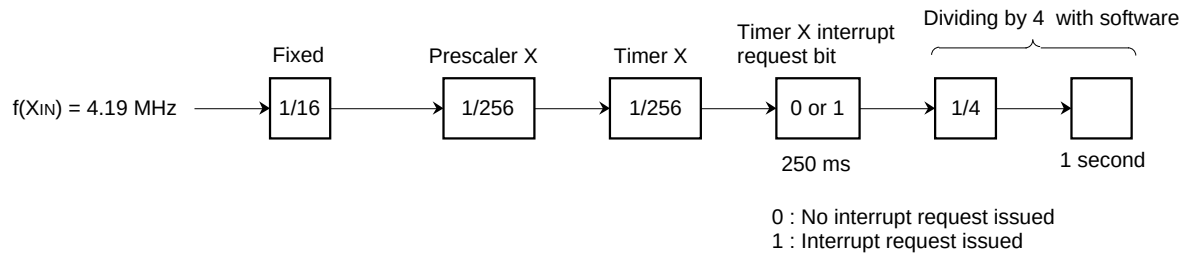


Fig. 2.3.11 Timers connection and setting of division ratios

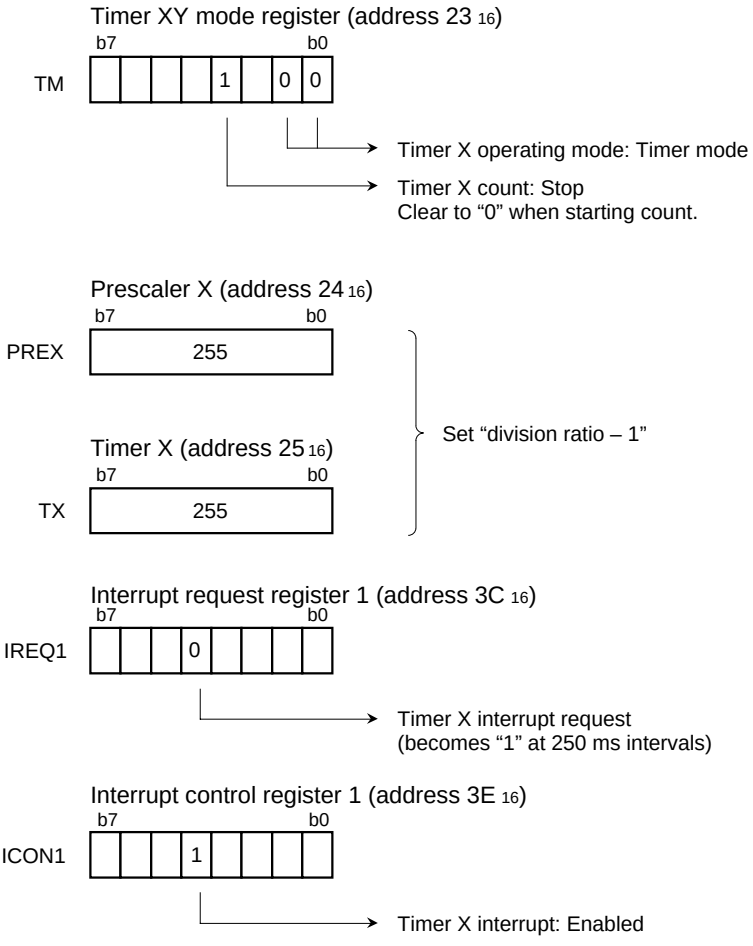


Fig. 2.3.12 Relevant registers setting

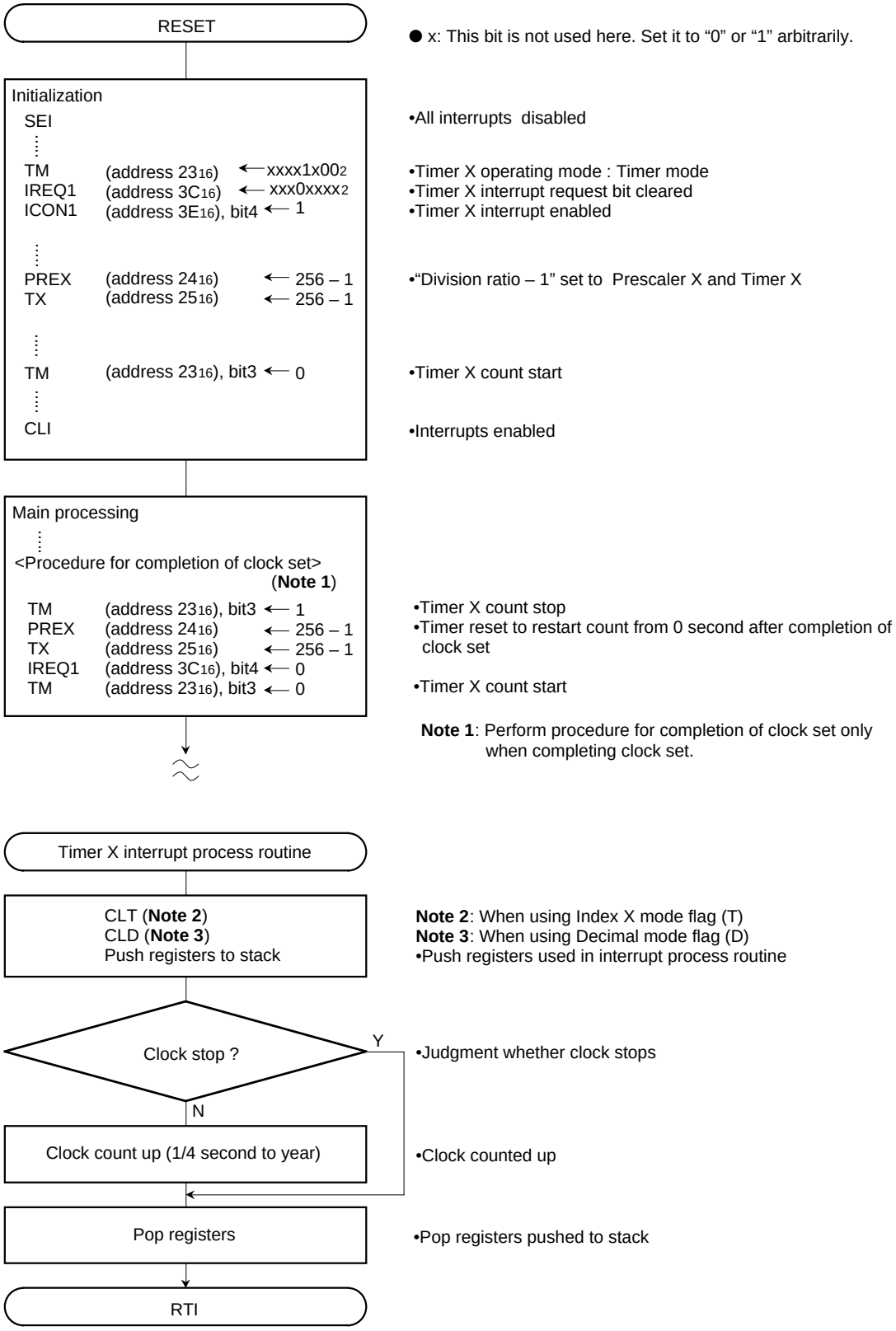


Fig. 2.3.13 Control procedure

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

Figure 2.3.14 shows a peripheral circuit example, and Figure 2.3.15 shows the timers connection and setting of division ratios. Figures 2.3.16 shows the relevant registers setting, and Figure 2.3.17 shows the control procedure.

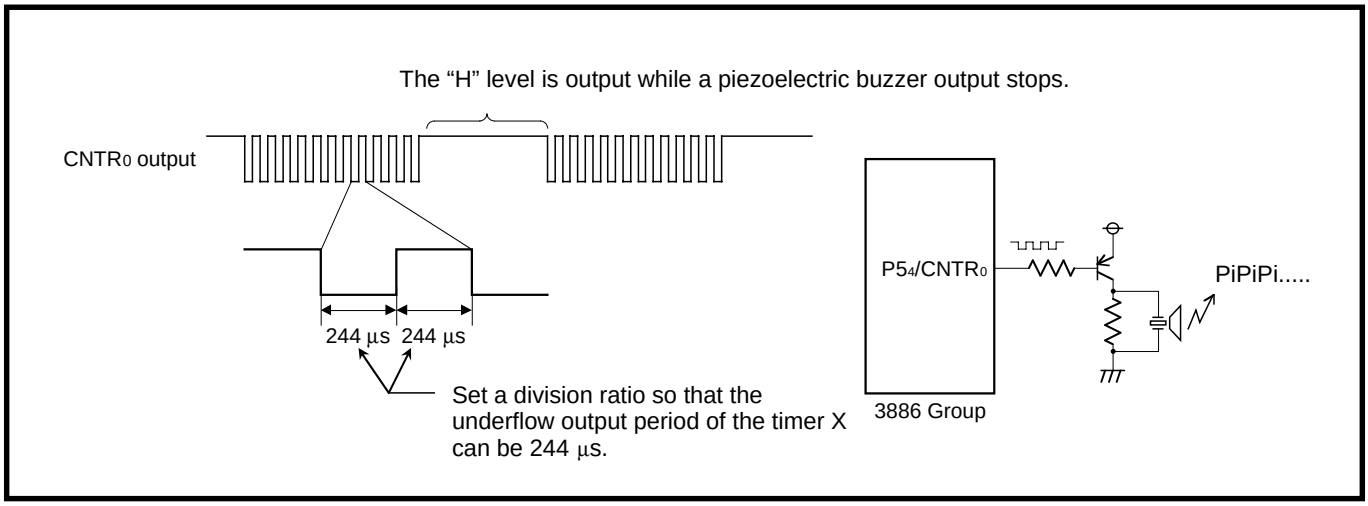


Fig. 2.3.14 Peripheral circuit example

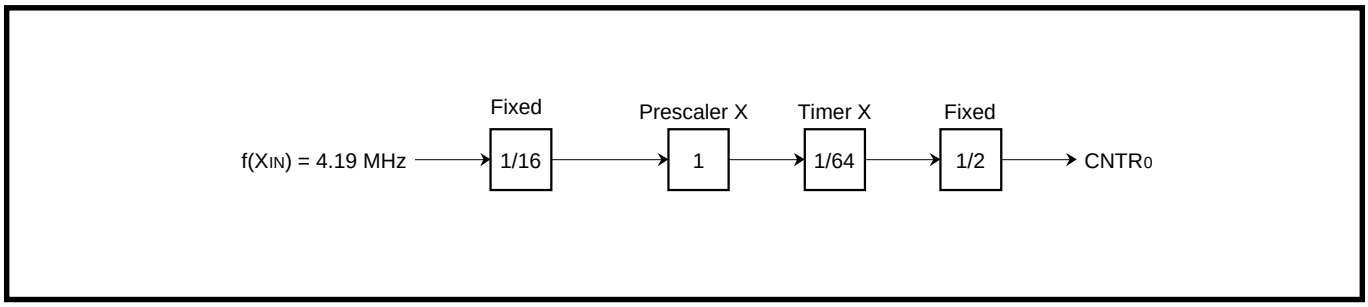


Fig. 2.3.15 Timers connection and setting of division ratios

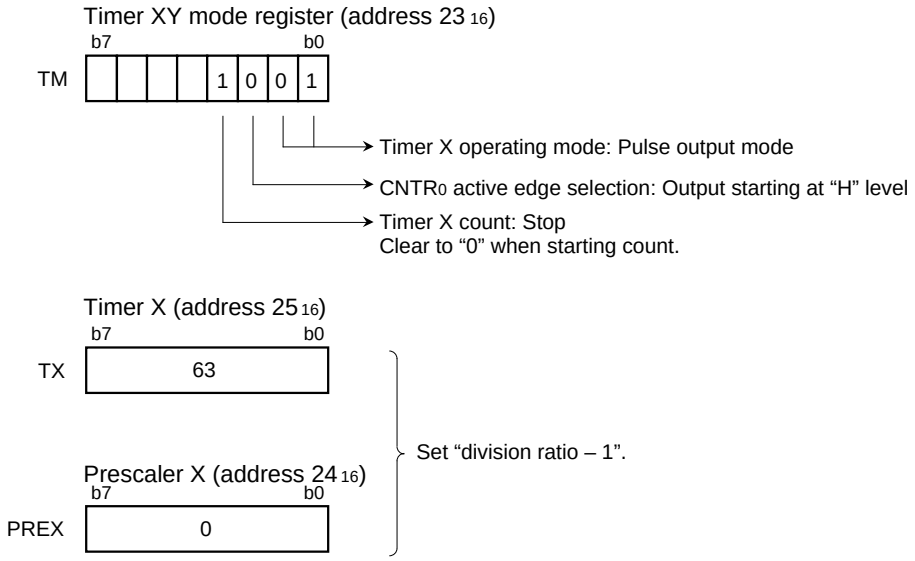


Fig. 2.3.16 Relevant registers setting

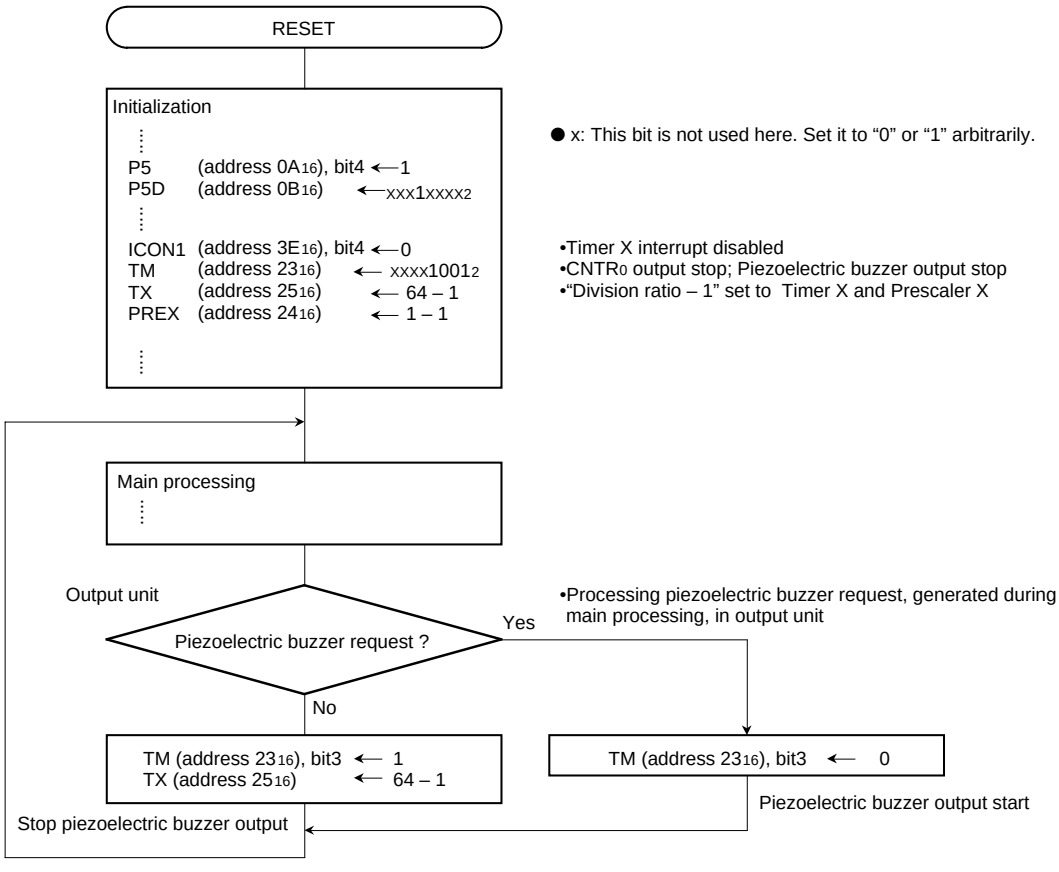


Fig. 2.3.17 Control procedure

(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 $P5_5/CNTR_1$ pin with the timer.
- •A reference value

Specifications: •The pulse is input to the $P5_5/CNTR_1$ pin and counted by the timer Y.

- •A count value 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 value.

Figure 2.3.18 shows the judgment method of valid/invalid of input pulses; Figure 2.3.19 shows the relevant registers setting; Figure 2.3.20 shows the control procedure.

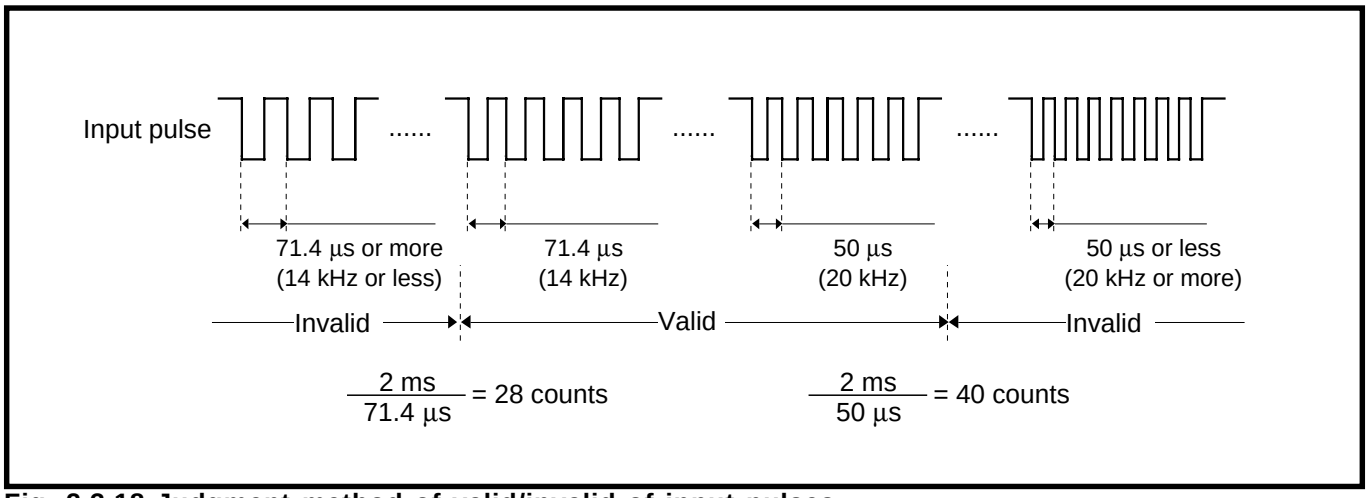


Fig. 2.3.18 Judgment method of valid/invalid of input pulses

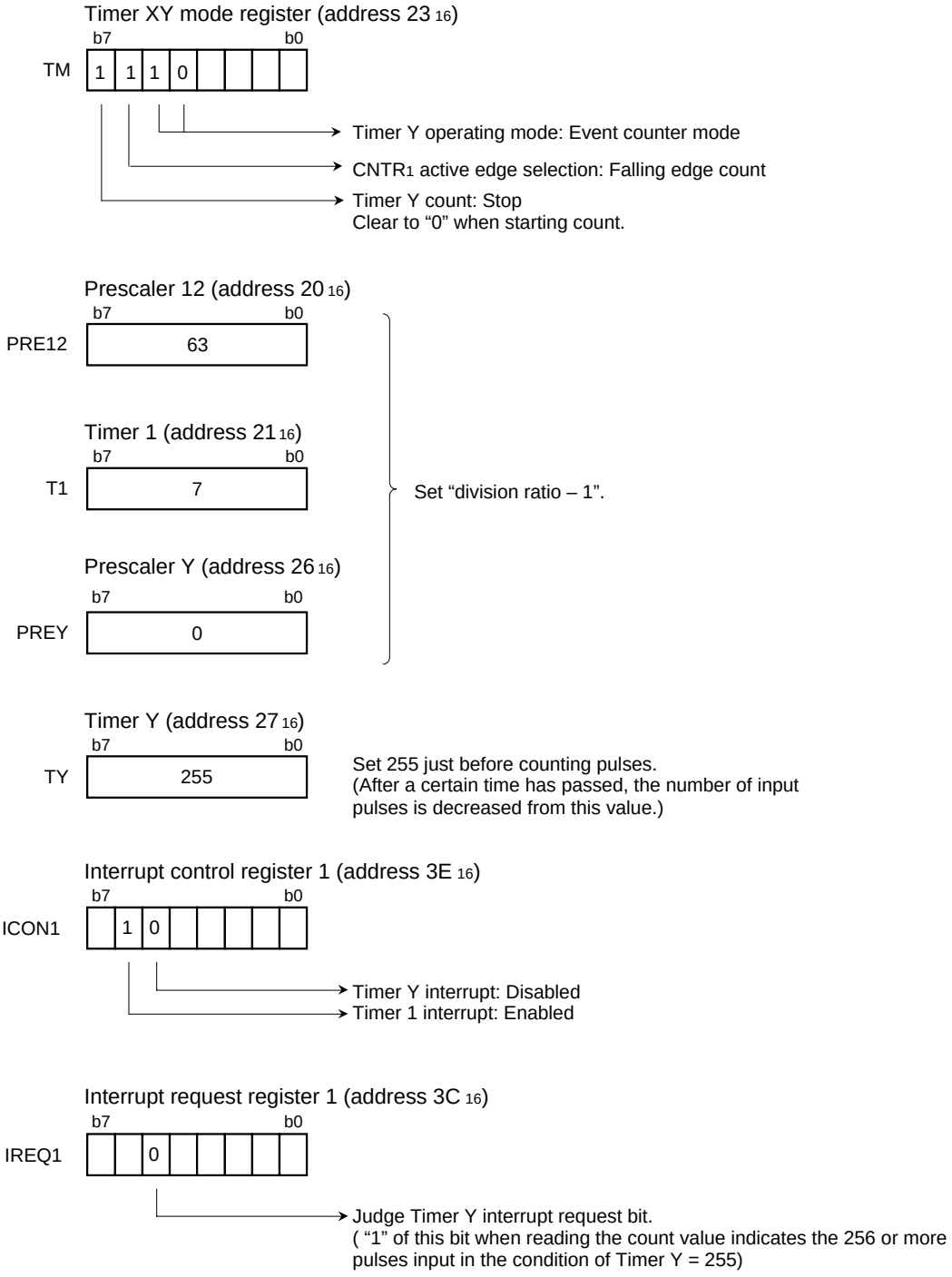


Fig. 2.3.19 Relevant registers setting

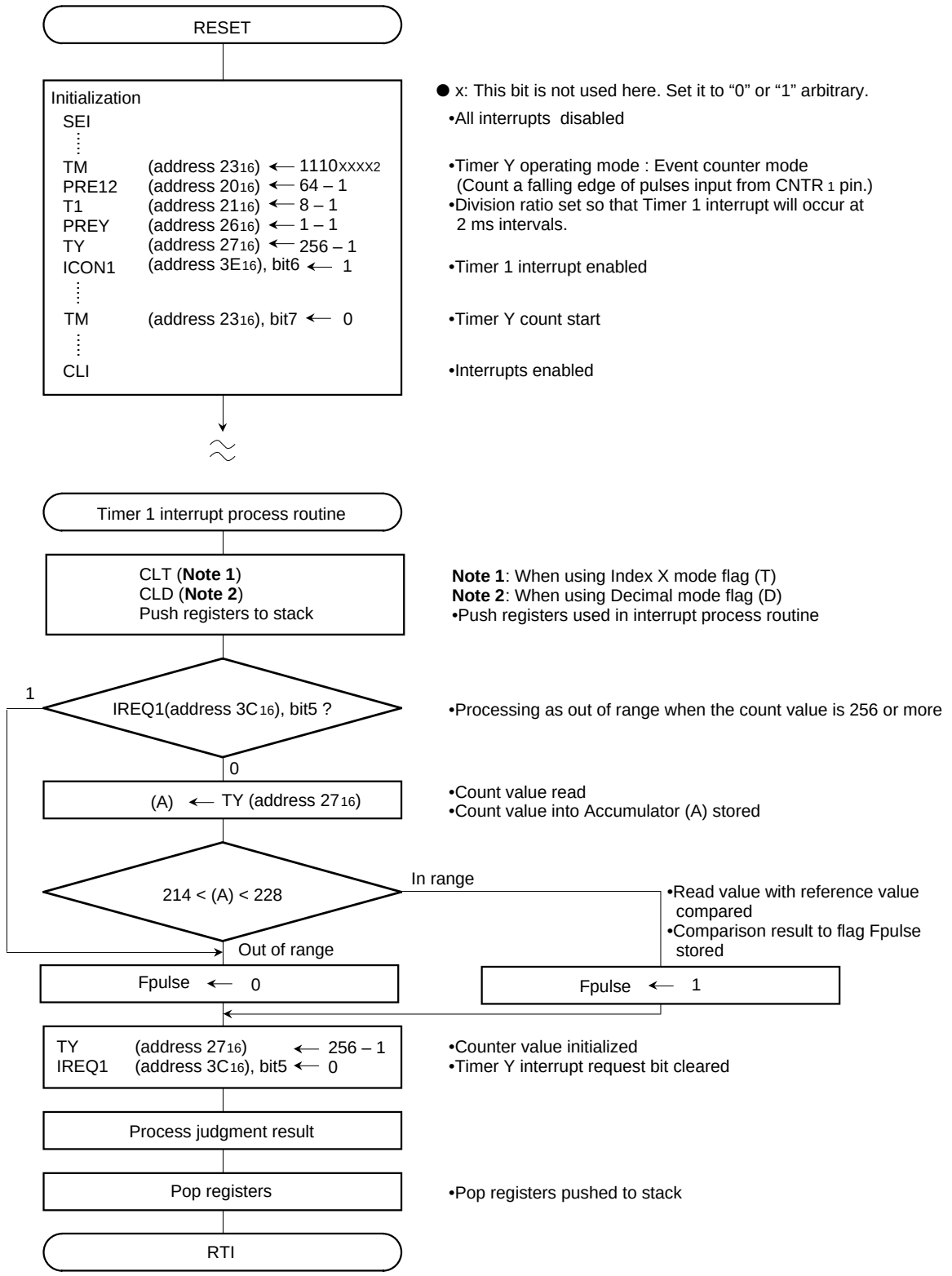


Fig. 2.3.20 Control procedure

(5) Timer application example 4: Measurement of FG pulse width for motor

Outline: The timer X counts the "H" level width of the pulses input to the $P5_4/CNTR_0$ pin. An underflow is detected by the timer X interrupt and an end of the input pulse "H" level is detected by the $CNTR_0$ interrupt.

Specifications: •The timer X counts the "H" level width of the FG pulse input to the $P5_4/CNTR_0$ pin.

<Example>

When the clock frequency is 4.19 MHz, the count source is 3.8 μs, which is obtained by dividing the clock frequency by 16. Measurement can be performed to 250 ms in the range of $FFFF_{16}$ to 0000_{16}.

Figure 2.3.21 shows the timers connection and setting of division ratio; Figure 2.3.22 shows the relevant registers setting; Figure 2.3.23 shows the control procedure.

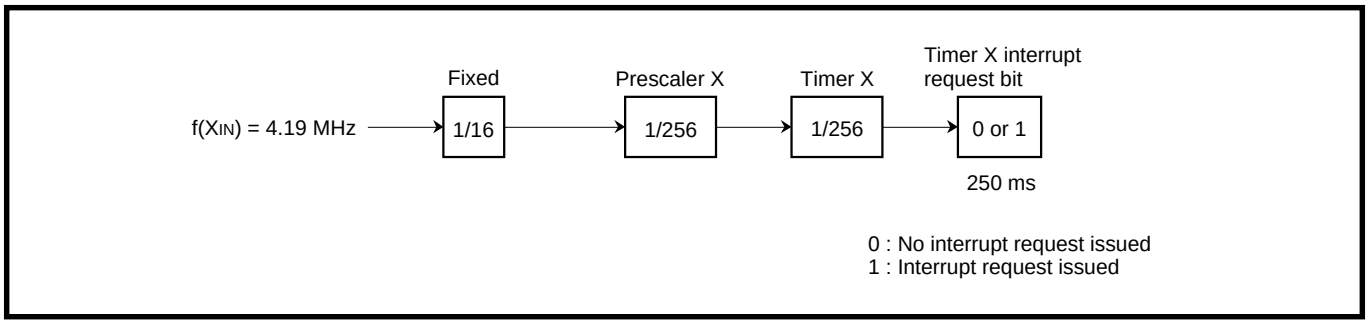


Fig. 2.3.21 Timers connection and setting of division ratios

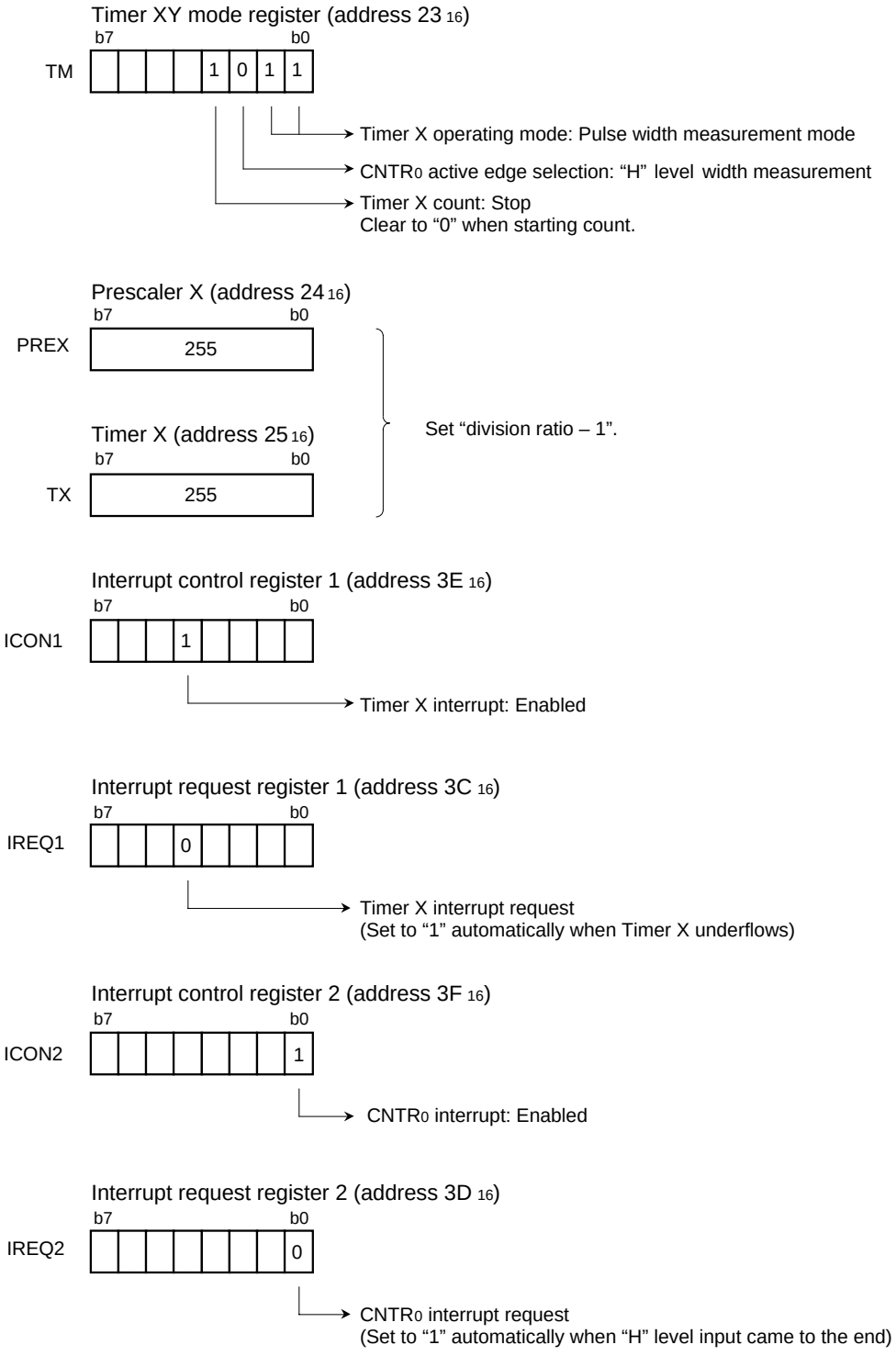
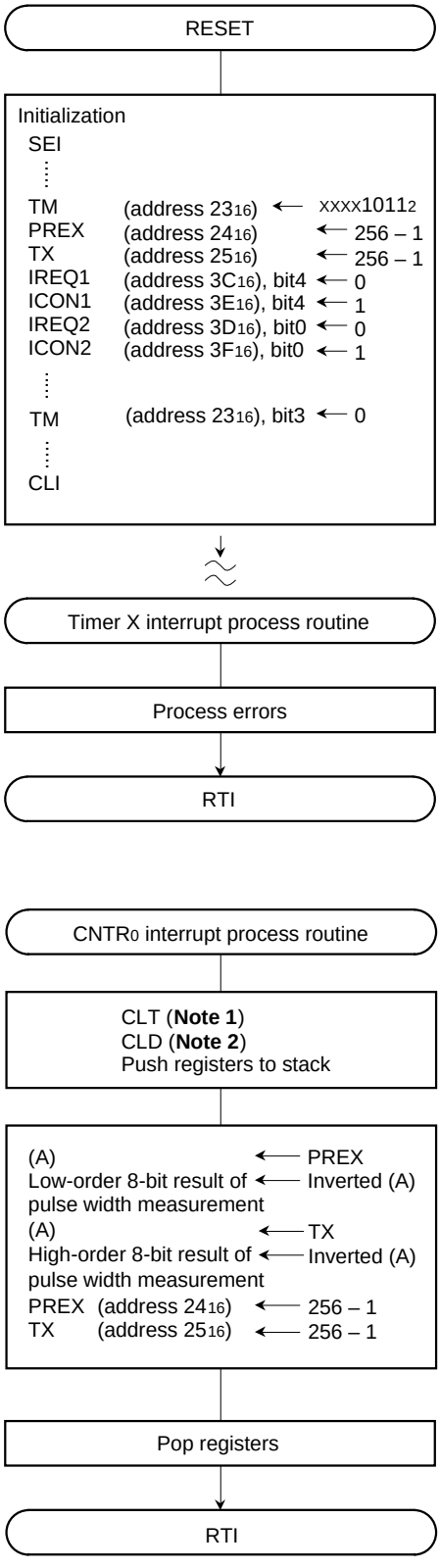


Fig. 2.3.22 Relevant registers setting



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

•All interrupts disabled

•Timer X operating mode : Pulse width measurement mode
(Measure “H” level of pulses input from CNTR 0 pin.)
•Set division ratio so that Timer X interrupt will occur at 250 ms intervals.
•Timer X interrupt request bit cleared
•Timer X interrupt enabled
•CNTR0 interrupt request bit cleared
•CNTR0 interrupt enabled

•Timer X count start

•Interrupts enabled

•Error occurs

Note 1: When using Index X mode flag (T)
Note 2: When using Decimal mode flag (D)
•Push registers used in interrupt process routine

•Read the count value and store it to RAM

•Division ratio set so that Timer X interrupt will occur at 250 ms intervals.

•Pop registers pushed to stack

Fig. 2.3.23 Control procedure

2.3.4 Notes on timer

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).
- When switching the count source by the timer Y count source selection bit, the value of timer count is altered in unconsiderable amount owing to generating of a thin pulses in the count input signals. Therefore, select the timer count source before set the value to the prescaler and the timer.

2.4 Serial I/O

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

2.4.1 Memory map

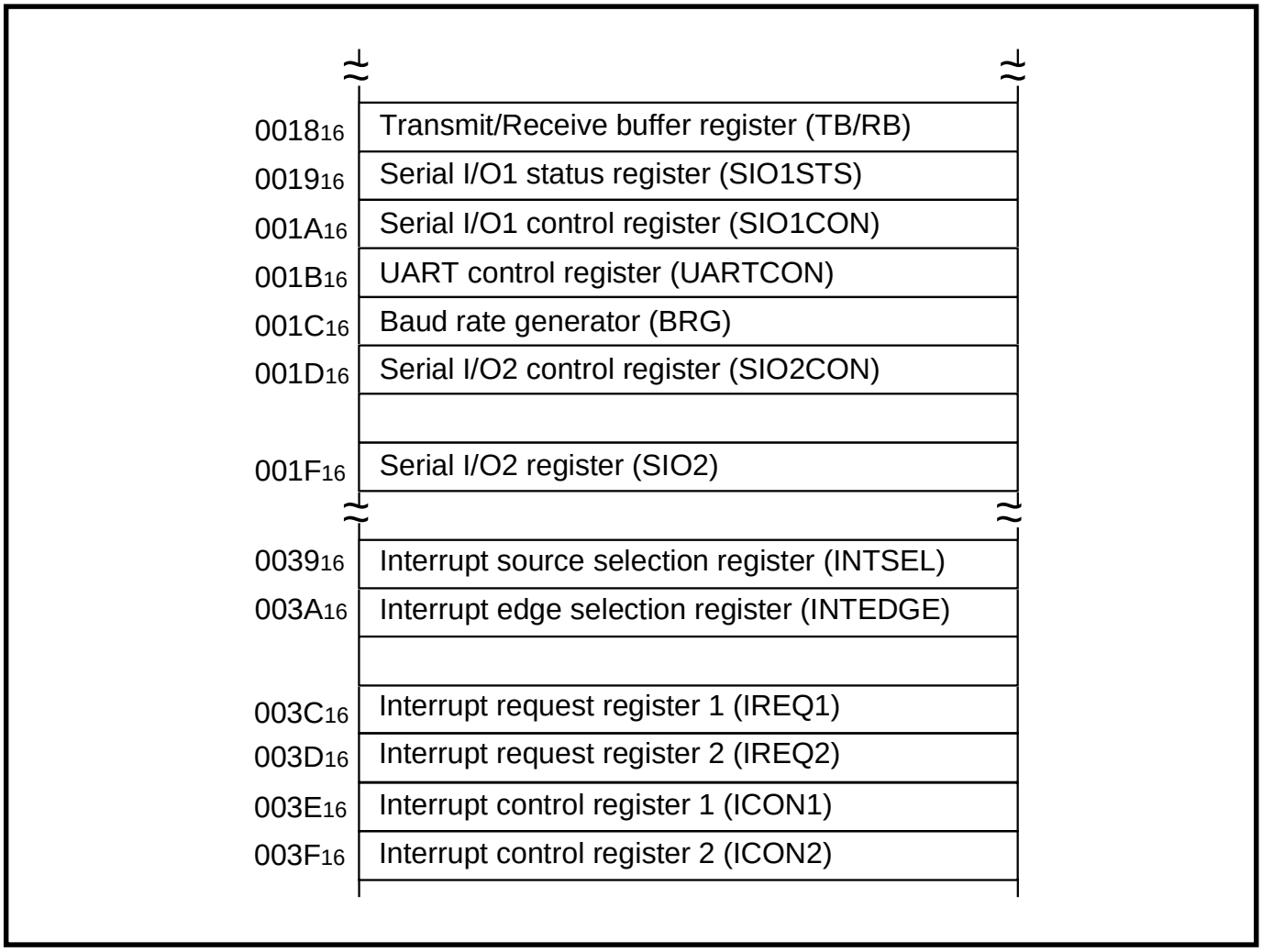


Fig. 2.4.1 Memory map of registers relevant to Serial I/O

2.4.2 Relevant registers

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

Transmit/Receive buffer register (TB/RB) [Address : 18_{16}]

B	Name	Function	At reset	R	W
0		The transmission data is written to or the receive data is read out from this buffer register. • At writing: A data is written to the transmit buffer register. • At reading: The contents of the receive buffer register are read out.	?	○	○
1			?	○	○
2			?	○	○
3			?	○	○
4			?	○	○
5			?	○	○
6			?	○	○
7			?	○	○

Note: The contents of transmit buffer register cannot be read out.
The data cannot be written to the receive buffer register.

Fig. 2.4.2 Structure of Transmit/Receive buffer register

Serial I/O1 status register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O1 status register (SIO1STS) [Address : 19_{16}]

B	Name	Function	At reset	R	W
0	Transmit buffer empty flag (TBE)	0 : Buffer full 1 : Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0 : Buffer empty 1 : Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0 : Transmit shift in progress 1 : Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0 : No error 1 : Overrun error	0	○	×
4	Parity error flag (PE)	0 : No error 1 : Parity error	0	○	×
5	Framing error flag (FE)	0 : No error 1 : Framing error	0	○	×
6	Summing error flag (SE)	0 : (OE) U (PE) U (FE) = 0 1 : (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "1".		1	○	×

Fig. 2.4.3 Structure of Serial I/O status register

Serial I/O1 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O1 control register (SIO1CON) [Address : $1A_{16}$]

B	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : f(X_{IN}) (f(X_{CIN}) in Low Speed Mode) 1 : f(X_{IN})/4 (f(X_{CIN})/4 in Low Speed Mode)	0	○	○
1	Serial I/O1 synchronous clock selection bit (SCS)	• In clock synchronous serial I/O 0 : BRG output divided by 4 1 : External clock input • In UART 0 : BRG output divided by 16 1 : External clock input divided by 16	0	○	○
2	$\overline{S_{RDY1}}$ output enable bit (SRDY)	0 : $P4_7$ pin operates as ordinary I/O pin 1 : $P4_7$ pin operates as $\overline{S_{RDY1}}$ output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0 : Interrupt when transmit buffer has emptied 1 : Interrupt when transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0 : Transmit disabled 1 : Transmit enabled	0	○	○
5	Receive enable bit (RE)	0 : Receive disabled 1 : Receive enabled	0	○	○
6	Serial I/O1 mode selection bit (SIOM)	0 : Clock asynchronous(UART) serial I/O 1 : Clock synchronous serial I/O	0	○	○
7	Serial I/O1 enable bit (SIOE)	0 : Serial I/O1 disabled (pins $P4_4$ to $P4_7$ operate as ordinary I/O pins) 1 : Serial I/O1 enabled (pins $P4_4$ to $P4_7$ operate as serial I/O pins)	0	○	○

Fig. 2.4.4 Structure of Serial I/O1 control register

UART control register

b7 b6 b5 b4 b3 b2 b1 b0

UART control register (UARTCON) [Address : $1B_{16}$]

B	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	○	○
1	Parity enable bit (PARE)	0 : Parity checking disabled 1 : Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	○	○
4	$P4_5$/TxD P-channel output disable bit (POFF)	In output mode 0 : CMOS output 1 : N-channel open-drain output	0	○	○
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "1".		1	○	×
6			1	○	×
7			1	○	×

Fig. 2.4.5 Structure of UART control register

Baud rate generator

b7 b6 b5 b4 b3 b2 b1 b0

Baud rate generator (BRG) [Address : $1C_{16}$]

B	Function	At reset	R	W
0	Set a count value of baud rate generator.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 2.4.6 Structure of Baud rate generator

Serial I/O2 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 control register (SIO2CON) [Address : $1D_{16}$]

B	Name	Function	At reset	R	W
0	Internal synchronous clock selection bits	b2 b1 b0 0 0 0: $f(X_{IN})/8$ 0 0 1: $f(X_{IN})/16$ 0 1 0: $f(X_{IN})/32$ 0 1 1: $f(X_{IN})/64$ 1 1 0: $f(X_{IN})/128$ 1 1 1: $f(X_{IN})/256$	0	○	○
1					
2					
3	Serial I/O2 port selection bit	0: I/O port ($P7_1$, $P7_2$) 1: S_{OUT2},S_{CLK2} signal output	0	○	○
4	$\overline{S_{RDY2}}$ output enable bit	0: I/O port ($P7_3$) 1: $\overline{S_{RDY2}}$ signal output	0	○	○
5	Transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
6	Serial I/O2 synchronous clock selection bit	0: External clock 1: Internal clock	0	○	○
7	Comparator reference input selection bit	0: $P0_0$/$P3_{REF}$ input 1: Reference input fixed	0	○	○

Fig. 2.4.7 Structure of Serial I/O2 control register

Serial I/O2 register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 register (SIO2) [Address : $1F_{16}$]

B	Name	Function	At reset	R	W
0		•Serial I/O2 register is the shift register for serial transfer. •At transmit: transmit data is set. •At receive: receive data is set.	?	○	○
1			?	○	○
2			?	○	○
3			?	○	○
4			?	○	○
5			?	○	○
6			?	○	○
7			?	○	○

Fig. 2.4.8 Structure of Serial I/O2 register

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register [INTSEL: address 0039_{16}]

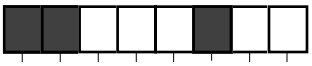
B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt source selection bit	0 : INT_0 interrupt 1 : Input buffer full interrupt	0	○	○
1	INT_1/output buffer empty interrupt source selection bit	0 : INT_1 interrupt 1 : Output buffer empty interrupt	0	○	○
2	Serial I/O1 transmit/S_{CL},S_{DA} interrupt source selection bit	0 : Serial I/O1 transmit interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
3	$CNTR_0$/S_{CL},S_{DA} interrupt source selection bit	0 : $CNTR_0$ interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
4	Serial I/O2/I^2C interrupt source selection bit	0 : Serial I/O2 interrupt ✻2 1 : I^2C interrupt	0	○	○
5	INT_2/I^2C interrupt source selection bit	0 : INT_2 interrupt ✻2 1 : I^2C interrupt	0	○	○
6	$CNTR_1$/key-on wake-up interrupt source selection bit	0 : $CNTR_1$ interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○
7	AD converter/key-on wake-up interrupt source selection bit	0 : A-D converter interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○

✻1: Do not write "1" to these bits simultaneously.
✻2: Do not write "1" to these bits simultaneously.
✻3: Do not write "1" to these bits simultaneously.

Fig. 2.4.9 Structure of Interrupt source selection register

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0



Interrupt edge selection register (INTEDGE) [Address : $3A_{16}$]

B	Name	Function	At reset	R	W
0	INT_0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT_1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is “0”.		0	○	×
3	INT_2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4	INT_3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
5	INT_4 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
6	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are “0”.		0	○	×
7			0	○	×

Fig. 2.4.10 Structure of Interrupt edge selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT_1/output buffer empty interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.4.11 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2 (IREQ2) [Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	$CNTR_1$/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT_4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.4.12 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	Function	At reset	R	W
0	INT_0/input buffer full interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1/output buffer empty interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.4.13 Structure of Interrupt control register 1

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 2 (ICON2) [Address : $3F_{16}$]

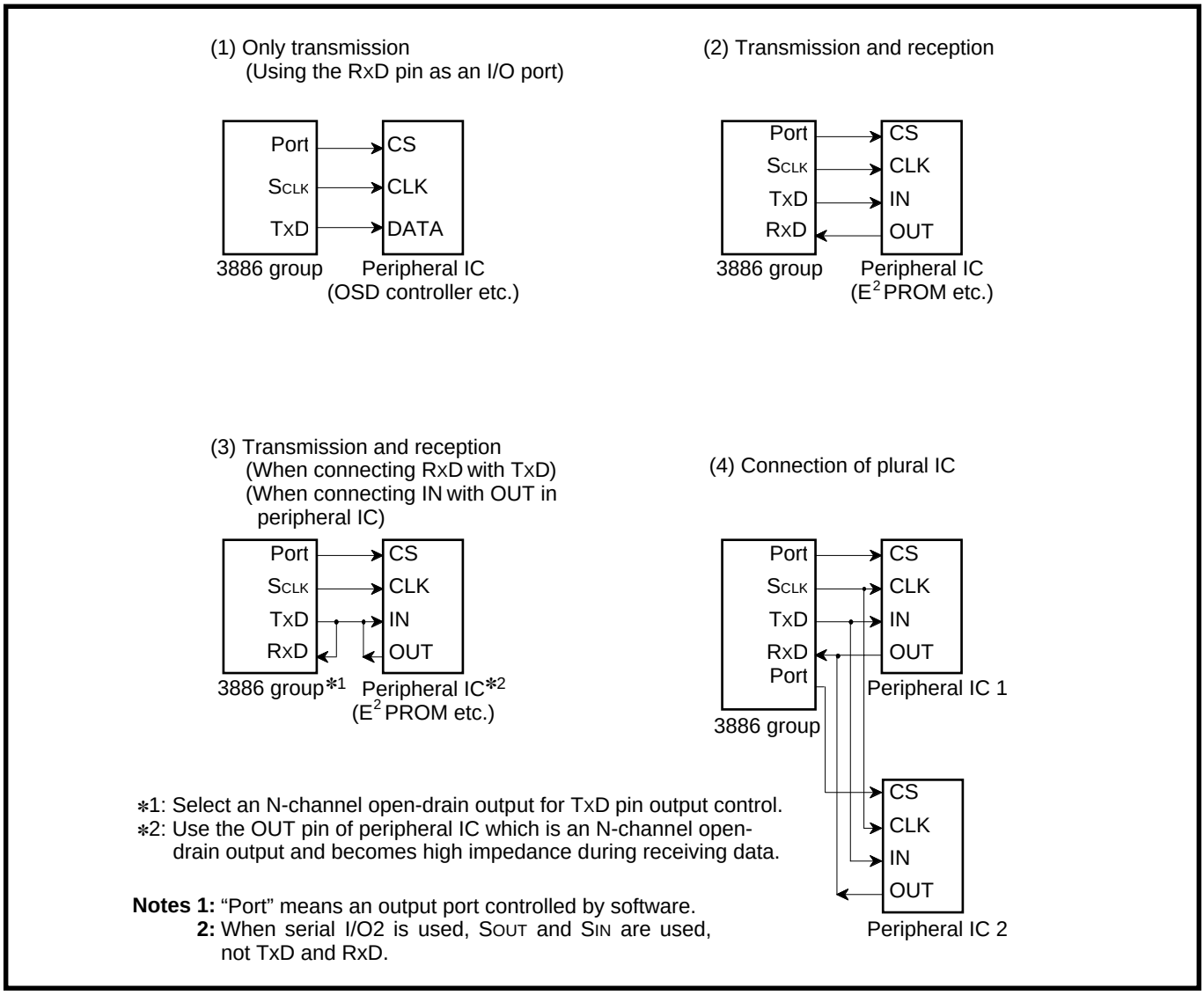
B	Name	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	$CNTR_1$/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	INT_4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.4.14 Structure of Interrupt control register 2

2.4.3 Serial I/O connection examples

(1) Control of peripheral IC equipped with CS pin

Figure 2.4.15 shows connection examples of a peripheral IC equipped with the CS pin. There are connection examples using a clock synchronous serial I/O mode.



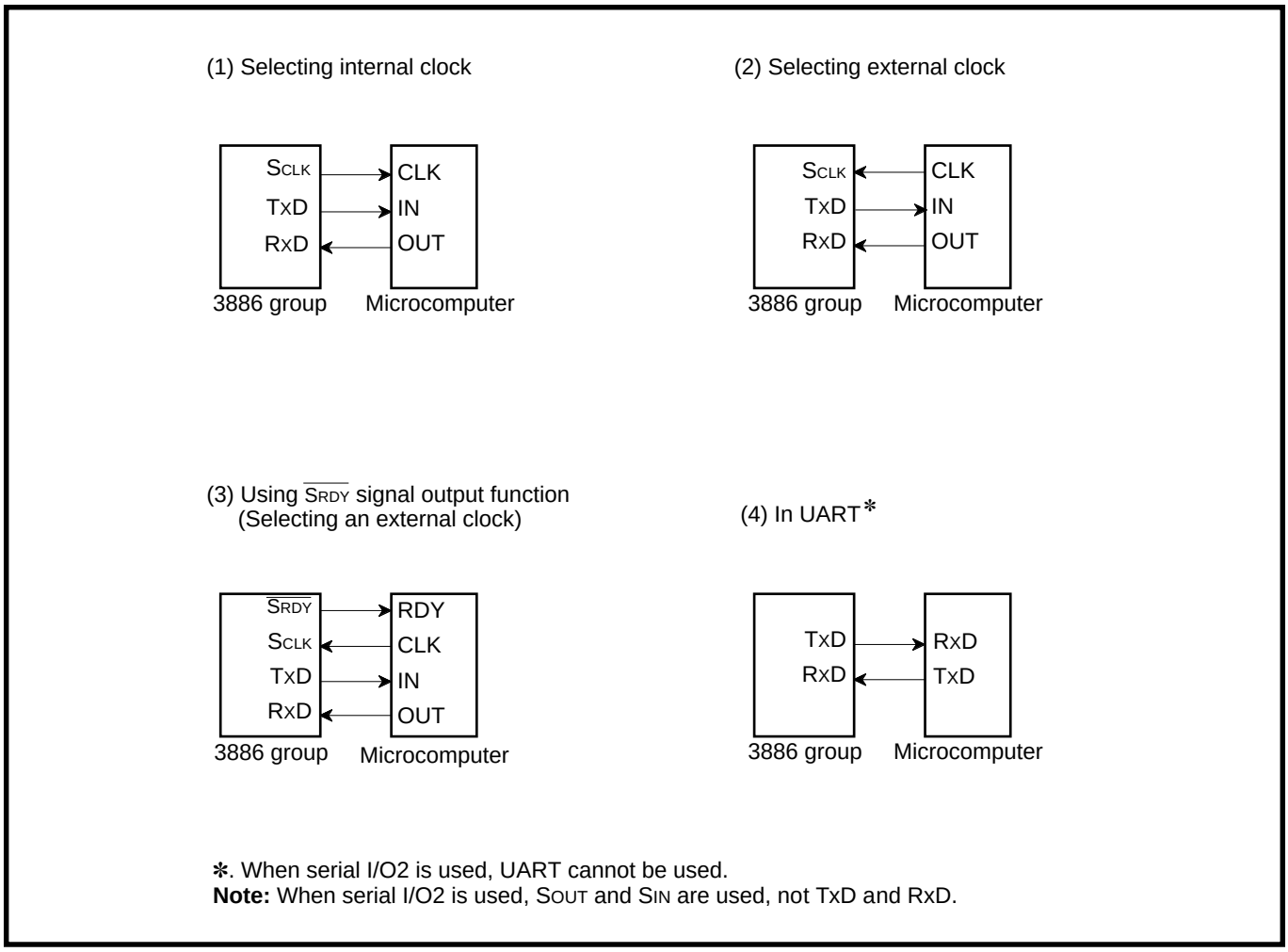
∗1: Select an N-channel open-drain output for TxD pin output control.
∗2: Use the OUT pin of peripheral IC which is an N-channel open-drain output and becomes high impedance during receiving data.

Notes 1: "Port" means an output port controlled by software.
2: When serial I/O2 is used, SOUT and SIN are used, not TxD and RxD.

Fig. 2.4.15 Serial I/O connection examples (1)

(2) Connection with microcomputer

Figure 2.4.16 shows connection examples with another microcomputer.



*. When serial I/O2 is used, UART cannot be used.

Note: When serial I/O2 is used, SOUT and SIN are used, not TxD and RxD.

Fig. 2.4.16 Serial I/O connection examples (2)

2.4.4 Setting of serial I/O transfer data format

A clock synchronous or clock asynchronous (UART) can be selected as a data format of Serial I/O1. A clock synchronous is used as a data format of Serial I/O2.

Figure 2.4.17 shows the serial I/O transfer data format.

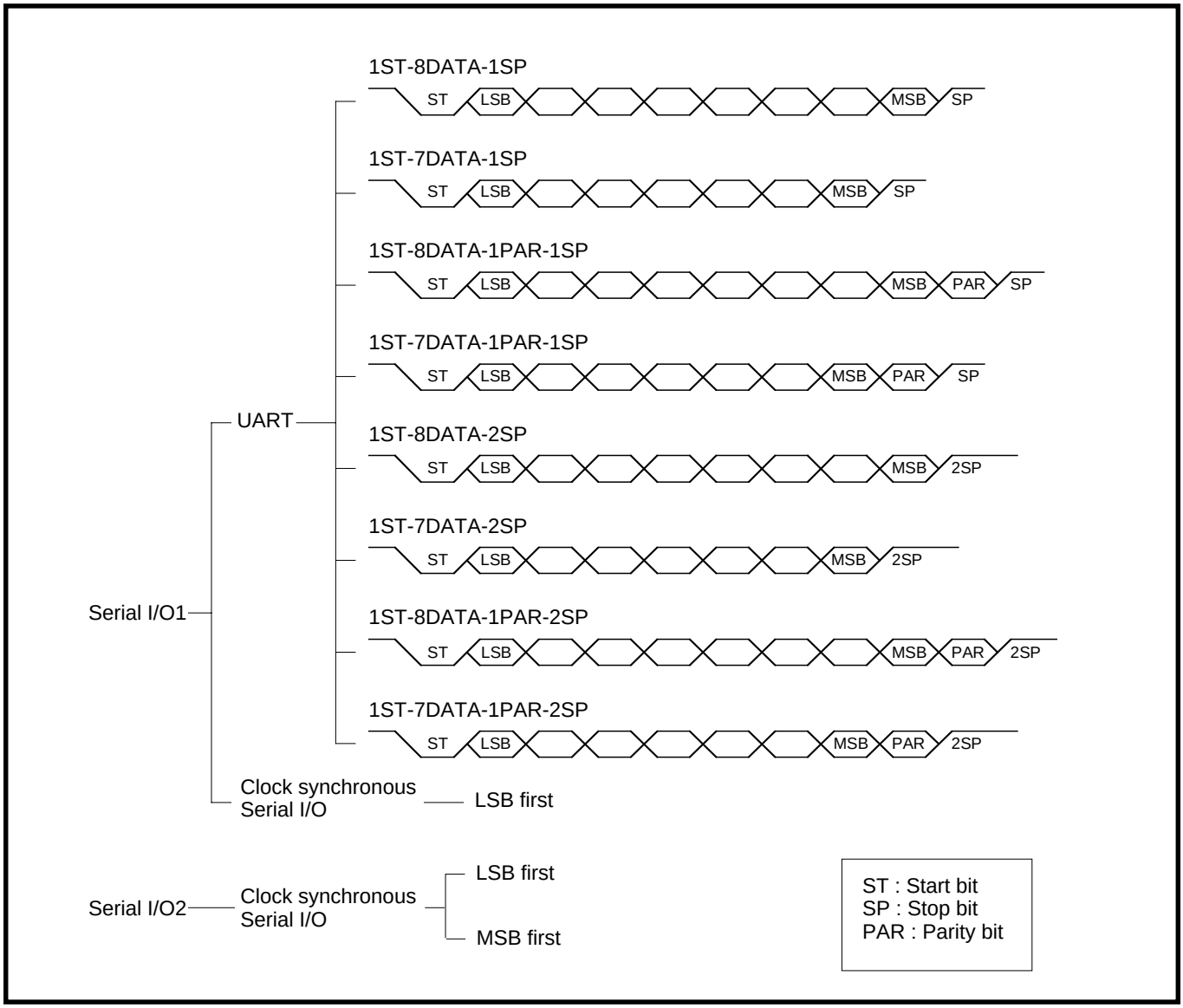


Fig. 2.4.17 Serial I/O transfer data format

2.4.5 Serial I/O application examples

(1) Communication using clock synchronous serial I/O (transmit/receive)

Outline : 2-byte data is transmitted and received, using the clock synchronous serial I/O. The $\overline{S_{RDY1}}$ signal is used for communication control.

Figure 2.4.18 shows a connection diagram, and Figure 2.4.19 shows a timing chart.
Figure 2.4.20 shows a registers setting relevant to the transmitting side, and Figure 2.4.21 shows registers setting relevant to the receiving side.

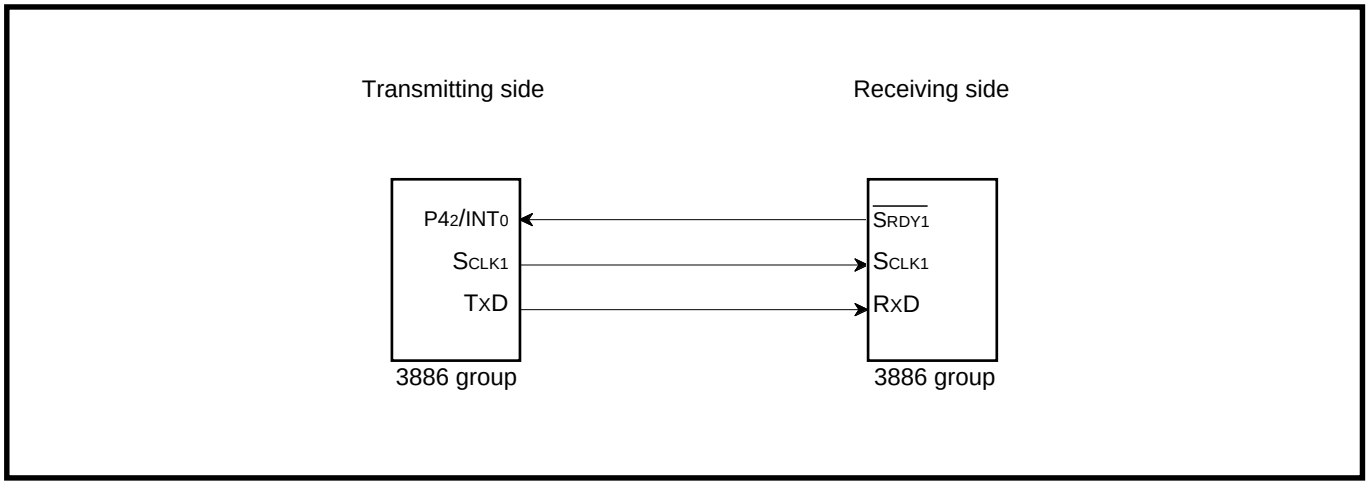


Fig. 2.4.18 Connection diagram

Specifications :
- The Serial I/O1 is used (clock synchronous serial I/O is selected.)
- Synchronous clock frequency : 125 kHz (f(X_{IN}) = 4 MHz is divided by 32)
- The $\overline{S_{RDY1}}$ (receivable signal) is used.
- The receiving side outputs the $\overline{S_{RDY1}}$ signal at intervals of 2 ms (generated by timer), and 2-byte data is transferred from the transmitting side to the receiving side.

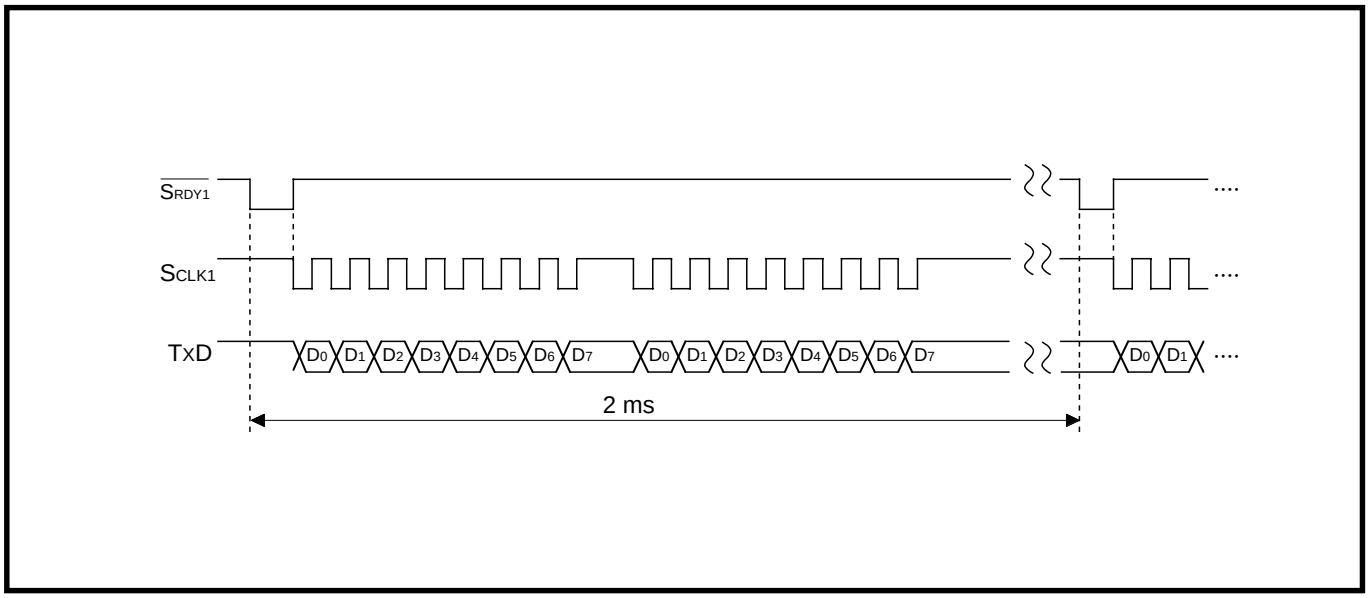


Fig. 2.4.19 Timing chart

Transmitting side

Serial I/O1 status register (Address : 19_{16})

SIO1STS

b7 b0

→ Transmit buffer empty flag
- Confirm that the data has been transferred from Transmit buffer register to Transmit shift register.
- When this flag is "1", it is possible to write the next transmission data in to Transmit buffer register.

→ Transmit shift register shift completion flag
Confirm completion of transmitting 1-byte data with this flag.
"1" : Transmit shift completed

Serial I/O1 control register (Address : $1A_{16}$)

SIO1CON

b7 b0

1	1	0	1			0	0

→ BRG counter source selection bit : f(X_{IN})
→ Serial I/O1 synchronous clock selection bit : BRG/4
→ Transmit enable bit : Transmit enabled
→ Receive enable bit : Receive disabled
→ Serial I/O1 mode selection bit : Clock synchronous serial I/O
→ Serial I/O1 enable bit : Serial I/O1 enabled

Baud rate generator (Address : $1C_{16}$)

BRG

b7 b0

7

Set "division ratio – 1".

Interrupt edge selection register (Address : $3A_{16}$)

INTEDGE

b7 b0

							0

→ INT_0 interrupt edge selection bit : Falling edge active

Fig. 2.4.20 Registers setting relevant to transmitting side

Receiving side

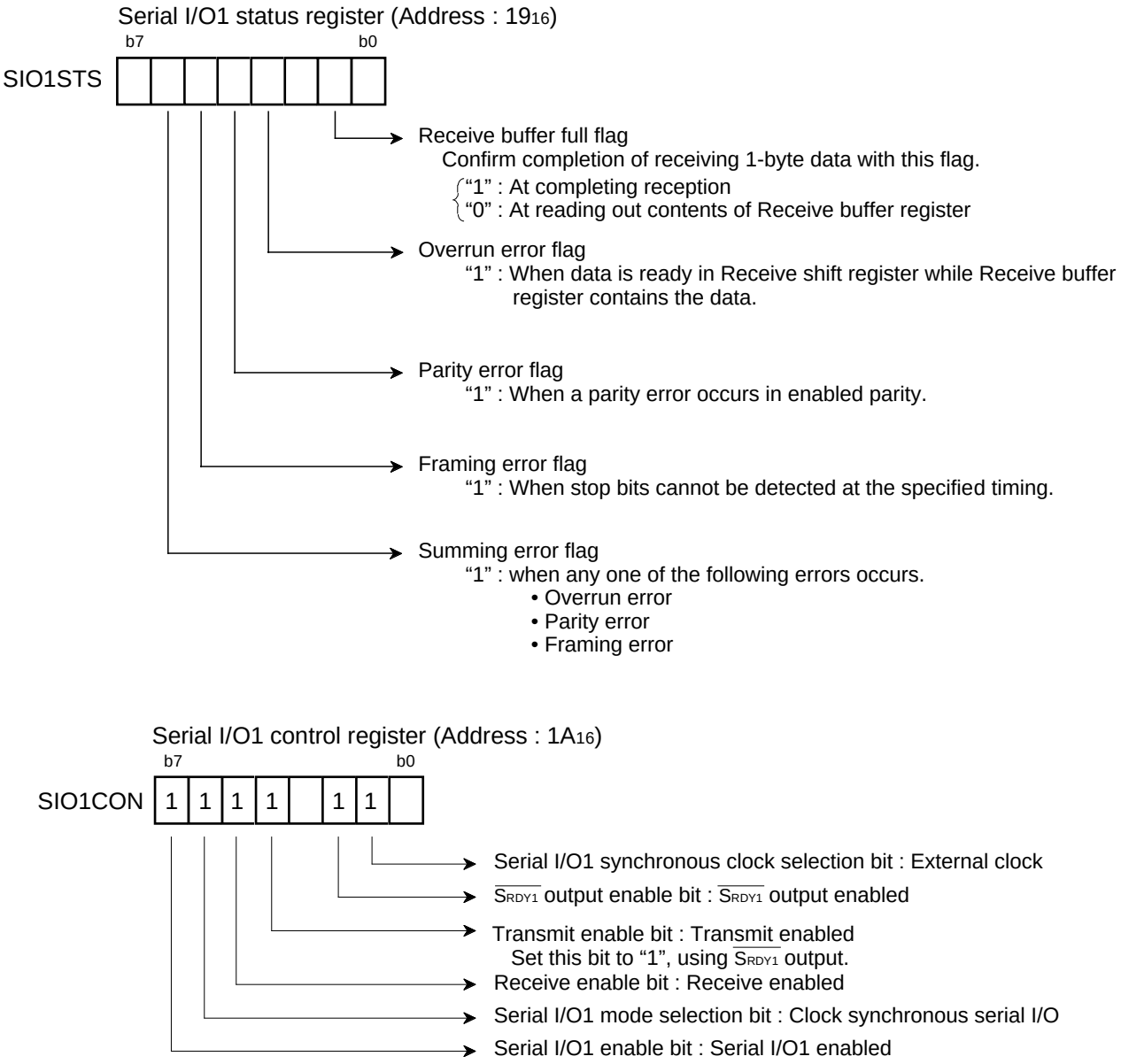


Fig. 2.4.21 Registers setting relevant to receiving side

Figure 2.4.22 shows a control procedure of the transmitting side, and Figure 2.4.23 shows a control procedure of the receiving side.

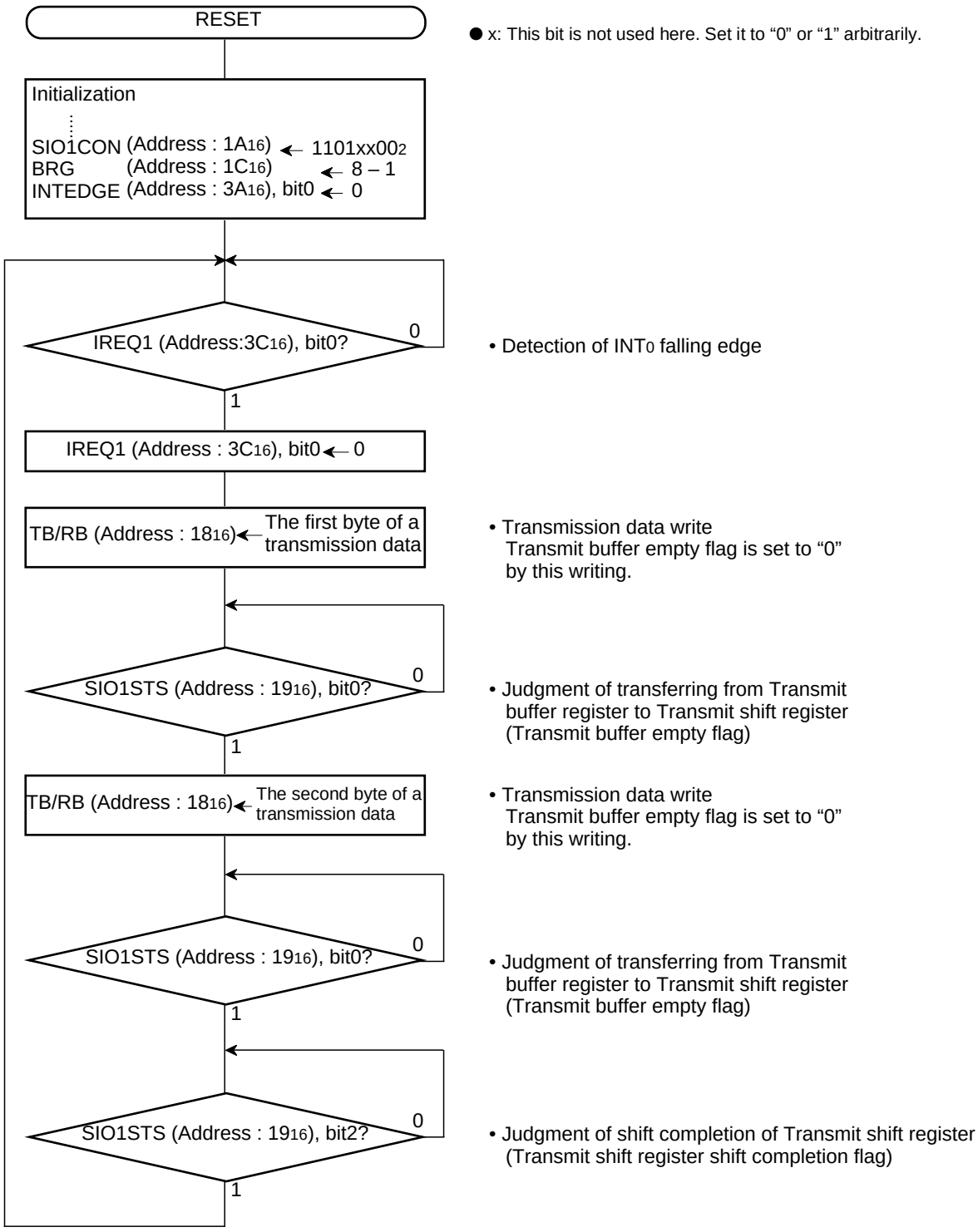


Fig. 2.4.22 Control procedure of transmitting side

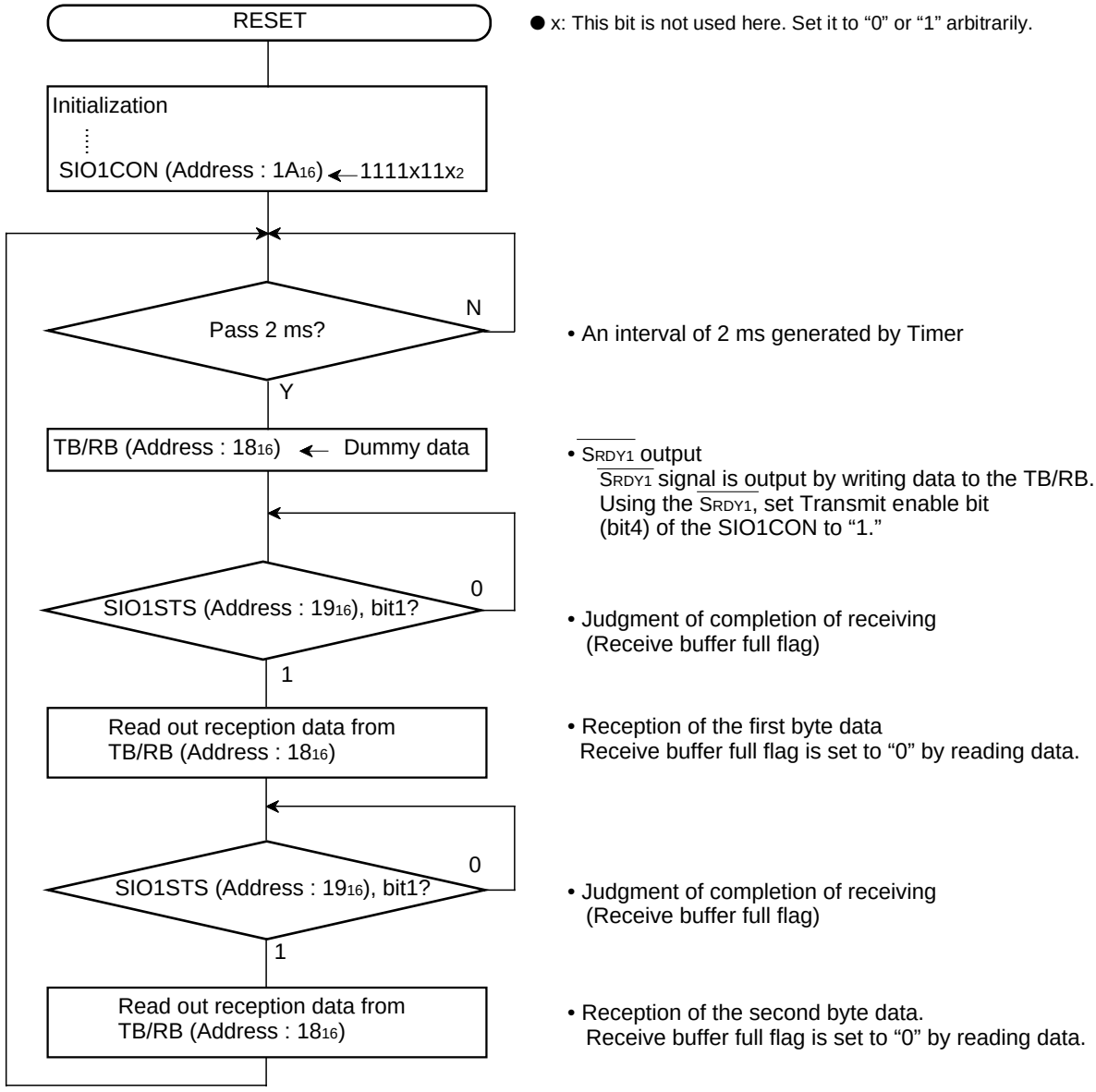


Fig. 2.4.23 Control procedure of receiving side

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

Outline : 4-byte data is transmitted and received, using the clock synchronous serial I/O. The CS signal is output to a peripheral IC through port $P5_3$.

The example for using Serial I/O1 and the example using for Serial I/O2 are shown. The specification of these examples are the same.
Figure 2.4.24 shows a connection diagram, and Figure 2.4.25 shows a timing chart.

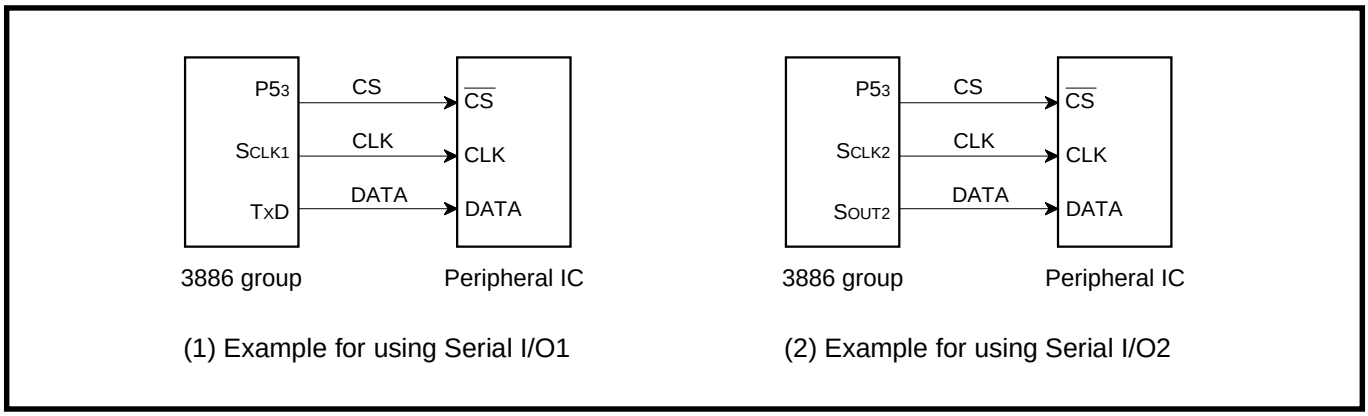
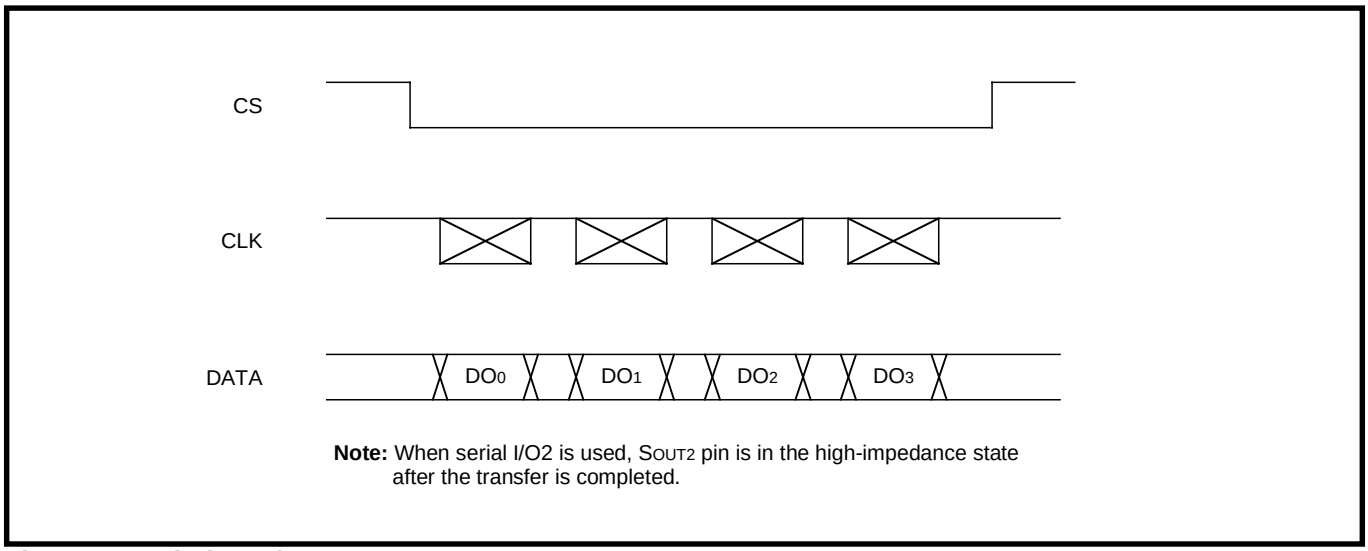


Fig. 2.4.24 Connection diagram

Specifications :
- The Serial I/O is used (clock synchronous serial I/O is selected.)
- Synchronous clock frequency : 125 kHz ($f(X_{IN})$ = 4 MHz is divided by 32)
- Transfer direction : LSB first
- The Serial I/O interrupt is not used.
- Port $P5_3$ is connected to the $\overline{CS}$ pin ("L" active) of the peripheral IC for transmission control; the output level of port $P5_3$ is controlled by software.



Note: When serial I/O2 is used, S_{OUT2} pin is in the high-impedance state after the transfer is completed.

Fig. 2.4.25 Timing chart

Figure 2.4.26 shows registers setting relevant to Serial I/O1, and Figure 2.4.27 shows a setting of serial I/O1 transmission data.

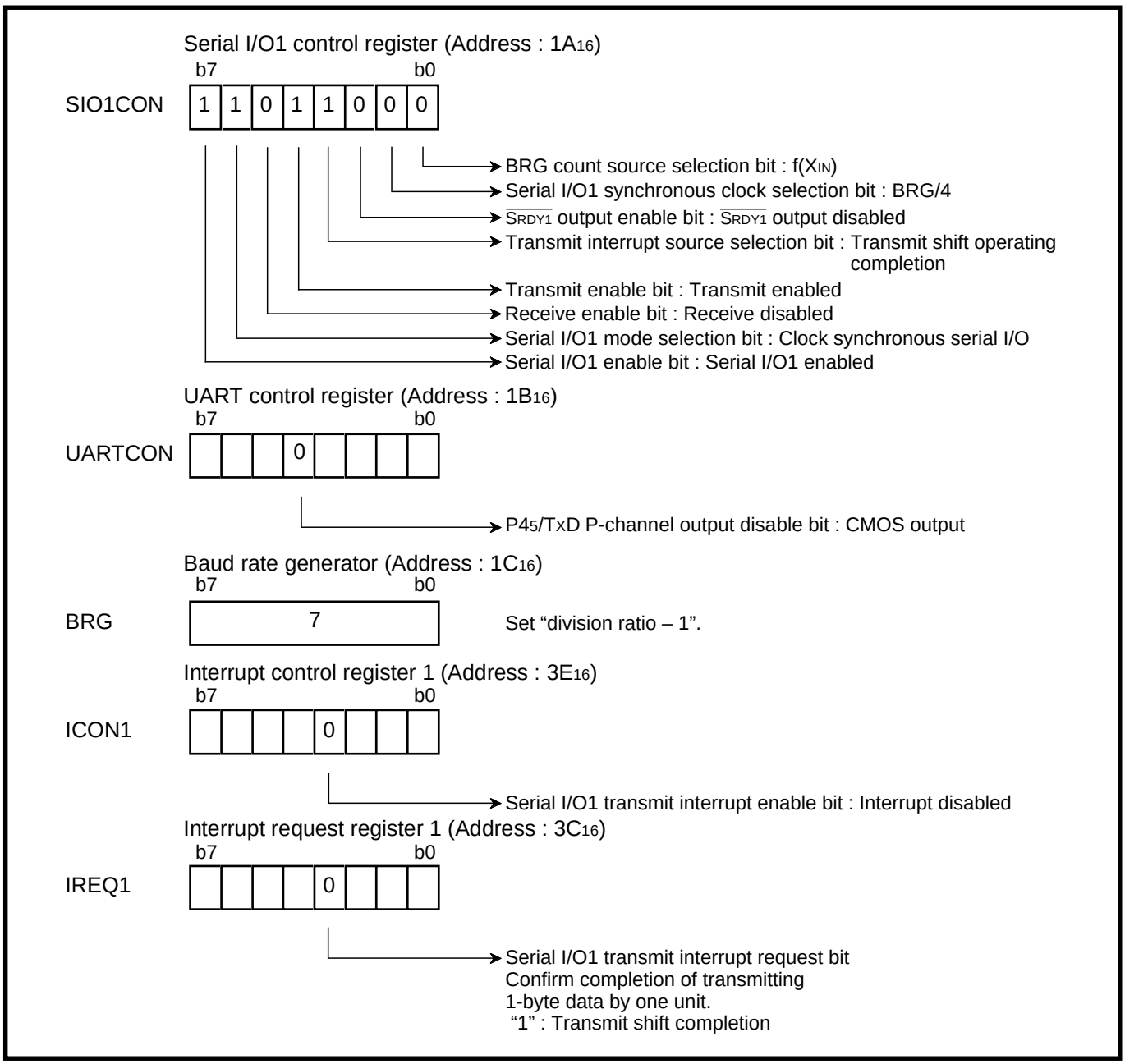


Fig. 2.4.26 Registers setting relevant to Serial I/O1

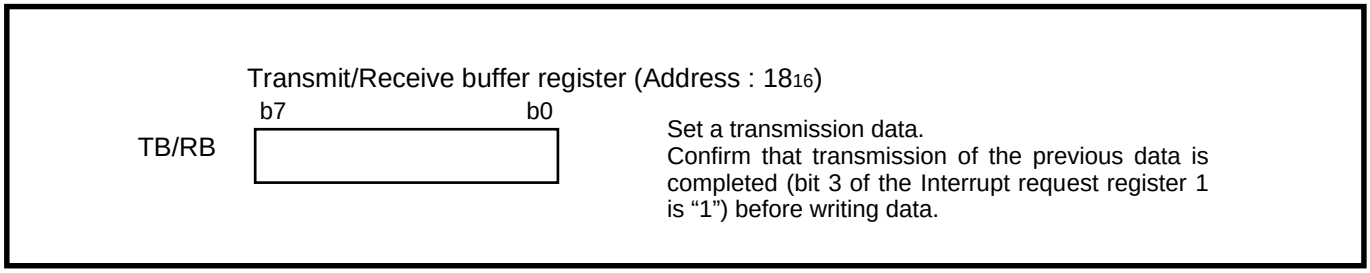


Fig. 2.4.27 Setting of serial I/O1 transmission data

When the registers are set as shown in Fig. 2.4.26, the Serial I/O1 can transmit 1-byte data by writing data to the transmit buffer register.
Thus, after setting the CS signal to "L", write the transmission data to the transmit buffer register by each 1 byte, and return the CS signal to "H" when the target number of bytes has been transmitted. Figure 2.4.28 shows a control procedure of Serial I/O1.

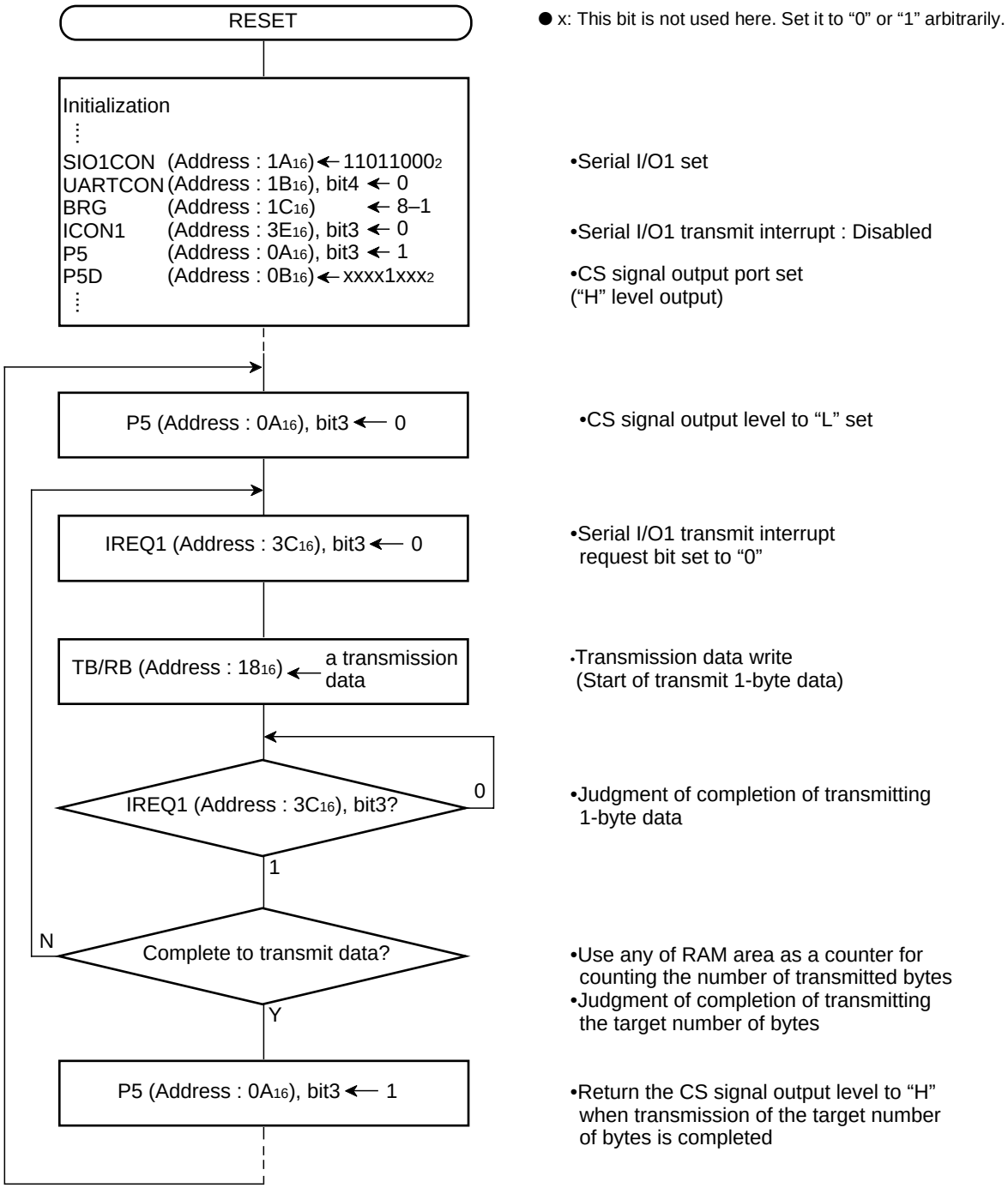


Fig. 2.4.28 Control procedure of Serial I/O1

Figure 2.4.29 shows registers setting relevant to Serial I/O2, and Figure 2.4.30 shows a setting of serial I/O2 transmission data.

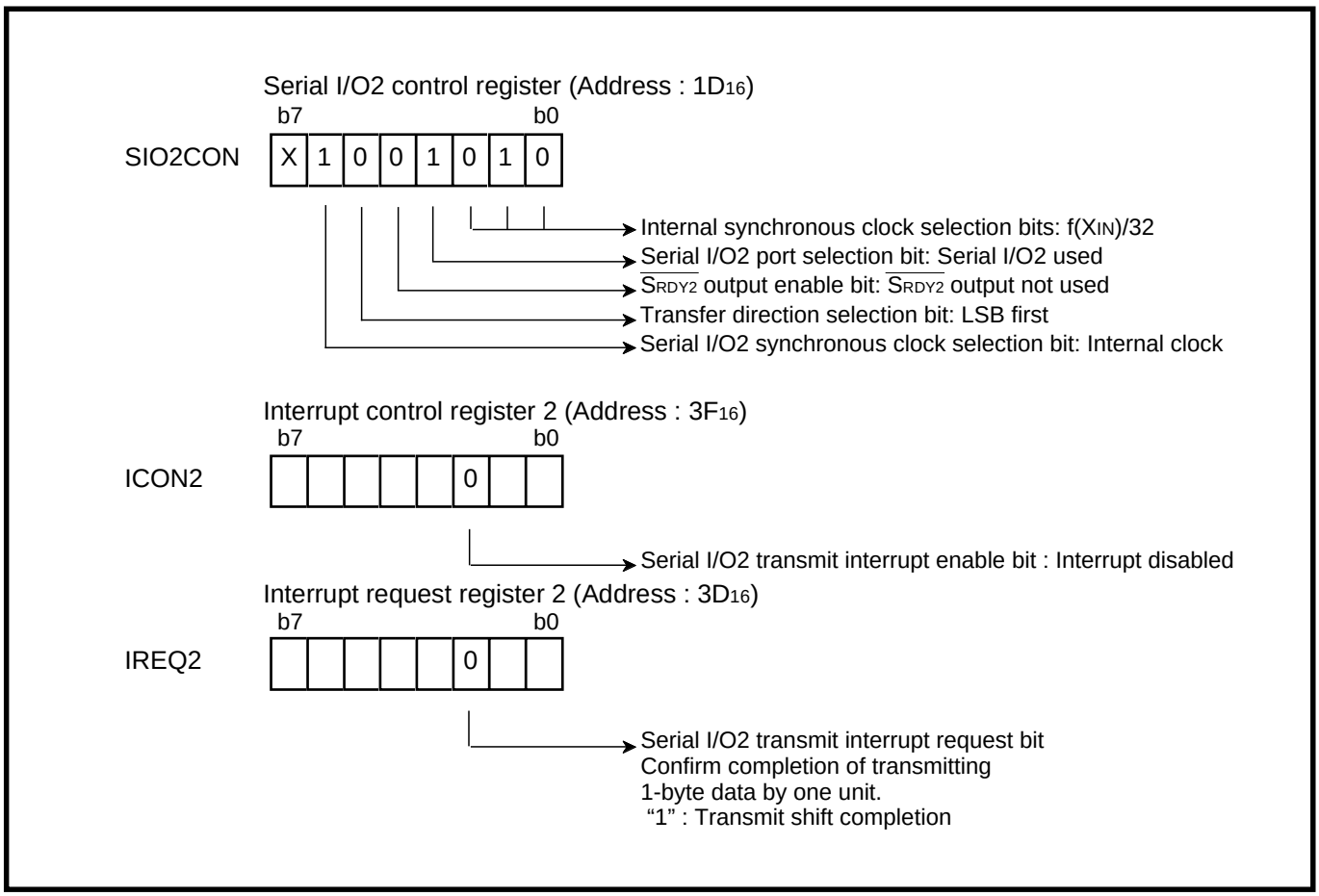


Fig. 2.4.29 Registers setting relevant to Serial I/O2

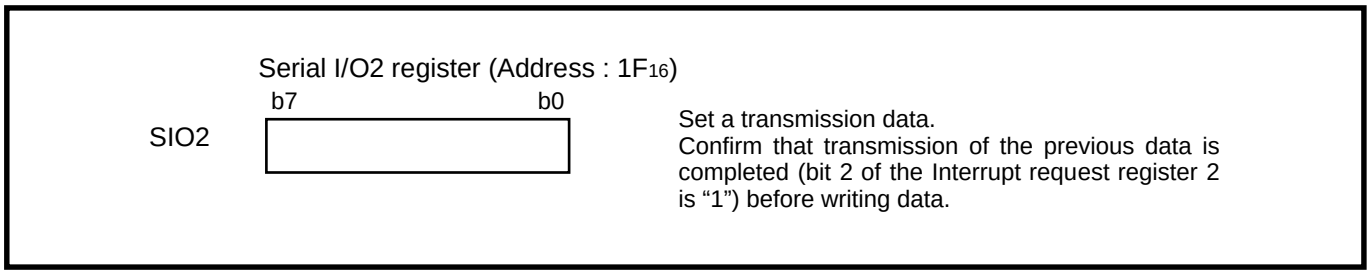


Fig. 2.4.30 Setting of serial I/O2 transmission data

When the registers are set as shown in Fig. 2.4.29, the Serial I/O2 can transmit 1-byte data by writing data to the serial I/O2 register.
Thus, after setting the CS signal to "L", write the transmission data to Serial I/O2 by each 1 byte, and return the CS signal to "H" when the target number of bytes has been transmitted.
Figure 2.4.31 shows a control procedure of Serial I/O2.

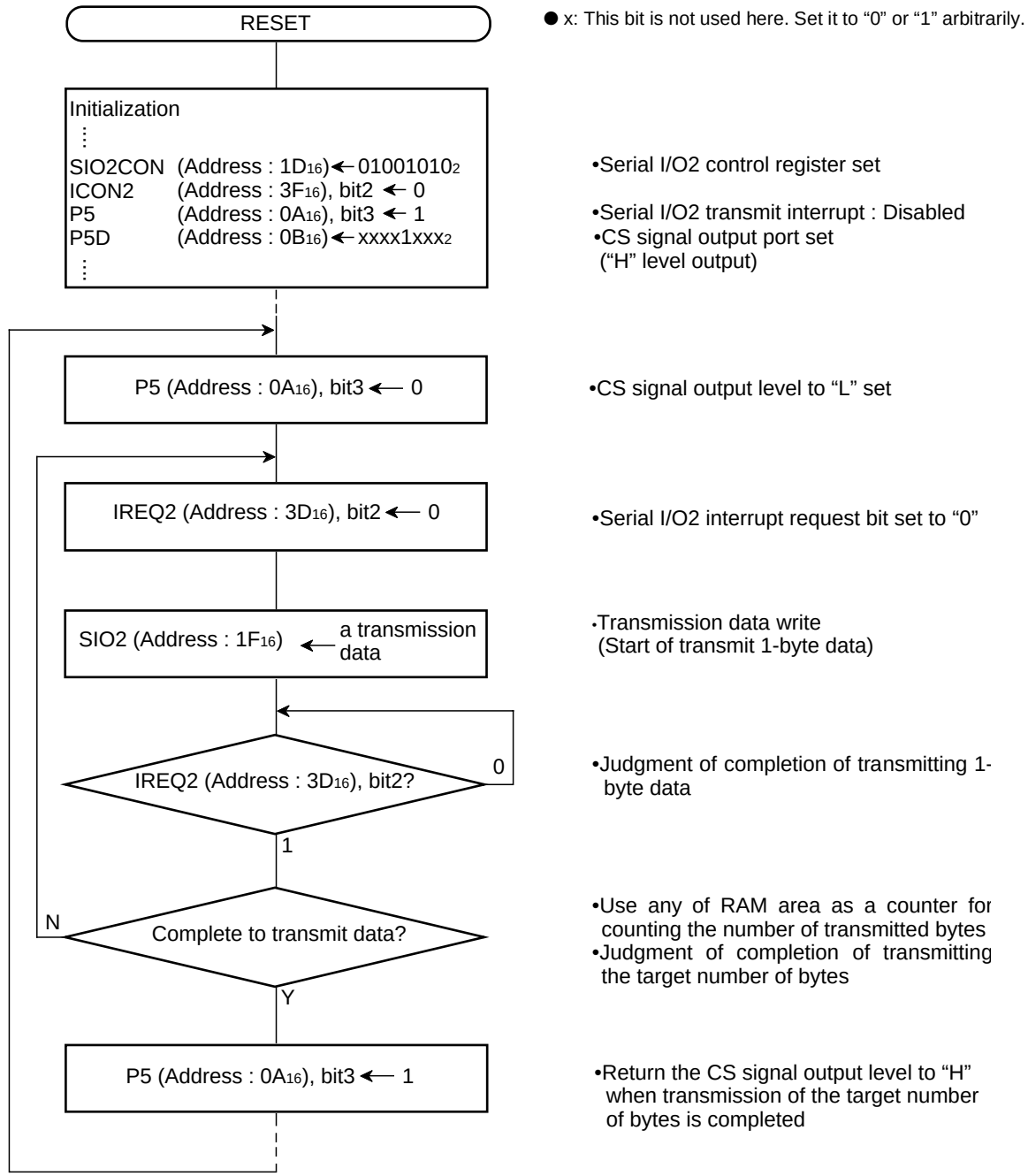


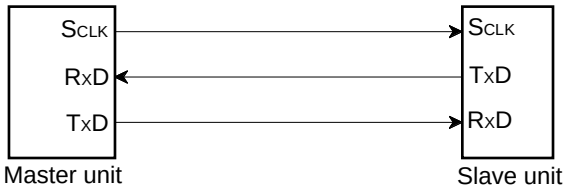
Fig. 2.4.31 Control procedure of Serial I/O2

(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 2.4.32 shows a connection diagram.



Note: When serial I/O2 is used, SOUT and SIN are used, not TxD and RxD.

Fig. 2.4.32 Connection diagram

Specifications :

- The serial I/O is used (clock synchronous serial I/O is selected).
- 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 byte/block
- Block transfer cycle : 16 ms
- Block transfer term : 3.5 ms
- Interval between blocks : 12.5 ms
- Heading adjustment time : 8 ms

Limitations of 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/O1 receive interrupt request to input of the next synchronous clock is 431 μs).
- "Heading adjustment time < interval between blocks" must be satisfied.

2.4 Serial I/O

The communication is performed according to the timing shown in Figure 2.4.33. In the slave unit, when a synchronous clock is not input within a certain time (heading adjustment time), the next clock input is processed as the beginning (heading) of a block.
When a clock is input again after one block (8 byte) is received, the clock is ignored.
Figure 2.4.34 shows relevant registers setting.

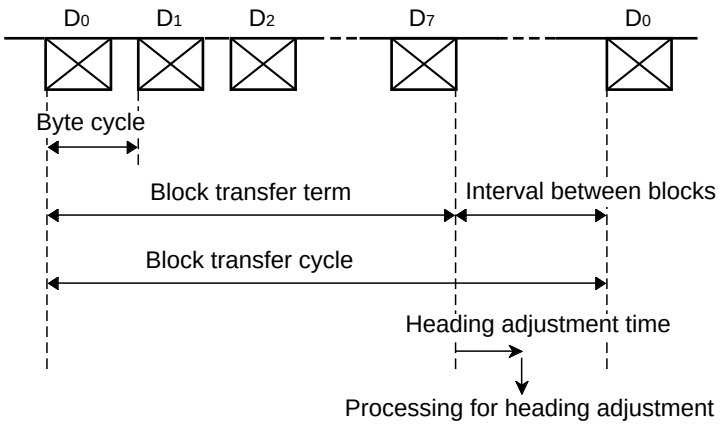


Fig. 2.4.33 Timing chart

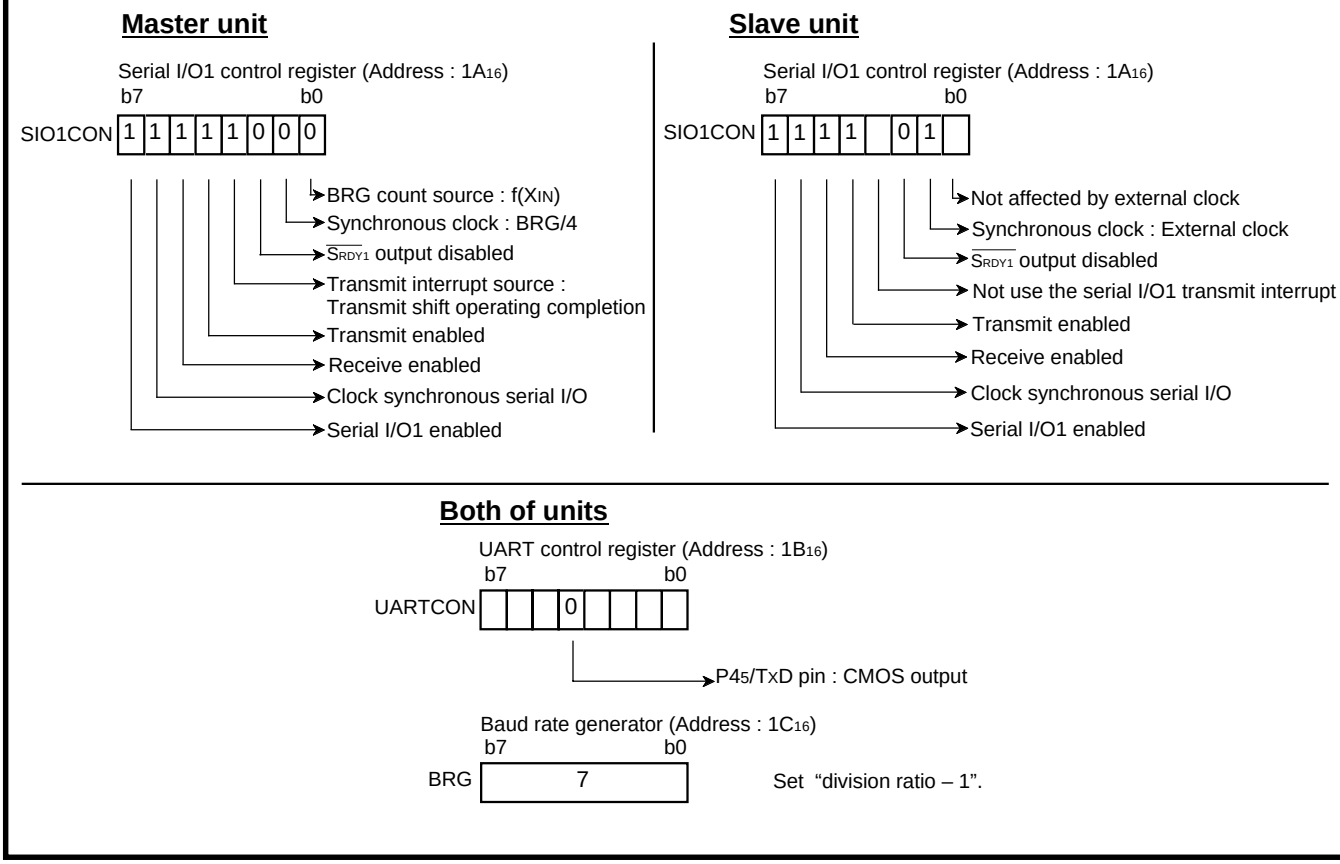


Fig. 2.4.34 Relevant registers setting

Control procedure :

● Control in the master unit

After setting the relevant registers shown in Figure 2.4.34, the master unit starts transmission or reception of 1-byte data by writing transmission data to the transmit buffer register.

To perform the communication in the timing shown in Figure 2.4.33, take the timing into account and write transmission data. Additionally, read out the reception data when the serial I/O1 transmit interrupt request bit is set to "1," or before the next transmission data is written to the transmit buffer register.

Figure 2.4.35 shows a control procedure of the master unit using timer interrupts.

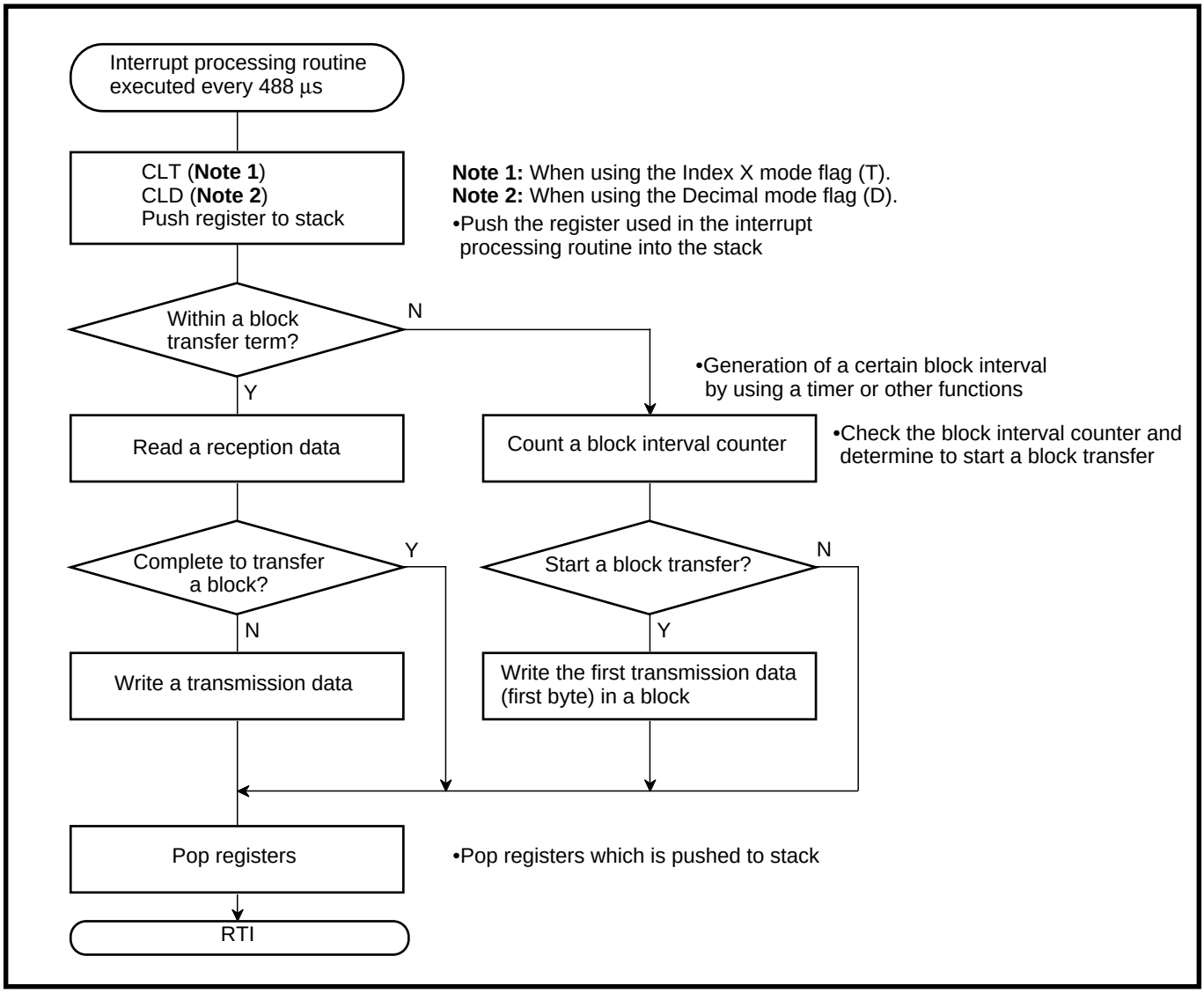


Fig. 2.4.35 Control procedure of master unit

● Control in the slave unit

After setting the relevant registers as shown in Figure 2.4.34, the slave unit becomes the state where a synchronous clock can be received at any time, and the serial I/O1 receive interrupt request bit is set to "1" each time an 8-bit synchronous clock is received.

In the serial I/O1 receive interrupt processing routine, the data to be transmitted next is written to the transmit buffer register after the received data is read out.

However, if no serial I/O1 receive 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 transmit buffer register.
2. The data to be received next is processed as the first 1 byte of the received data in the block.

Figure 2.4.36 shows a control procedure of the slave unit using the serial I/O1 receive interrupt and any timer interrupt (for heading adjustment).

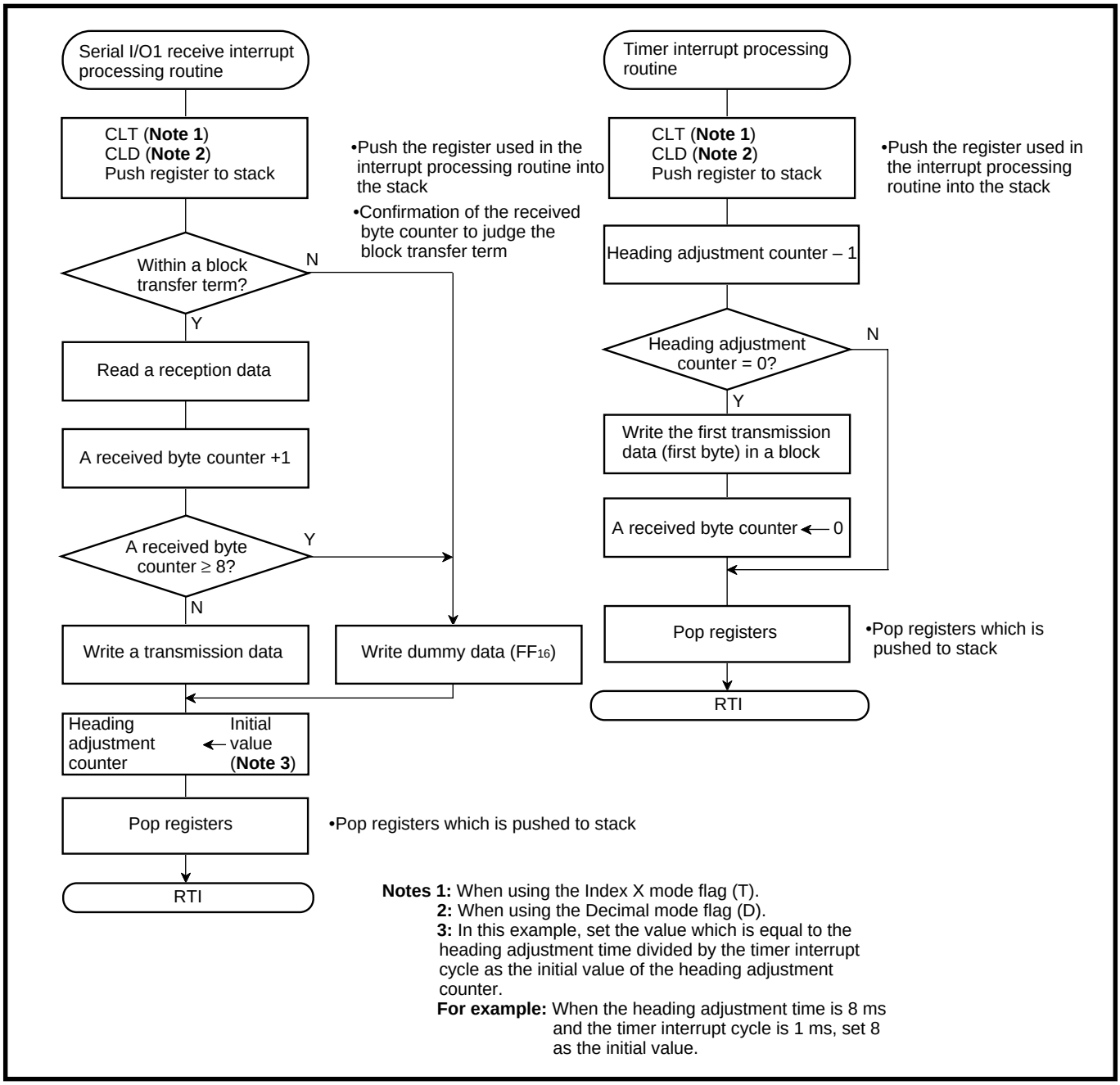


Fig. 2.4.36 Control procedure of slave unit

(4) Communication (transmit/receive) using asynchronous serial I/O (UART)

Outline : 2-byte data is transmitted and received, using the asynchronous serial I/O. Port $P4_0$ is used for communication control.

Figure 2.4.37 shows a connection diagram, and Figure 2.4.38 shows a timing chart.

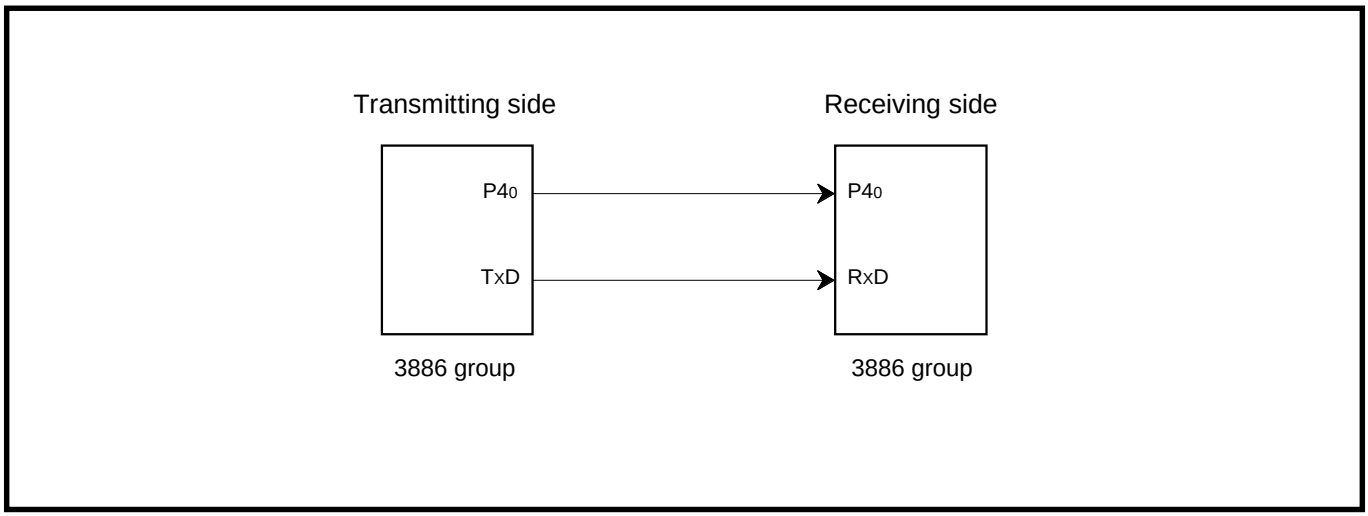


Fig. 2.4.37 Connection diagram (Communication using UART)

Specifications :
- The Serial I/O1 is used (UART is selected).
- Transfer bit rate : 9600 bps ($f(X_{IN})$ = 4.9152 MHz is divided by 512)
- Communication control using port $P4_0$
  (The output level of port $P4_0$ is controlled by software.)
- 2-byte data is transferred from the transmitting side to the receiving side at intervals of 10 ms generated by the timer.

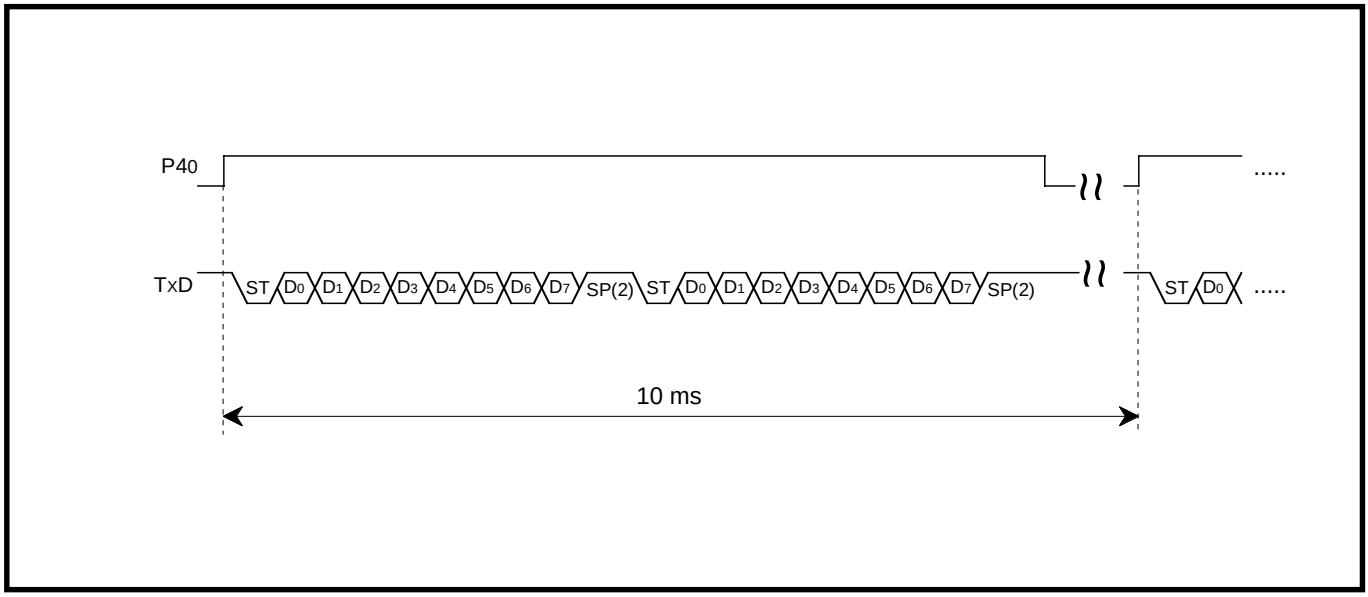


Fig. 2.4.38 Timing chart (using UART)

2.4 Serial I/O

Table 2.4.1 shows setting examples of the baud rate generator (BRG) values and transfer bit rate values; Figure 2.4.39 shows registers setting relevant to the transmitting side; Figure 2.4.40 shows registers setting relevant to the receiving side.

Table 2.4.1 Setting examples of Baud rate generator values and transfer bit rate values

BRG count source (Note 1)	BRG setting value	Transfer bit rate (bps) (Note 2)	
		at f(XIN) = 4.9152 MHz	at f(XIN) = 8 MHz
f(XIN)/4	255(FF_{16})	300	488.28125
f(XIN)/4	127($7F_{16}$)	600	976.5625
f(XIN)/4	63($3F_{16}$)	1200	1953.125
f(XIN)/4	31($1F_{16}$)	2400	3906.25
f(XIN)/4	15($0F_{16}$)	4800	7812.5
f(XIN)/4	7(07_{16})	9600	15625
f(XIN)/4	3(03_{16})	19200	31250
f(XIN)/4	1(01_{16})	38400	62500
f(XIN)	3(03_{16})	76800	125000
f(XIN)	1(01_{16})	153600	250000
f(XIN)	0(00_{16})	307200	500000

Notes 1: Select the BRG count source with bit 0 of the serial I/O1 control register (Address : $1A_{16}$).

2: Equation of transfer bit rate:

$$\text{Transfer bit rate (bps)} = \frac{f(X_{IN})}{(\text{BRG setting value} + 1) \times 16 \times m^{*}}$$

✻m: When bit 0 of the serial I/O1 control register (Address : $1A_{16}$) is set to "0," a value of m is 1.
When bit 0 of the serial I/O1 control register (Address : $1A_{16}$) is set to "1," a value of m is 4.

Transmitting side

Serial I/O1 status register (Address : 19_{16})

SIO1STS (b7 … b0)

- Transmit buffer empty flag
  - Confirm that the data has been transferred from Transmit buffer register to Transmit shift register.
  - When this flag is "1", it is possible to write the next transmission data in to Transmit buffer register.
- Transmit shift register shift completion flag
  Confirm completion of transmitting 1-byte data with this flag.
  "1" : Transmit shift completed

Serial I/O1 control register (Address : $1A_{16}$)

SIO1CON (b7 … b0) = 1 0 0 1 0 0 0 1

- BRG count source selection bit : $f(X_{IN})/4$
- Serial I/O1 synchronous clock selection bit : BRG/16
- $\overline{S_{RDY1}}$ output enable bit : $\overline{S_{RDY1}}$ out disabled
- Transmit enable bit : Transmit enabled
- Receive enable bit : Receive disabled
- Serial I/O1 mode selection bit : Asynchronous serial I/O(UART)
- Serial I/O1 enable bit : Serial I/O1 enabled

UART control register (Address : $1B_{16}$)

UARTCON (b7 … b0) = 0 1 0 0

- Character length selection bit : 8 bits
- Parity enable bit : Parity checking disabled
- Stop bit length selection bit : 2 stop bits
- $P4_5/T_XD$ P-channel output disable bit : CMOS output

Baud rate generator (Address : $1C_{16}$)

BRG (b7 … b0) = 7

$$\text{Set } \frac{f(X_{IN})}{\text{Transfer bit rate} \times 16 \times m \ *} - 1$$

* When bit 0 of the Serial I/O1 control register (Address : $1A_{16}$) is set to "0," a value of m is 1.
When bit 0 of the Serial I/O1 control register (Address : $1A_{16}$) is set to "1," a value of m is 4.

Fig. 2.4.39 Registers setting relevant to transmitting side

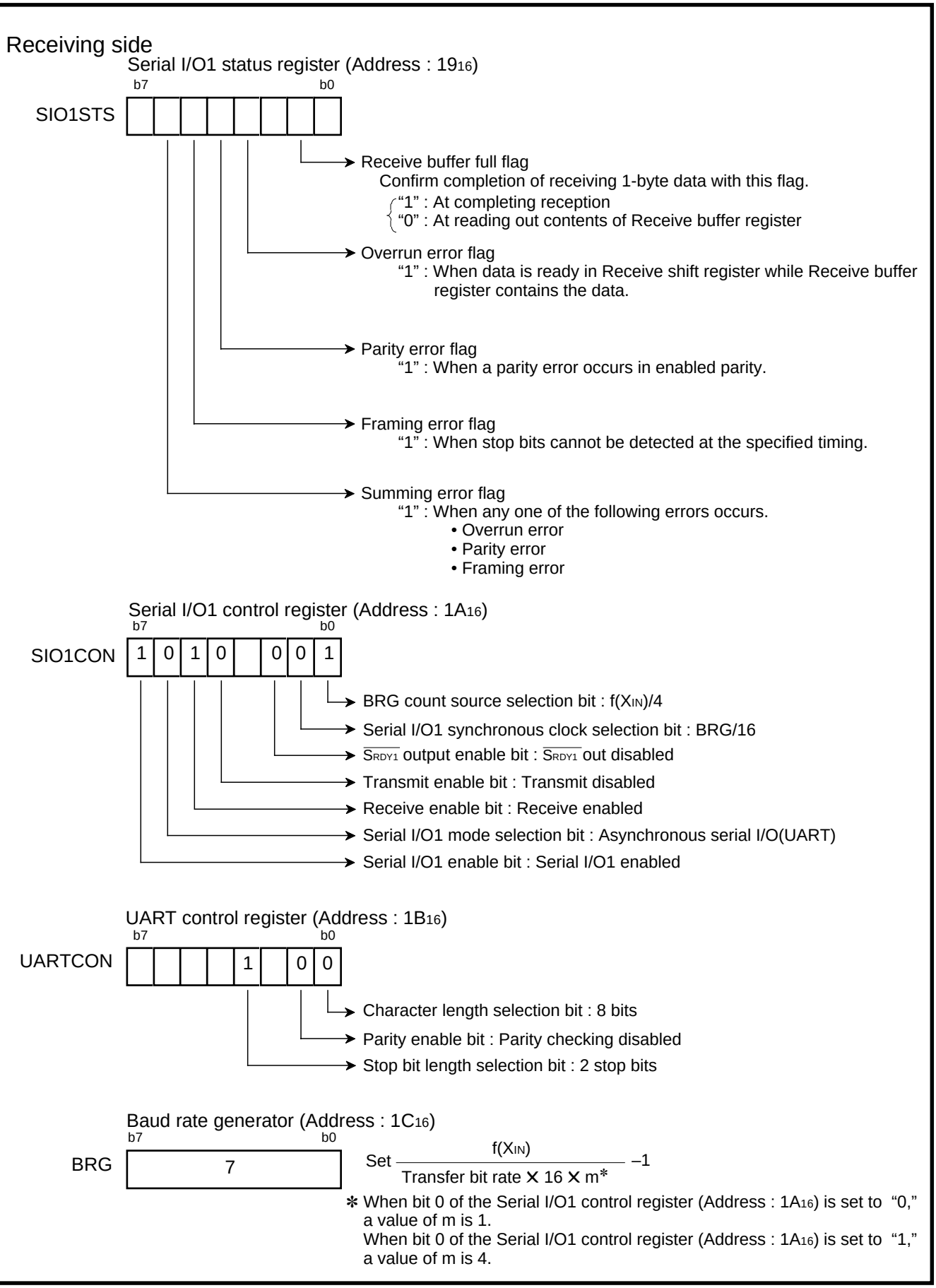


Fig. 2.4.40 Registers setting relevant to receiving side

Figure 2.4.41 shows a control procedure of the transmitting side, and Figure 2.4.42 shows a control procedure of the receiving side.

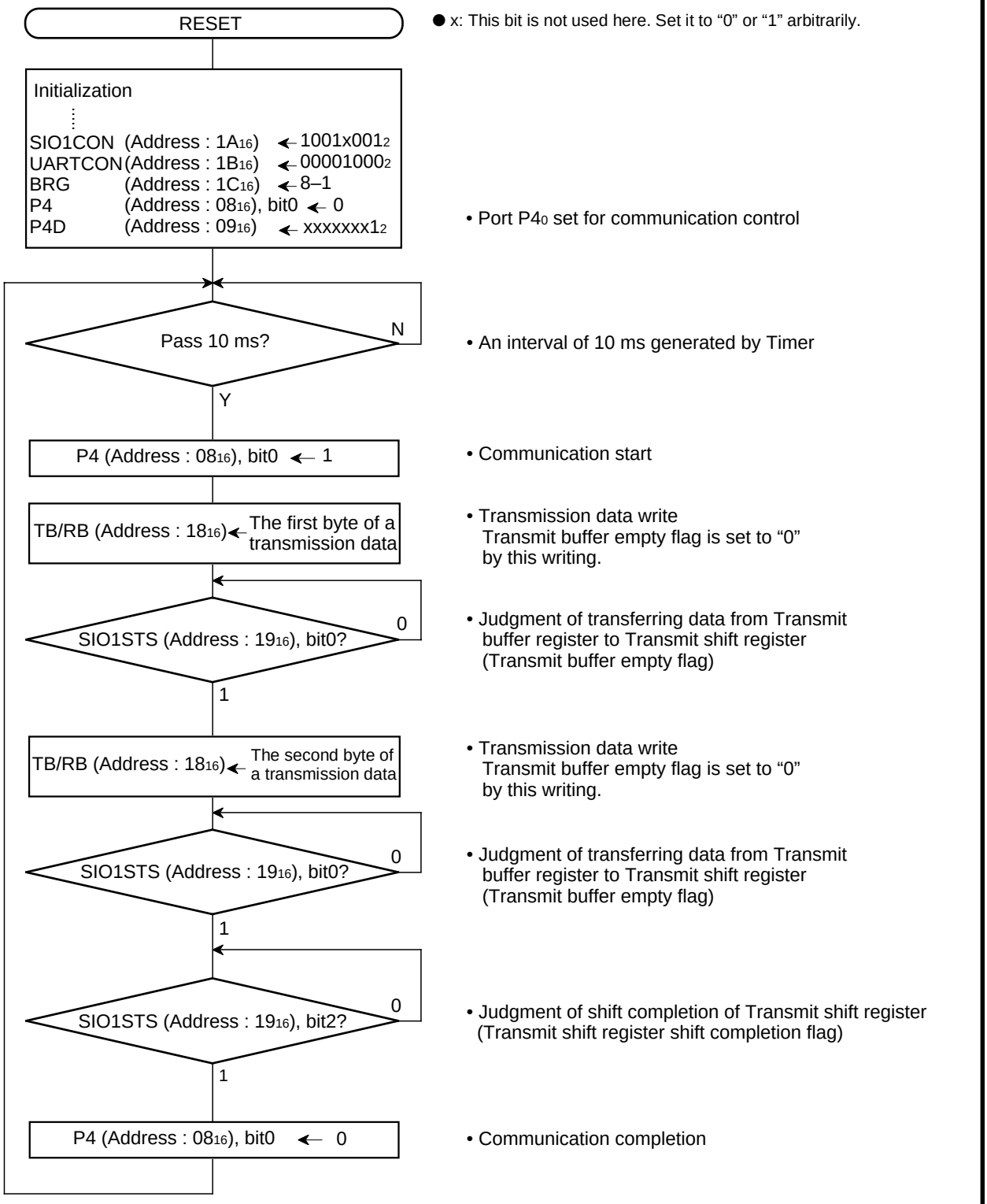


Fig. 2.4.41 Control procedure of transmitting side

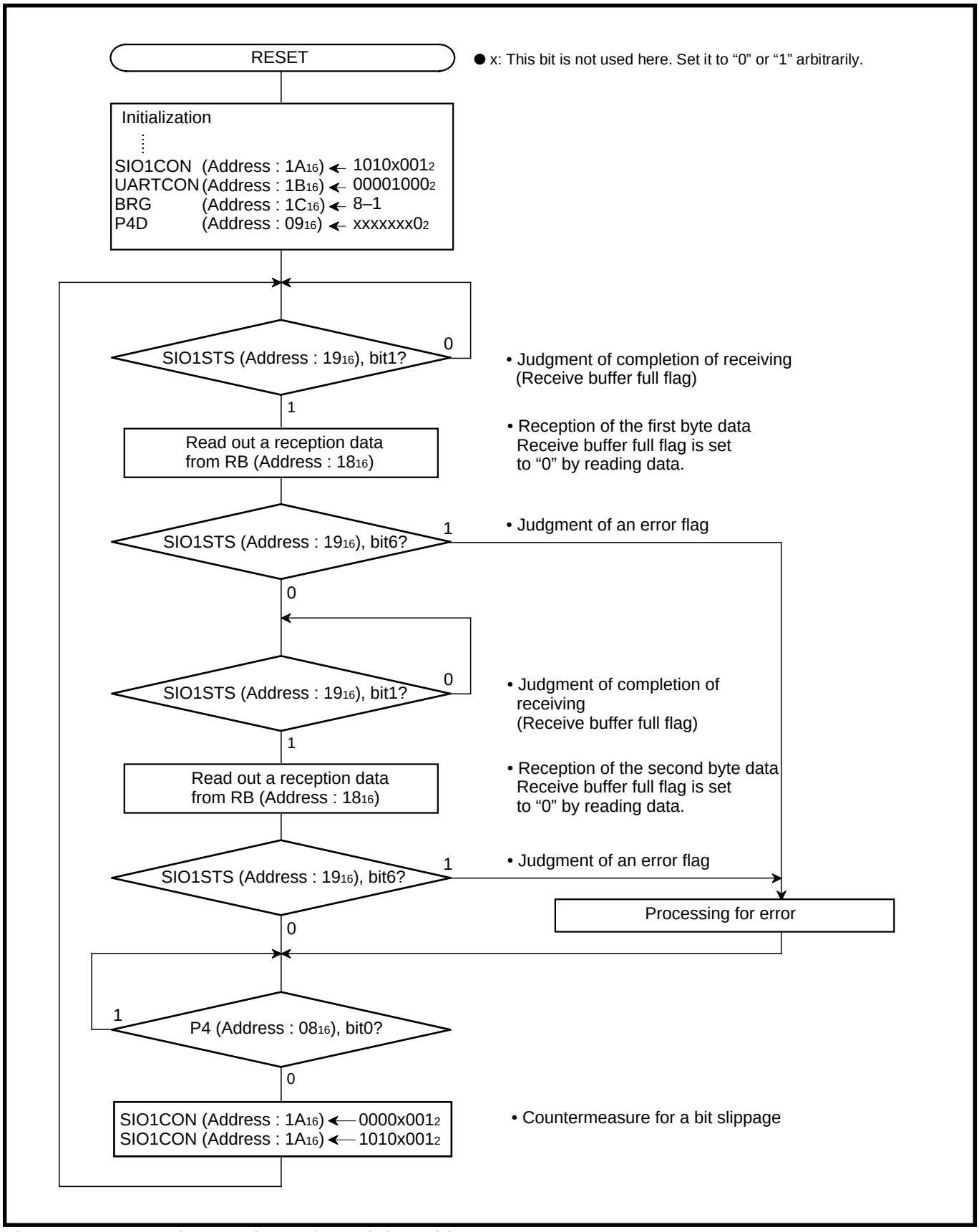


Fig. 2.4.42 Control procedure of receiving side

2.4.6 Notes on serial I/O

(1) Notes when selecting clock synchronous serial I/O (Serial I/O1)

① **Stop of transmission operation**
Clear the serial I/O1 enable bit and the transmit enable bit to "0" (Serial I/O1 and transmit disabled).

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

② **Stop of receive operation**
Clear the receive enable bit to "0" (receive disabled), or clear the serial I/O1 enable bit to "0" (Serial I/O1 disabled).

③ **Stop of transmit/receive operation**
Clear the transmit enable bit and receive enable bit to "0" simultaneously (transmit and receive disabled).
(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

● **Reason**
In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to "0" (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O1 enable bit to "0" (Serial I/O1 disabled) (refer to (1) ①).

(2) Notes when selecting clock asynchronous serial I/O (Serial I/O1)

① **Stop of transmission operation**
Clear the transmit enable bit to "0" (transmit disabled).

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

② **Stop of receive operation**
Clear the receive enable bit to "0" (receive disabled).

③ **Stop of transmit/receive operation**
Only transmission operation is stopped.
Clear the transmit enable bit to "0" (transmit disabled).

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

Only receive operation is stopped.
Clear the receive enable bit to "0" (receive disabled).

(3) $\overline{\text{SRDY1}}$ output of reception side (Serial I/O1)
When signals are output from the $\overline{\text{SRDY1}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDY1}}$ output enable bit, and the transmit enable bit to "1" (transmit enabled).

(4) Setting serial I/O1 control register again (Serial I/O1)
Set the serial I/O1 control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to "0."

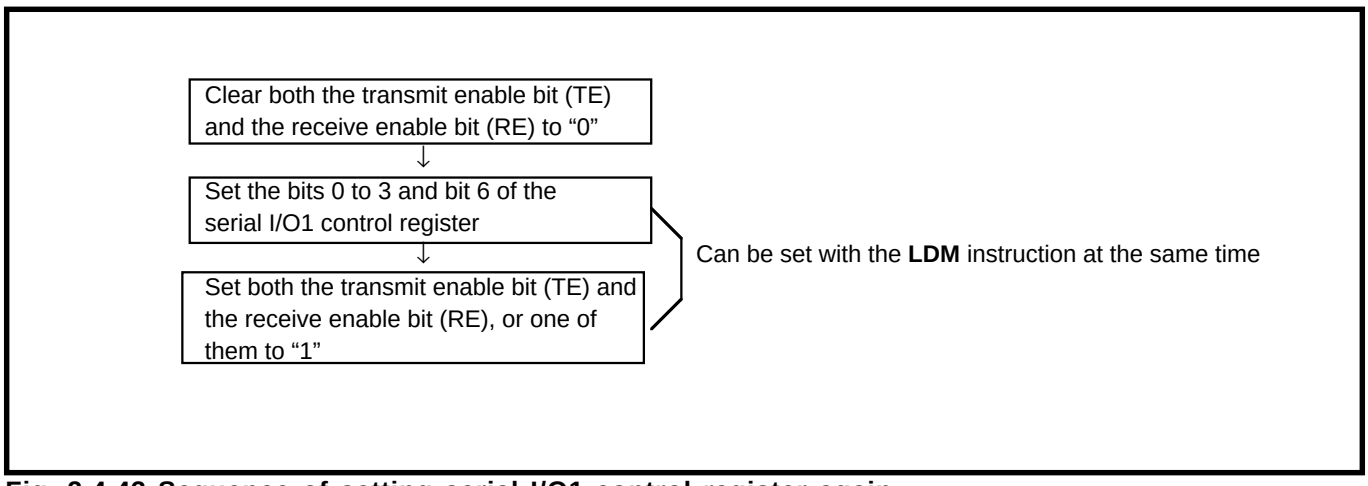


Fig. 2.4.43 Sequence of setting serial I/O1 control register again

(5) Data transmission control with referring to transmit shift register completion flag (Serial I/O1)

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

(6) Transmission control when external clock is selected (Serial I/O1)

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to "1" at "H" of the SCLK input level. Also, write the transmit data to the transmit buffer register (serial I/O shift register) at "H" of the SCLK input level.

(7) Transmit interrupt request when transmit enable bit is set (Serial I/O1)

When the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as shown in the following sequence.

① Set the interrupt enable bit to "0" (disabled) with CLB instruction.
② Prepare serial I/O for transmission/reception.
③ Set the interrupt request bit to "0" with CLB instruction after 1 or more instruction has been executed.
④ Set the interrupt enable bit to "1" (enabled).

● Reason

When the transmission enable bit is set to "1", the transmit buffer empty flag and transmit shift register completion flag are set to "1". The interrupt request is generated and the transmission interrupt bit is set regardless of which of the two timings listed below is selected as the timing for the transmission interrupt to be generated.

- Transmit buffer empty flag is set to "1"
- Transmit shift register completion flag is set to "1"

(8) Transmit data writing (Serial I/O2)

In the clock synchronous serial I/O, when selecting an external clock as synchronous clock, write the transmit data to the serial I/O2 register (serial I/O shift register) at "H" of the transfer clock input level.

2.5 Multi-master I²C-BUS interface

The multi-master I²C-BUS interface is a serial communication circuit, conforming to the Philips I²C-BUS data transfer format. This paragraph explains the overview and the communication example.

2.5.1 Memory map

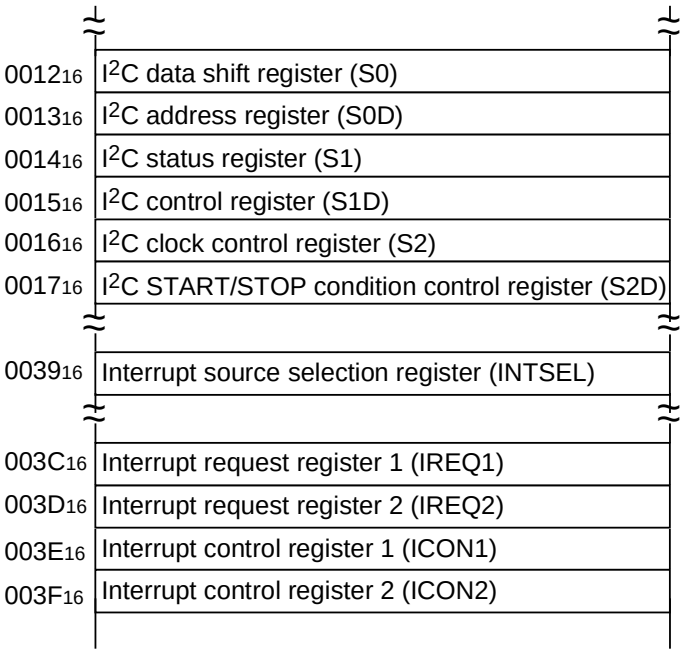


Fig. 2.5.1 Memory map of registers relevant to I²C-BUS interface

2.5.2 Relevant registers

I²C data shift register

b7 b6 b5 b4 b3 b2 b1 b0

I²C data shift register (S0) [Address : 12₁₆]

B	Function	At reset	R	W
0	This register is an 8-bit shift register to store receive data or write transmit data.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Note: Secure 8 machine cycles from clearing MST bit to "0" (slave mode) until writing data to I²C data shift register.
If executing the read-modify-write instruction(SEB, CLB etc.) for this register during transfer, data may become a value not intended.

Fig. 2.5.2 Structure of I²C data shift register

I²C address register

b7 b6 b5 b4 b3 b2 b1 b0

I²C address register (S0D) [Address : 13_{16}]

B	Name	Function	At reset	R	W
0	Read / write bit (RWB)	0 : Write bit 1 : Read bit	0	○	○
1	Slave address (SAD0, SAD1, SAD2, SAD3, SAD4, SAD5, SAD6)	These bits are compared with the address data transmitted from the master.	0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Note: If the read-modify-write instruction(SEB, CLB, etc.) is executed for this register at detecting the stop condition, data may become a value not to intend.

Fig. 2.5.3 Structure of I²C address register

I²C status register

b7 b6 b5 b4 b3 b2 b1 b0

I²C status register (S1) [Address : 14_{16}]

B	Name	Function	At reset	R	W
0	Last receive bit (LRB)	0 : Last bit = "0" 1 : Last bit = "1" (**Note1**)	?	○	×
1	General call detecting flag(AD0)	0 : No general call detected 1 : General call detected(**Note1, 2**)	0	○	×
2	Slave address comparison flag (AAS)	0 : Address disagreement 1 : Address agreement (**Note1, 2**)	0	○	×
3	Arbitration lost detecting flag (AL)	0 : Not detected 1 : Detected (**Note1**)	0	○	×
4	SCL pin low hold bit (PIN)	0 : SCL pin low hold 1 : SCL pin low release	1	○	✻
5	Bus busy flag (BB)	0 : Bus free 1 : Bus busy	0	○	○
6	Communication mode specification bits (TRX, MST)	00 : Slave receive mode 01 : Slave transmit mode 10 : Master receive mode 11 : Master transmit mode	0	○	○
7			0	○	○

Notes 1: These bits and flags can be read out, but cannot be written.

2: These bits can be detected when data format selection bit (ALS) of I²C control register is "0".

3: Do not execute the read-modify-write instruction (SEB, CLB) for this register because all bits of this register are changed by hardware.

✻: "1" can be written to this bit, but "0" cannot be written by program.

Fig. 2.5.4 Structure of I²C status register

I²C control register

b7 b6 b5 b4 b3 b2 b1 b0

I²C control register (S1D) [Address :15₁₆]

B	Name	Function	At reset	R	W
0 1 2	Bit counter (Number of transmit/receive bits) (BC0, BC1, BC2)	b2 b1 b0 0 0 0 : 8 0 0 1 : 7 0 1 0 : 6 0 1 1 : 5 1 0 0 : 4 1 0 1 : 3 1 1 0 : 2 1 1 1 : 1	0	○	○
3	I²C-BUS interface enable bit (ES0)	0 : Disabled 1 : Enabled	0	○	○
4	Data format selection bit (ALS)	0 : Addressing format 1 : Free data format	0	○	○
5	Addressing format selection bit (10 BIT SAD)	0 : 7-bit addressing format 1 : 10-bit addressing format	0	○	○
6	System clock stop selection bit (CLKSTP)	0 : System clock stop when executing WIT or STP instruction 1 : Not system clock stop when executing WIT instruction (Do not use the STP instruction.)	0	○	○
7	I²C-BUS interface pin input level selection bit (TISS)	0 : CMOS input 1 : SMBUS input	0	○	○

Notes : When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended.

Fig. 2.5.5 Structure of I²C control register

I²C clock control register

b7	b6	b5	b4	b3	b2	b1	b0

I²C clock control register (S2) [Address : 16_{16}]

B	Name	Function	At reset	R	W
0	SCL frequency control bits (CCR0, CCR1, CCR2, CCR3, CCR4)	Refer to Table 2.5.1	0	○	○
1					
2					
3					
4					
5	SCL mode specification bit (FAST MODE)	0 : Standard clock mode 1 : High-speed clock mode	0	○	○
6	ACK bit (ACK BIT)	0 : ACK is returned 1 : ACK is not returned	0	○	○
7	ACK clock bit (ACK)	0 : No ACK clock 1 : ACK clock	0	○	○

Fig. 2.5.6 Structure of I²C clock control register

Table 2.5.1 Set value of I²C clock control register and SCL frequency

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

Notes 1: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at ϕ = 4 MHz). "H" duration of the clock fluctuates from –4 to +2 machine cycles in the standard clock mode, and fluctuates from –2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because "L" duration is extended instead of "H" duration reduction.
These are value when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2: Each value of SCL frequency exceeds the limit at ϕ = 4 MHz or more. When using these setting value, use ϕ of 4 MHz or less.

3: The data formula of SCL frequency is described below:
ϕ/(8 × CCR value) Standard clock mode
ϕ/(4 × CCR value) High-speed clock mode (CCR value ≠ 5)
ϕ/(2 × CCR value) High-speed clock mode (CCR value = 5)
Do not set 0 to 2 as CCR value regardless of ϕ frequency.
Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

I²C START/STOP condition control register

b7 b6 b5 b4 b3 b2 b1 b0

I²C START/STOP condition control register (S2D) [Address : 17_{16}]

B	Name	Function	At reset	R	W
0	START/STOP condition set bit (SSC0, SSC1, SSC2, SSC3, SSC4) (**Note**)	SCL release time = ϕ(μs) × (SSC+1) Set up time = ϕ(μs) × (SSC+1)/2 Hold time = ϕ(μs) × (SSC+1)/2	?	○	○
1					
2					
3					
4					
5	SCL/SDA interrupt pin polarity selection bit(SIP)	0 : Falling edge active 1 : Rising edge active	0	○	○
6	SCL/SDA interrupt pin selection bit (SIS)	0 : SDA valid 1 : SCL valid	0	○	○
7	START/STOP condition generating selection bit (STSPSEL)	0 : Setup/Hold time short mode 1 : Setup/Hold time long mode	0	○	○

Note: Do not set "00000_2" or an odd number to the START/STOP condition set bit (SSC4 to SSC0).

Fig. 2.5.7 Structure of I²C START/STOP condition control register

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register [INTSEL: address 0039_{16}]

B	Name	Function	At reset	R	W
0	INT0/input buffer full interrupt source selection bit	0 : INT0 interrupt 1 : Input buffer full interrupt	0	○	○
1	INT1/output buffer empty interrupt source selection bit	0 : INT1 interrupt 1 : Output buffer empty interrupt	0	○	○
2	Serial I/O1 transmit/SCL,SDA interrupt source selection bit	0 : Serial I/O1 transmit interrupt ✻1 1 : SCL,SDA interrupt	0	○	○
3	CNTR0/SCL,SDA interrupt source selection bit	0 : CNTR0 interrupt ✻1 1 : SCL,SDA interrupt	0	○	○
4	Serial I/O2/I²C interrupt source selection bit	0 : Serial I/O2 interrupt ✻2 1 : I²C interrupt	0	○	○
5	INT2/I²C interrupt source selection bit	0 : INT2 interrupt ✻2 1 : I²C interrupt	0	○	○
6	CNTR1/key-on wake-up interrupt source selection bit	0 : CNTR1 interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○
7	AD converter/key-on wake-up interrupt source selection bit	0 : A-D converter interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○

✻1: Do not write "1" to these bits simultaneously.
✻2: Do not write "1" to these bits simultaneously.
✻3: Do not write "1" to these bits simultaneously.

Fig. 2.5.8 Structure of Interrupt source selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT_1/output buffer empty interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.5.9 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2 (IREQ2) [Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	$CNTR_1$/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/I²C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2/I²C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT_4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.5.10 Structure of Interrupt request register 2

2.5 Multi-master I²C-BUS interface

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1 (ICON1) [Address : $3E_{16}$]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1/output buffer empty interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.5.11 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	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	$CNTR_1$/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/I²C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2/I²C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	INT_4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.5.12 Structure of Interrupt control register 2

2.5.3 I²C-BUS overview

The I²C-BUS is a both directions serial bus connected with two signal lines; the SCL which transmits a clock and the SDA which transmits data.

Each port has an N-channel open-drain structure for output and a CMOS structure for input. The devices connected with the I²C-BUS interface use an open drain, so that external pull-up resistors are required. Accordingly, while any one of devices always outputs “L”, other devices cannot output “H”.

Figure 2.5.13 shows the I²C-BUS connection structure.

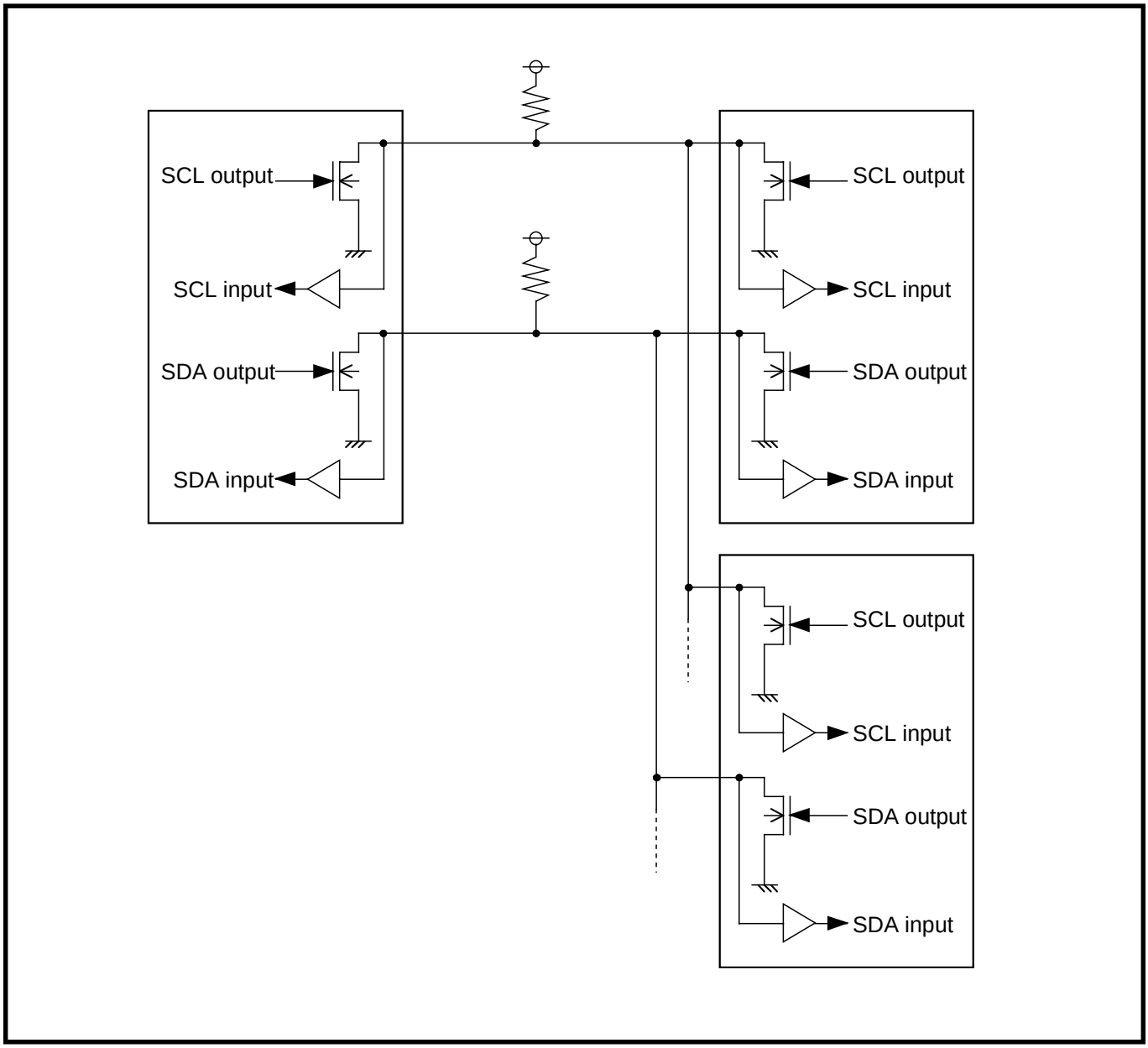


Fig. 2.5.13 I²C-BUS connection structure

2.5.4 Communication format

Figure 2.5.14 shows an I²C-BUS communication format example.
The I²C-BUS consists of the following:
•START condition to indicate communication start
•Slave address and data to specify each device
•ACK to indicate acknowledgment of address and data
•STOP condition to indicate communication completion.

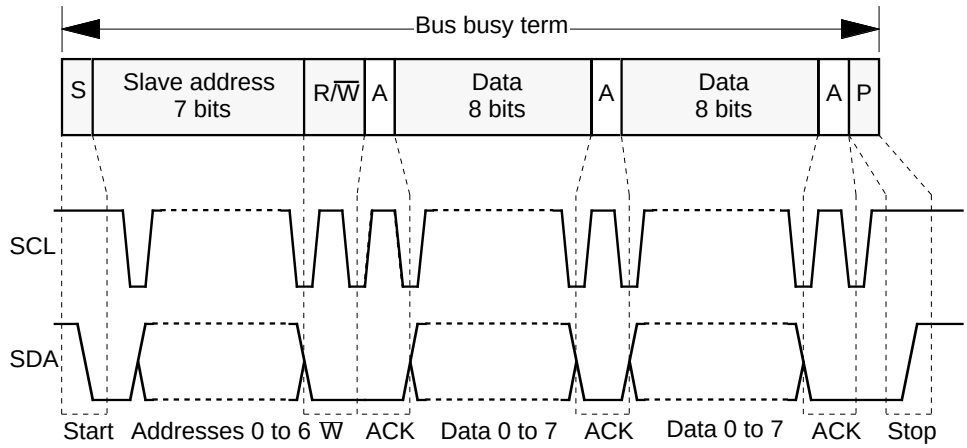


Fig. 2.5.14 I²C-BUS communication format example

(1) START condition

When communication starts, the master device outputs the START condition to the slave device. The I²C-BUS defines that data can be changed when a clock line is "L". Accordingly, data change when a clock line is "H" is treated as STOP or START condition.
The data line change from "H" to "L" when a clock line is "H" is START condition.

(2) STOP condition

Just as in START condition, the data line change from "L" to "H" when a clock line is "H" is STOP condition.
The term from START condition to STOP condition is called "Bus busy". The master device is inhibited from starting data transfer during that term.
The Bus busy status can be judged by using the BB flag of I²C status register (bit 5 of address 0014_{16}).

(3) Slave address

The slave address is transmitted after START condition. This address consists of 7 bits and the 7-th bit functions as the read/write (R/$\overline{W}$) bit which indicates a data transmission method. The slave devices connected with the same I²C-BUS must have their addresses, individually. It is because that address is defined for the master to specify the transmitted/received slave device.
The read/write (R/$\overline{W}$) bit indicates a data transmission direction; "L" means write from the master to the slave, and "H" means read in.

(4) Data

The data has an 8-bit length. There are two cases depending on the read/write (R/$\overline{W}$) bit of a slave address; one is from the master to the slave and the other is from the slave to the master.

(5) ACK bit

The ACK bit clock is generated by the master. This is used for indication of acknowledgment on the SDA line, the slave's busy and the data end.

For example, the slave device makes the SDA line "L" for acknowledgment when confirming the slave address following the START condition. The built-in I²C-BUS interface has the slave address automatic judgment function and the ACK acknowledgment function. "L" is automatically output when the ACK bit of I²C clock control register (bit 6 of address 0016_{16}) is "0" and an address data is received. When the slave address and the address data do not correspond, "H" (NACK) is automatically output.

In case the slave device cannot receive owing to an interrupt process, performing operation or others, the master can output STOP condition and complete data transfer by making the ACK data of the slave address "H" for acknowledgment. Even in case the slave device cannot receive data during data transferring, the communication can be interrupted by performing NACK acknowledgment to the following data.

When the master is receiving the data from the slave, the master can notify the slave of completion of data reception by performing NACK acknowledgment to the last data received from the slave.

(6) RESTART condition

The master can receive or transmit data without transmission of STOP condition while the master is transmitting or receiving a data.

For example, after the master transmitted a data to the slave, transmitting a slave address + R (Read) following RESTART condition can make the following data treat as a reception data.

Additionally, transmitting a slave address + $\overline{W}$ (Write) following RESTART condition can make the following data treat as a transmission data.

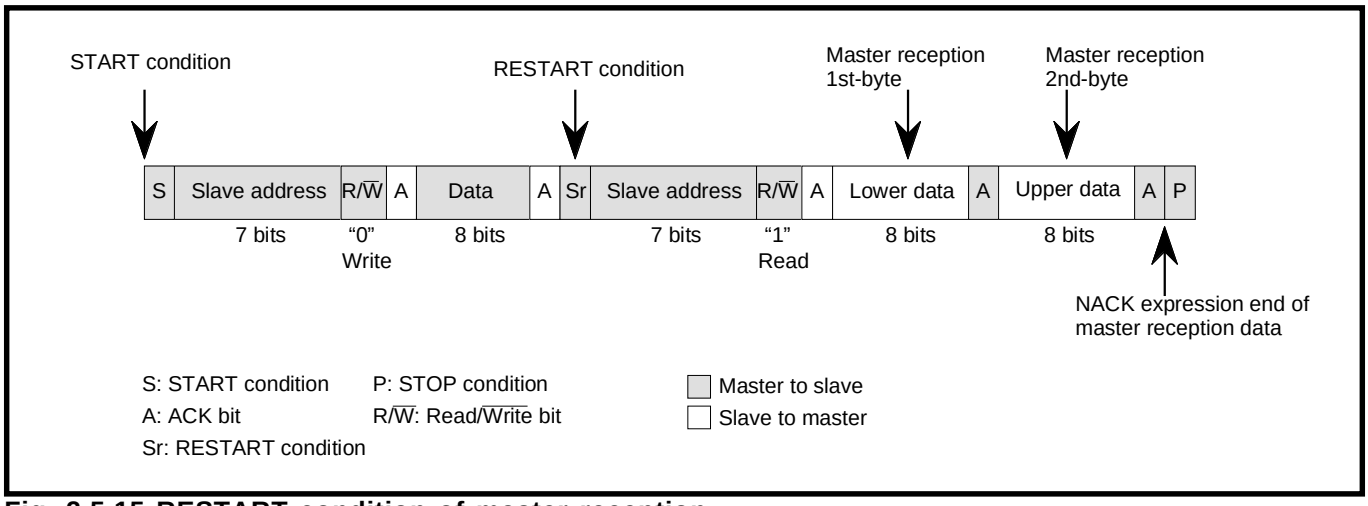


Fig. 2.5.15 RESTART condition of master reception

2.5.5 Synchronization and Arbitration lost

(1) Synchronization

When a plural master exists on the I²C-BUS and the masters, which have different speed, are going to simultaneously communicate; there is a rule to unify clocks so that a clock of each bit can be output correctly.

Figure 2.5.16 shows a synchronized SCL line example. The SCL (A) and the SCL (B) are the master devices having a different speed. The SCL is synchronized waveforms.

As shown by Figure 2.5.16, the SCL lines can be synchronized by the following method; the device which first finishes "H" term makes the SCL line "L" and the device which last remains "L" makes the SCL line "H".

2.5 Multi-master I²C-BUS interface

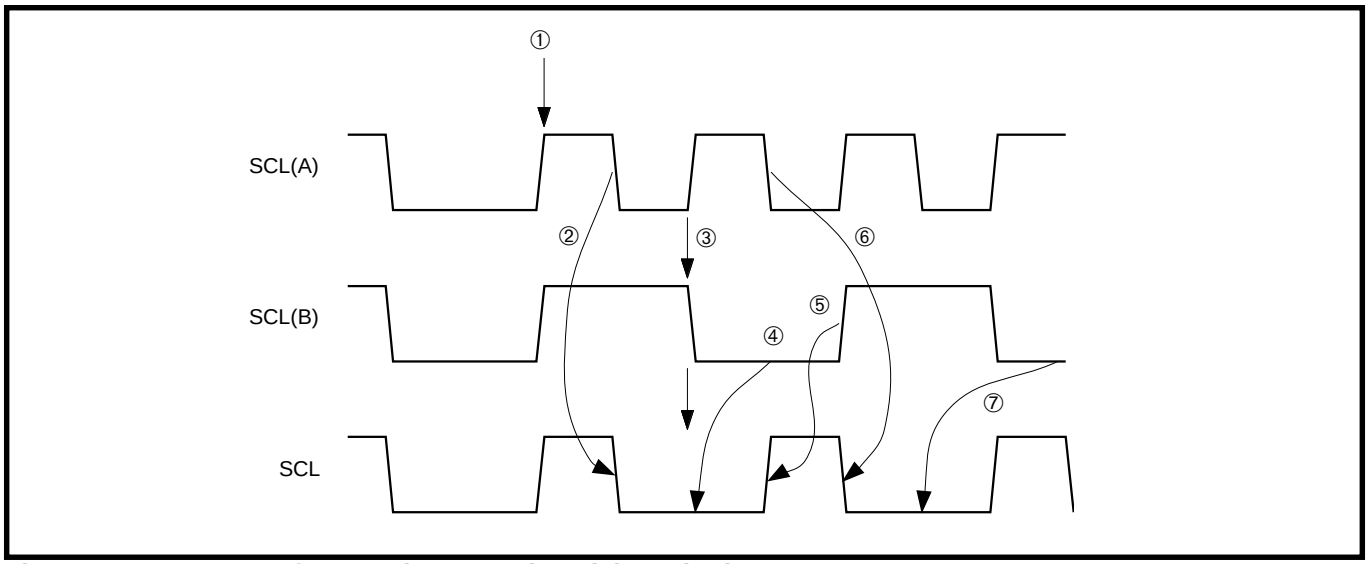


Fig. 2.5.16 SCL waveforms when synchronizing clocks

① After START condition, the masters, which have different speed, simultaneously start clock transmission.
② The SCL outputs “L” because (A) finished counting “H” output; then (B)’s “H” output counting is interrupted and (B) starts counting “L” output.
③ The (A) outputs “H” because (A) finished counting “L” term; the SCL level does not become “H” because (B) outputs “L”, and counting “H” term does not start but stop.
④ (B) outputs “L” term.
⑤ The SCL outputs “H” because (B) finished counting “L” term; then (B)’s “H” output counting is started at the same time as (A).
⑥ The SCL outputs “L” because (A) first finished counting “H” output; then (B)’s “H” output counting is interrupted and (B) starts counting “L” output.
⑦ The above are repeatedly performed.

(2) Clock synchronization during communication

In the I²C-BUS, the slave device is permitted to retain the SCL line “L” and become waiting status for transmission from the master. By byte unit, for the reception preparation of the slave device, the master can become waiting status by making the SCL line “L”, which is after completion of byte reception or the ACK.
By bit unit, it is possible to slow down a clock speed by retaining the SCL line “L” for slave devices having limited hardware.

The 3886 group can transmit data correctly without reduction of data bits toward waiting status request from the slave device. It is because the synchronization circuit is included for the case when retaining the SCL line “L” as an internal hardware.
After the last bit, including the ACK bit, of a transmission/reception data byte, the SCL line automatically remains “L” and waiting status is generated until completion of an interrupt process or reception preparation.

(3) Arbitration lost

A plural master exists on the same bus in the I²C-BUS and there are possibility to start communication simultaneously. Even when the master devices having the same transmission frequency start communication simultaneously, which device must transmit data correctly. Accordingly, there is the definition to detect a communication confliction on the SDA line in the I²C-BUS.
The SDA line is output at the timing synchronized by the SCL, however, the synchronization among the SDA signals is not performed.

2.5.6 I²C-BUS communication usage example

This clause explains a control example using the I²C-BUS. This is a control example as the master device and the slave device in the Read Word protocol of I²C-BUS protocol.
The following is a communication example of E²PROM (24C0X).

Communication specifications:

- Communication frequency = 100 kHz
- Slave address of communication destination, E²PROM, = "1010000X_2" (X means the read/$\overline{\text{write}}$ bit)
- Address = E²PROM internal address
- The communication process is performed in the interrupt process. However, the main process performs an occurrence of the first START condition and a slave address set.
- A communication buffer is established. Data transfer between the main process and the interrupt process is performed through the communication buffer.

(1) Initial setting

Figure 2.5.17 shows an initial setting example using I²C-BUS communication.

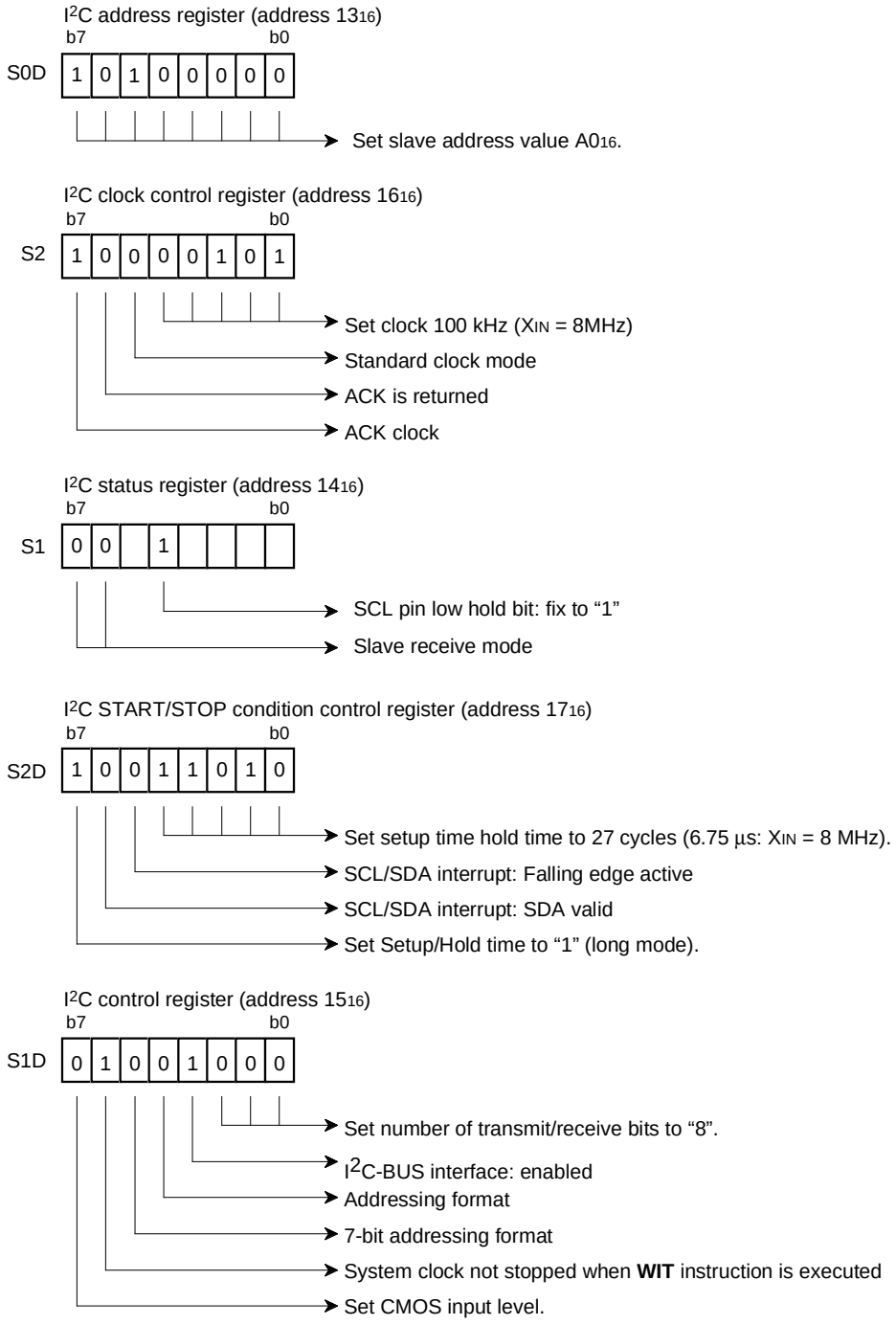


Fig. 2.5.17 Initial setting example

(2) Communication example in master device

The master device follows the procedures ① to ⑥ shown by Figure 2.5.18.
Additionally, the shaded area in the figure is a transmission data from the master device and the white area is a transmission data from the slave device.

① Generating of START condition; Transmission of slave address + write bit
② Transmission of command
③ Generating of RESTART condition; Transmission of slave address + read bit
④ Reception of lower data
⑤ Reception of upper data
⑥ Generating of STOP condition

Figures 2.5.19 to 2.5.24 show the procedures ① to ⑥.

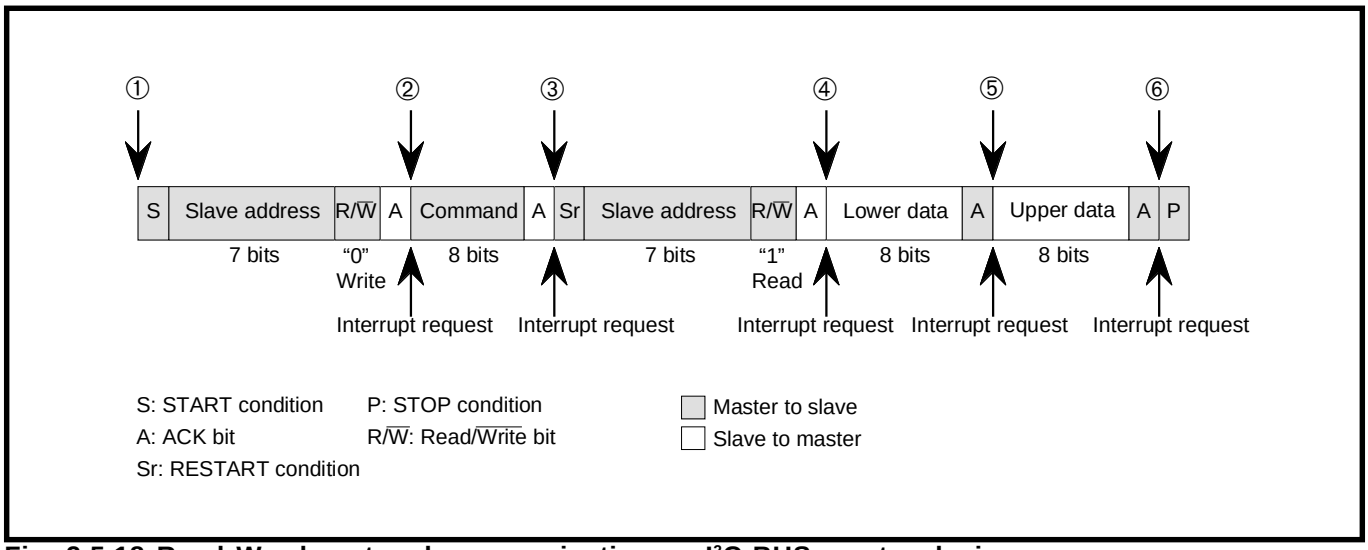


Fig. 2.5.18 Read Word protocol communication as I²C-BUS master device

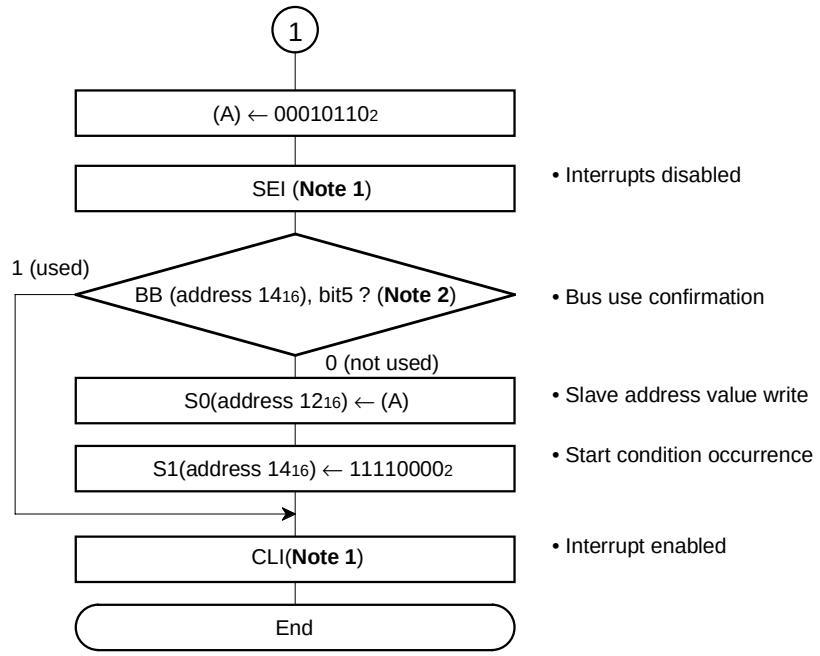
2.5 Multi-master I²C-BUS interface

① **Generating of START condition; Transmission of slave address + write bit**

After confirming that other master devices do not use the bus, generate the START condition, because the I²C-BUS is a multi-master.

Write “slave address + write bit” to the I²C data shift register (address 0012_{16}) before performing to make the START condition generate. It is because the SCL of 1-byte unit is output, following occurrence of the START condition.

If other master devices start communication until an occurrence of the START condition after confirming the bus use, it cannot communicate correctly. However in this case, that situation does not affect other master devices owing to detection of an arbitration lost or the START condition duplication preventing function.



Notes 1: In this example, the SEI instruction to disable interrupts need not be executed because this processing is going to be performed in the interrupt processing. When the start condition is generated out of the interrupt processing, execute the SEI instruction to disable interrupts.

2: Use the branch bit instruction to confirm bus busy.

Fig. 2.5.19 Generating of START condition and transmission process of slave address + write bit

② **Transmission of command**

Confirm correct completion of communication at ① before command transmission. When receiving the STOP condition, a process not to transmit a command is required, because the internal I²C-BUS generates an interrupt request also owing to the STOP condition transmitted to other devices. After confirming correct completion of communication, write a command to the I²C data shift register (address 0012_{16}).

In case the AL bit (bit 3 of address 0014_{16}) is "1", check the slave address comparison flag (ASS bit; bit 2 of address 0014_{16}) to judge whether the device given a right of master transmission owing to an arbitration specifies itself as a slave address. When it is "1", perform the slave reception; when "0", wait for a STOP condition occurrence caused by other devices and the communication completion.

In case the AL bit is "0", check the last received bit (LRB bit; bit 0 of address 0014_{16}). When it is "1", make the STOP condition generate and release the bus use, because the specified slave device does not exist on the I²C-BUS.

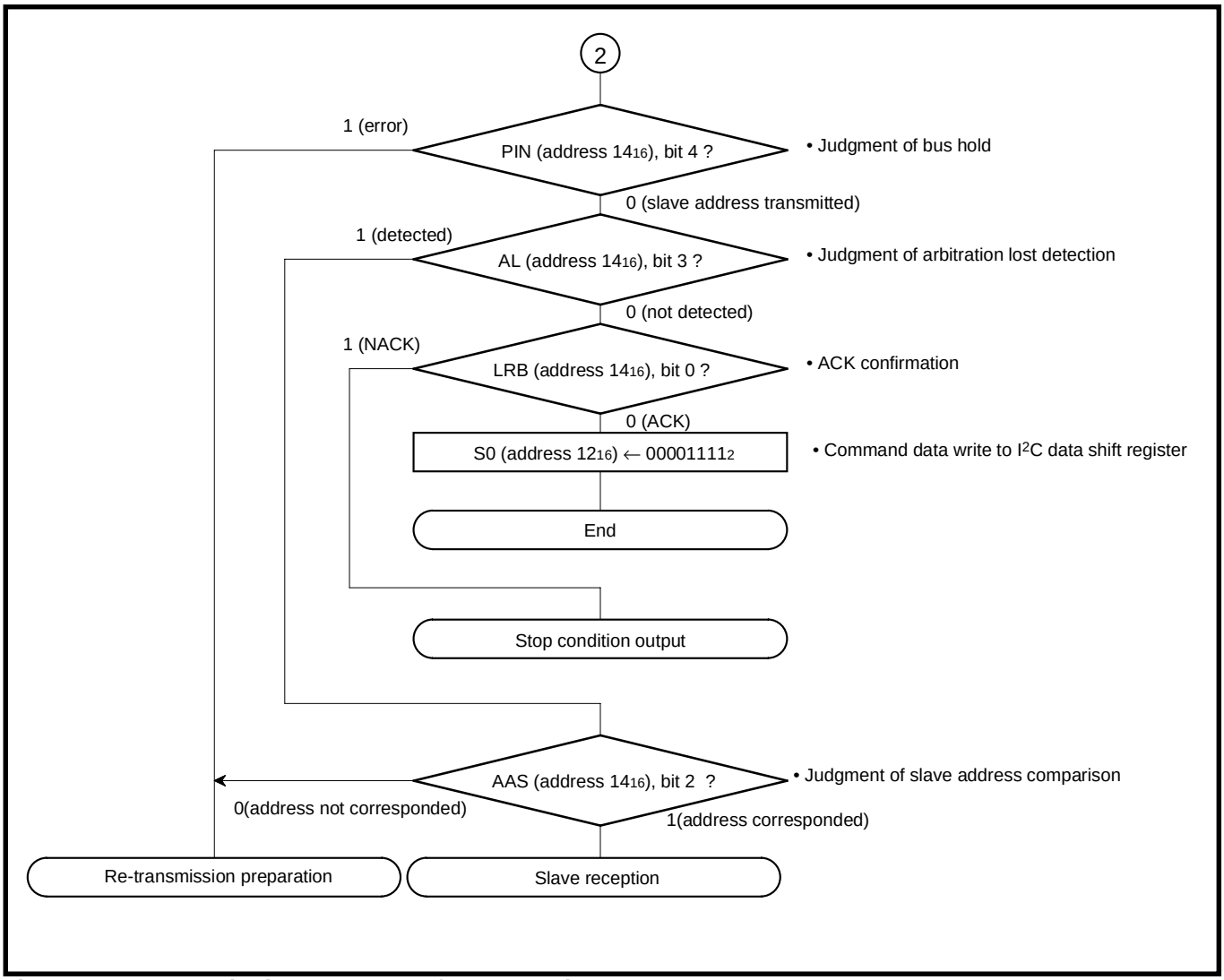


Fig. 2.5.20 Transmission process of command

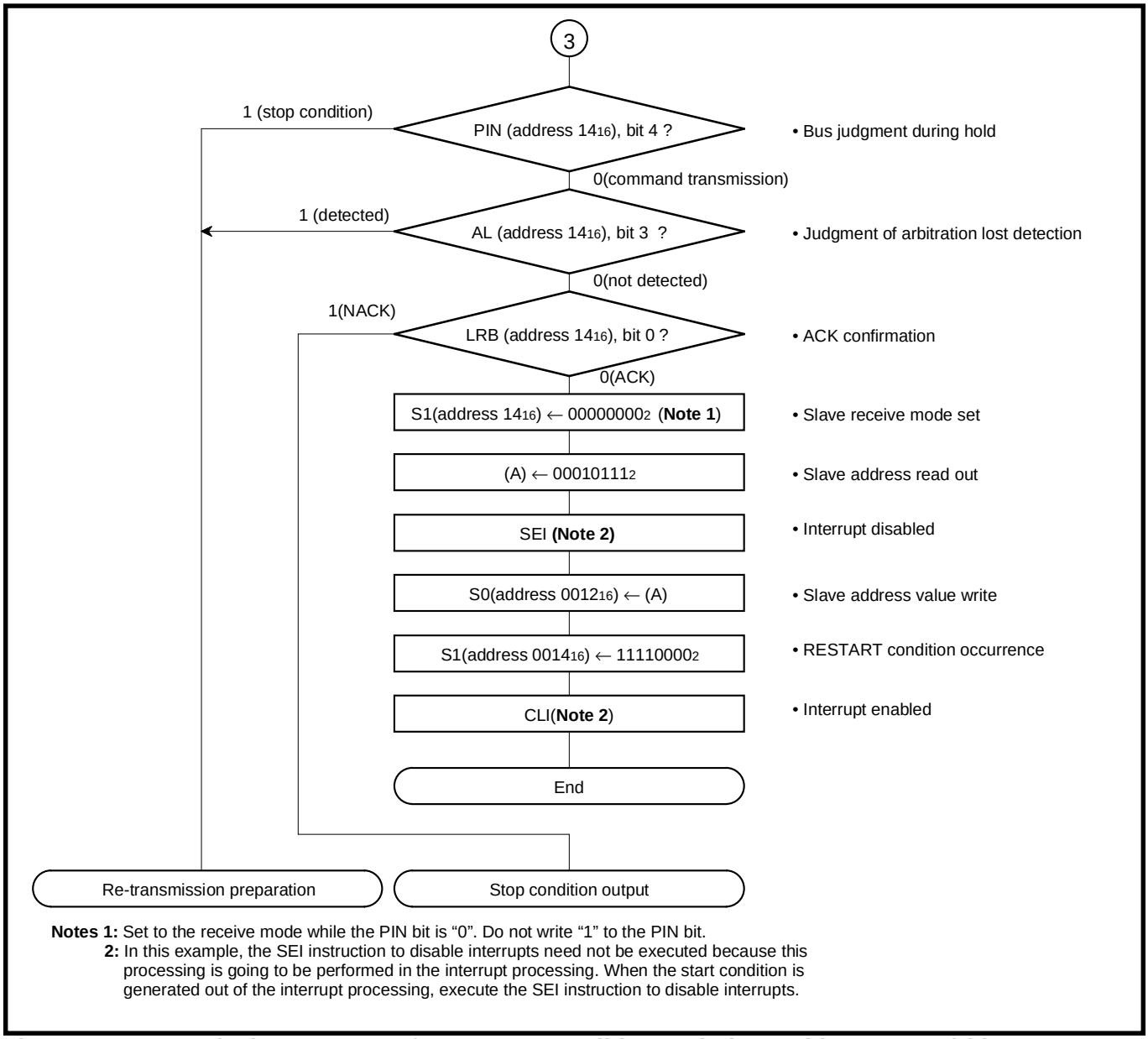
③ **Generating of RESTART condition; Transmission of slave address + read bit**

Confirm correct completion of communication at ② before generating the RESTART condition. After confirming correct completion, generate the RESTART condition and perform the transmission process of "slave address + read bit". Note that procedure because that is different from ①'s process.

As the same reason as ①, write "slave address + read bit" to the I²C data shift register (address 0012_{16}) before performing to make the START condition generate. However, when writing a slave address to the I²C data shift register in this condition, a slave address is output at that time. Consequently, the RESTART condition cannot be generated. Therefore, follow the slave reception procedure before those processes.

In case the arbitration lost detecting flag (AL bit, bit 3 of address 0014_{16}) is "1", return to the process ①, because other master devices will have priority to communicate.

When the last received bit (LRB bit; bit 0 of address 0014_{16}) is "1", generate the STOP condition and make the bus release, because acknowledgment cannot be done owing to BUSY status of the slave device specified on the I²C-BUS or other reasons.



Notes 1: Set to the receive mode while the PIN bit is "0". Do not write "1" to the PIN bit.
2: In this example, the SEI instruction to disable interrupts need not be executed because this processing is going to be performed in the interrupt processing. When the start condition is generated out of the interrupt processing, execute the SEI instruction to disable interrupts.

Fig. 2.5.21 Transmission process of RESTART condition and slave address + read bit

④ **Reception of lower data**

Confirm correct completion of communication at ③ before receiving the lower data. After confirming correct completion, clear the ACK bit (bit 6 of address 0016_{16}) to "0", in which ACK is returned, and set to the master receive mode. After that, write dummy data to the I²C data shift register (address 0012_{16}).

When the MST bit (bit 7 of address 0014_{16}) is "0", perform the error process explained as follows and return to the process ①.

When the last received bit (LRB bit; bit 0 of address 0014_{16}) is "1", generate the STOP condition and make the bus release, because the slave device specified on the I²C-BUS does not exist.

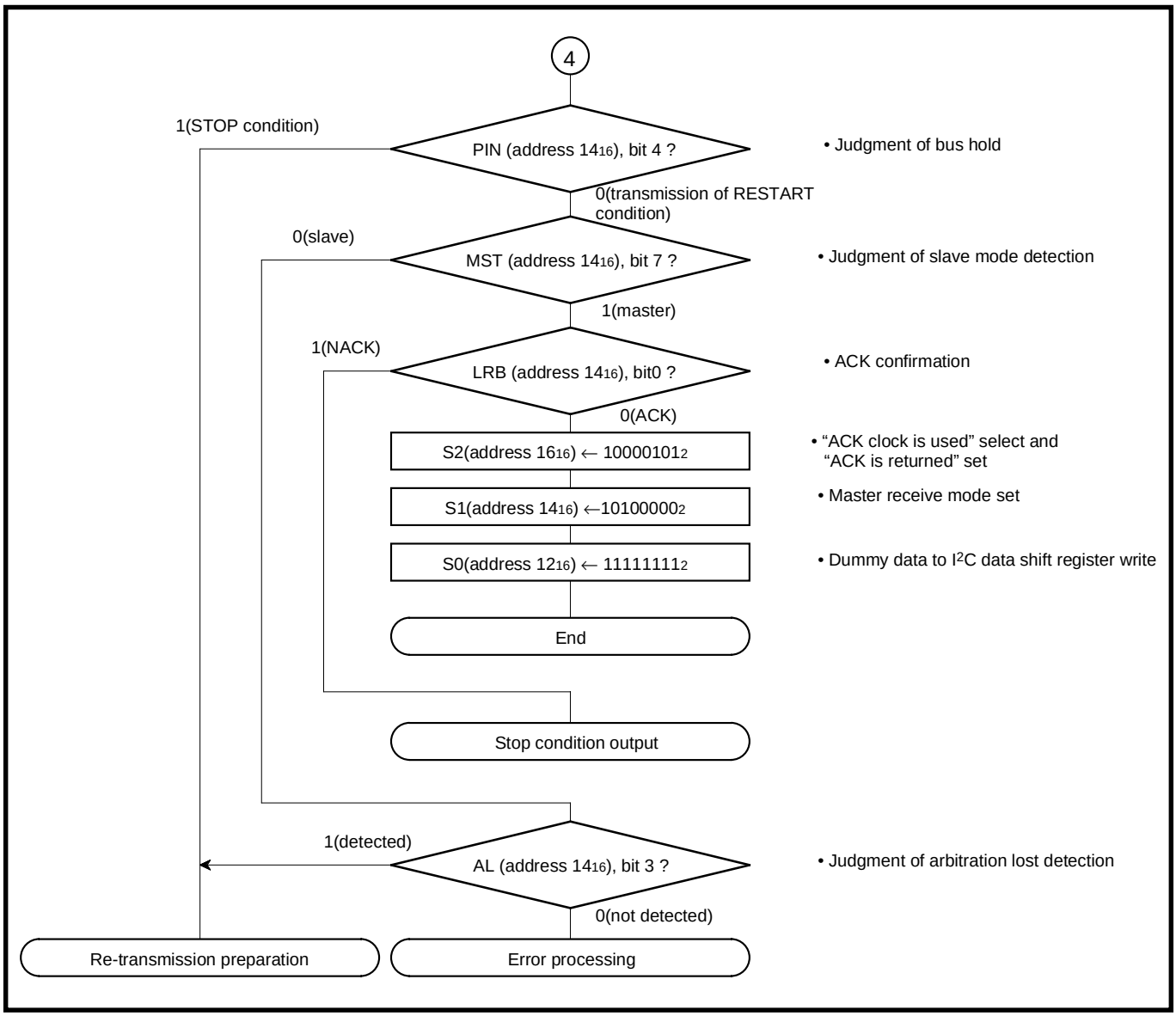


Fig. 2.5.22 Reception process of lower data

⑤ **Transmission of upper data**

Confirm correct completion of communication at ④ before receiving the upper data. After confirming correct completion, store the received data (lower data).

Set the ACK bit (bit 6 of address 0016_{16}) to "1", in which ACK is not returned and write dummy data to the I²C data shift register (address 0012_{16}).

When the MST bit (bit 7 of address 0014_{16}) is "0", return to the process ①, because other devices have priority to communicate.

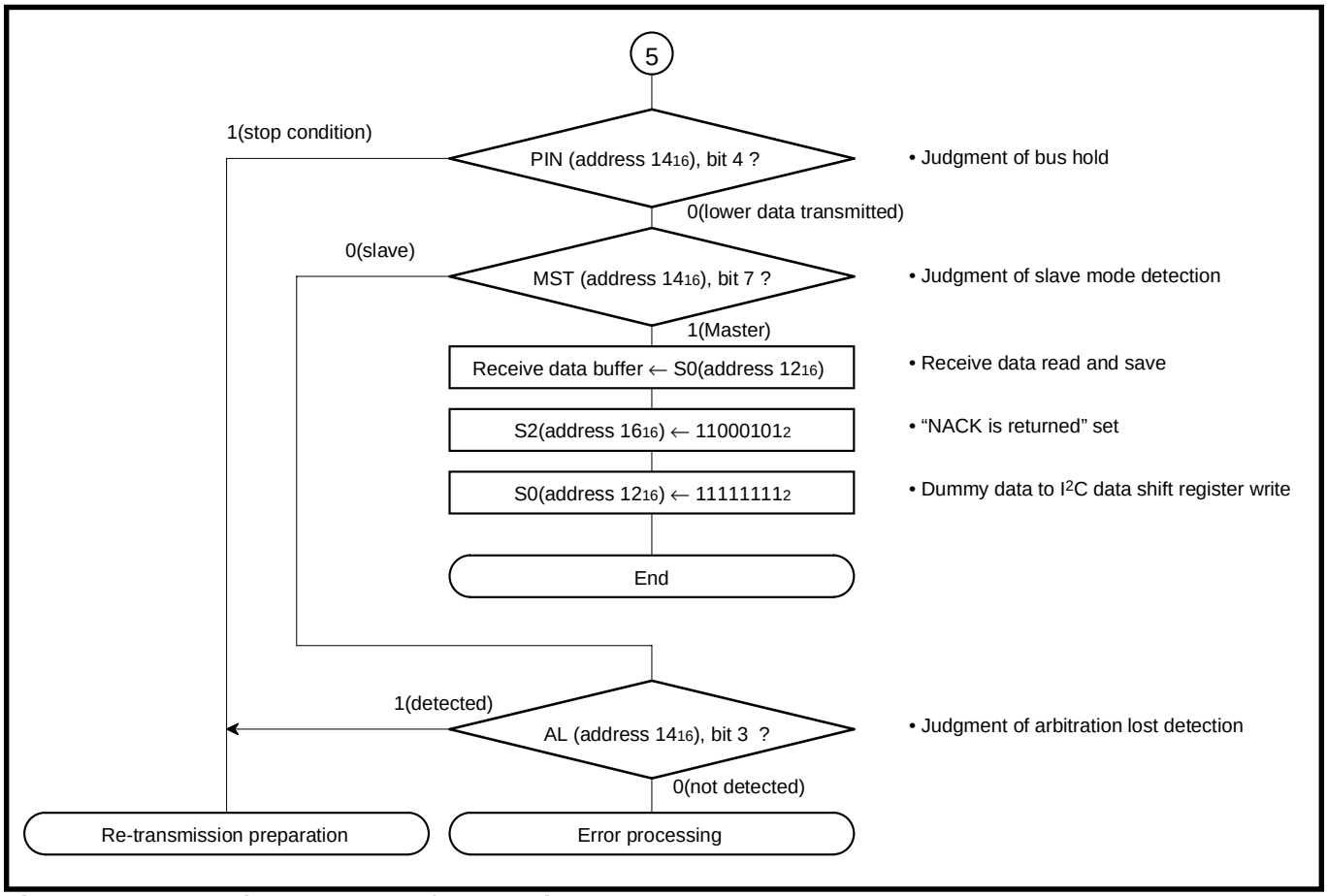


Fig. 2.5.23 Reception process of upper data

⑥ **Generating of STOP condition**

Confirm correct completion of communication at ⑤ before generating the STOP condition. After confirming correct completion, store the received data (upper data).

Clear the ACK bit (bit 6 of address 0016_{16}) to "0", in which ACK is returned, and generate the STOP condition. The communication mode is set to the slave receive mode by the occurrence of STOP condition.

When the MST bit (bit 7 of address 0014_{16}) is "0", return to the process ①, because other devices have priority to communicate.

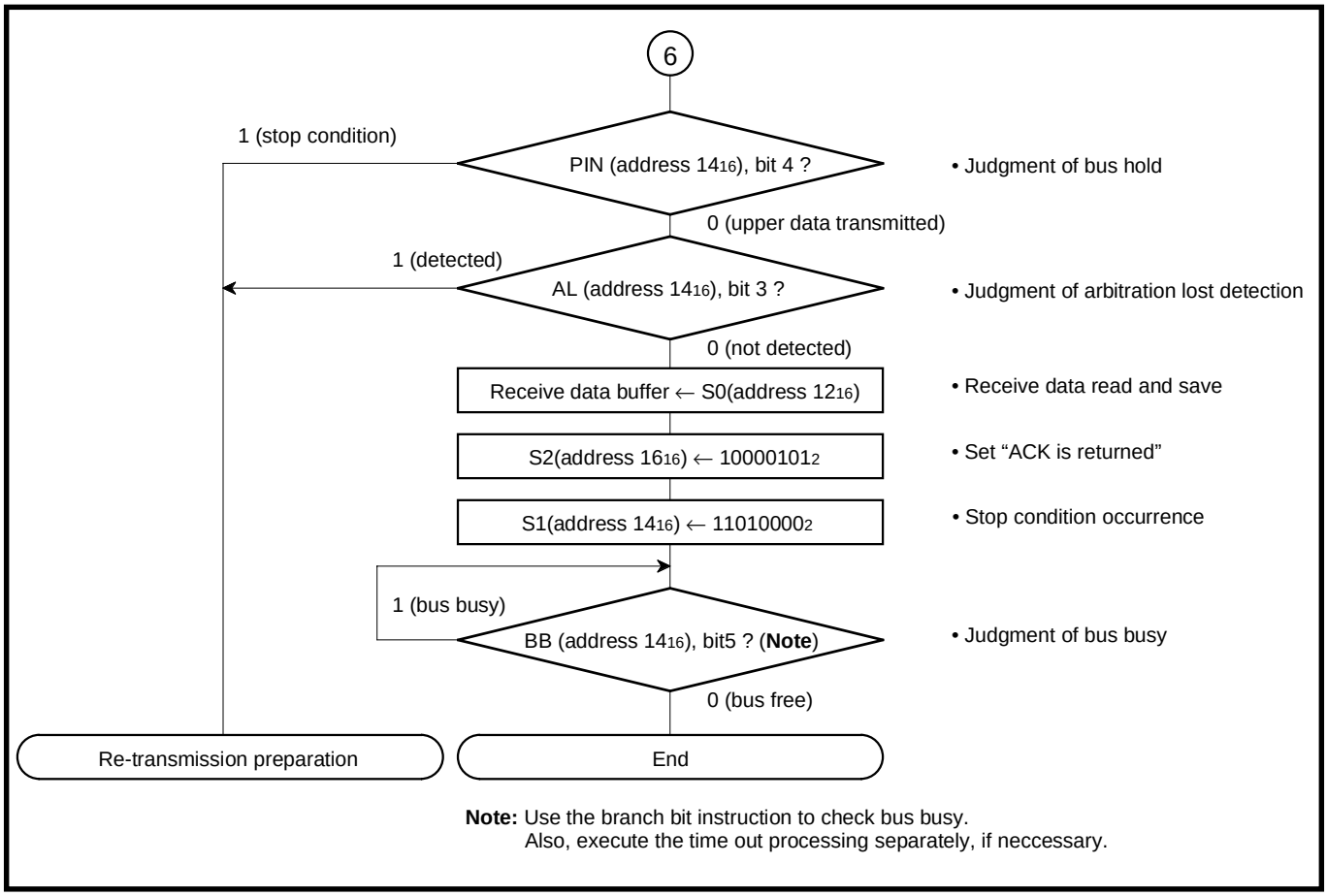


Fig. 2.5.24 Generating of STOP condition

(3) Communication example in slave device

The slave device follows the procedures ① to ⑥ shown by Figure 2.5.25.
The only difference from the master device's communication is an occurrence of interrupt request after detection of STOP condition.

① Reception of START condition; Transmission of ACK bit due to slave address correspondence
② Reception of command
③ Reception of RESTART condition; Reception of slave address + read bit
④ Transmission of lower data
⑤ Transmission of upper data
⑥ Reception of STOP condition

Figures 2.5.26 to 2.5.31 show the procedures ① to ⑥.

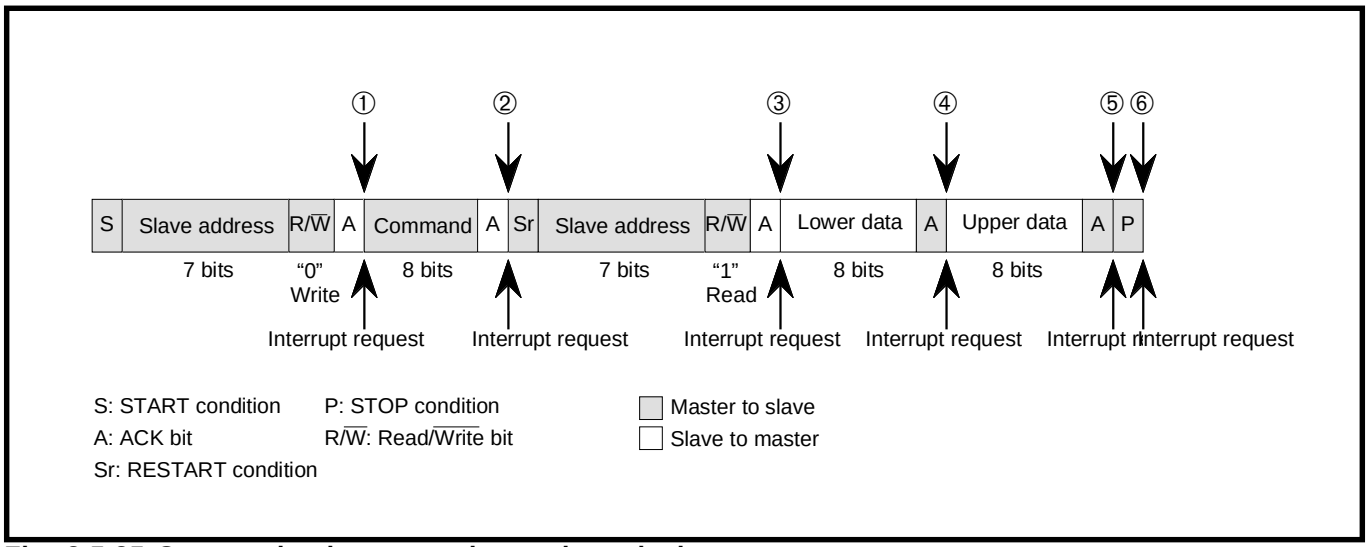


Fig. 2.5.25 Communication example as slave device

① **Reception of START condition; Transmission of ACK bit due to slave address correspondence**
In the case of operation as the slave, all processes are performed in the interrupt after setting of the slave reception in the main process, because an interrupt request does not occur until correspondence of a slave address.
In the first interrupt, after confirming correspondence of the slave address, write dummy data to receive a command into the I²C data shift register.

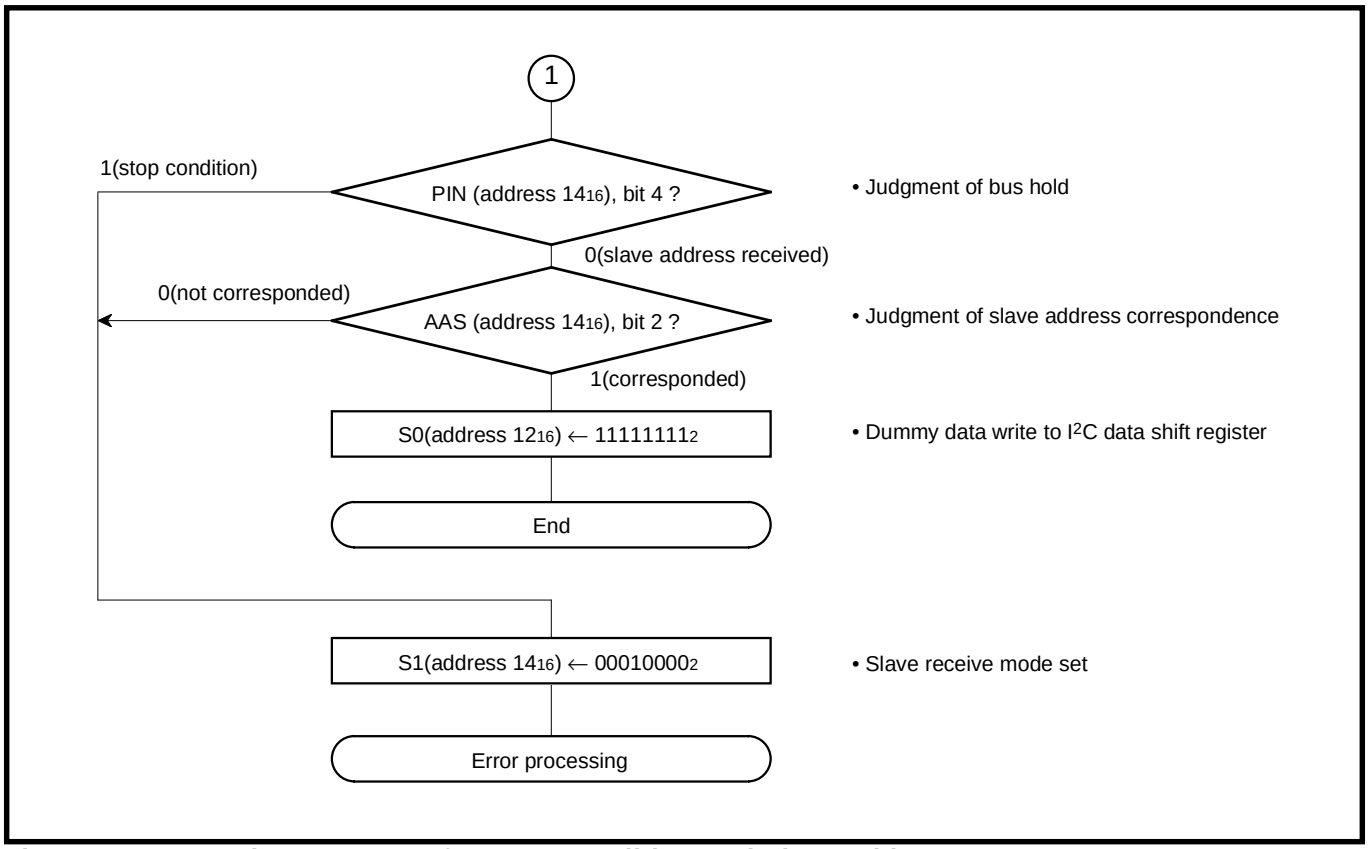


Fig. 2.5.26 Reception process of START condition and slave address

② **Reception of command**

Confirm correct completion of the command reception in the interrupt after receiving the command. After confirming correct command from the host, write dummy data to the I²C data shift register to wait for reception of the next slave address.

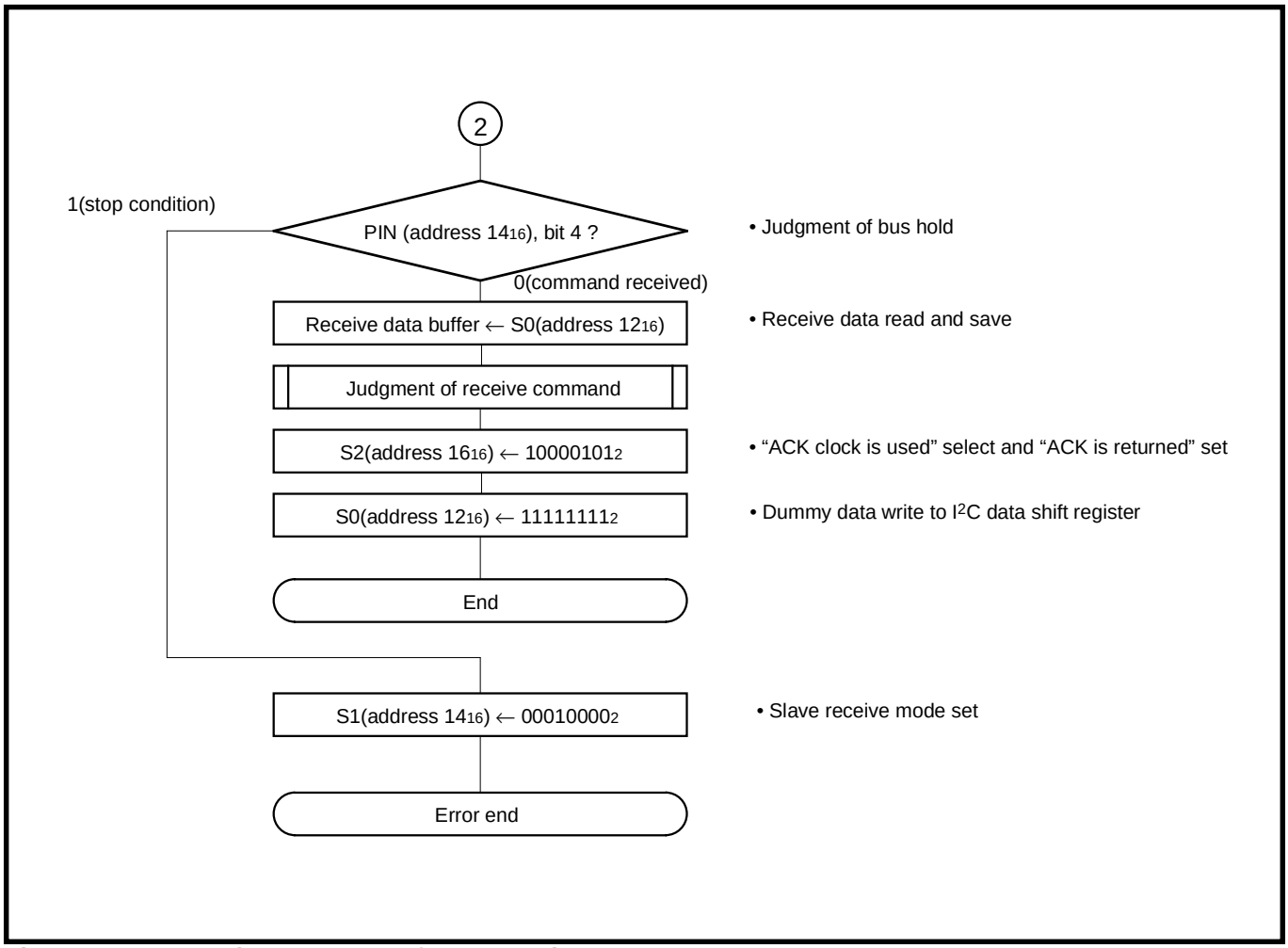


Fig. 2.5.27 Reception process of command

③ **Reception of RESTART condition and slave address**

After receiving a slave address, prepare transmission data.

Judgment whether receiving data or transmitting is required, because the mode is automatically switched between the receive mode and the transmit mode depending on the R/$\overline{W}$ bit of the received slave address. Accordingly, judge whether read or write referring the slave address comparison flag (AAS bit; bit 2 of address 0014_{16}).

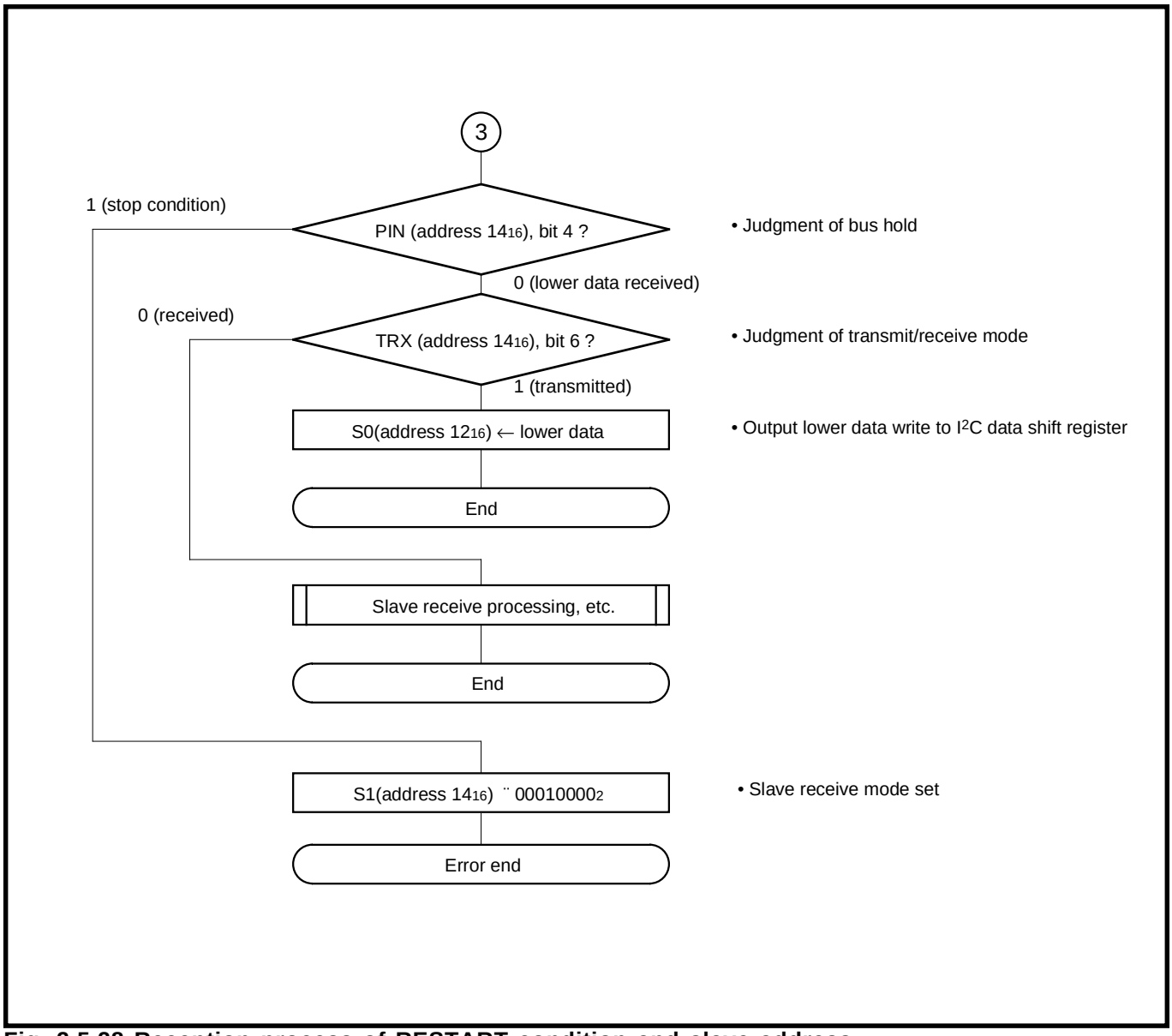


Fig. 2.5.28 Reception process of RESTART condition and slave address

④ **Transmission of lower data**

Before transmitting the upper data, restart to transmit the data at ④ and confirm correct completion of transmission of the lower data set in the slave address reception interrupt.
After that, transmit the upper data.

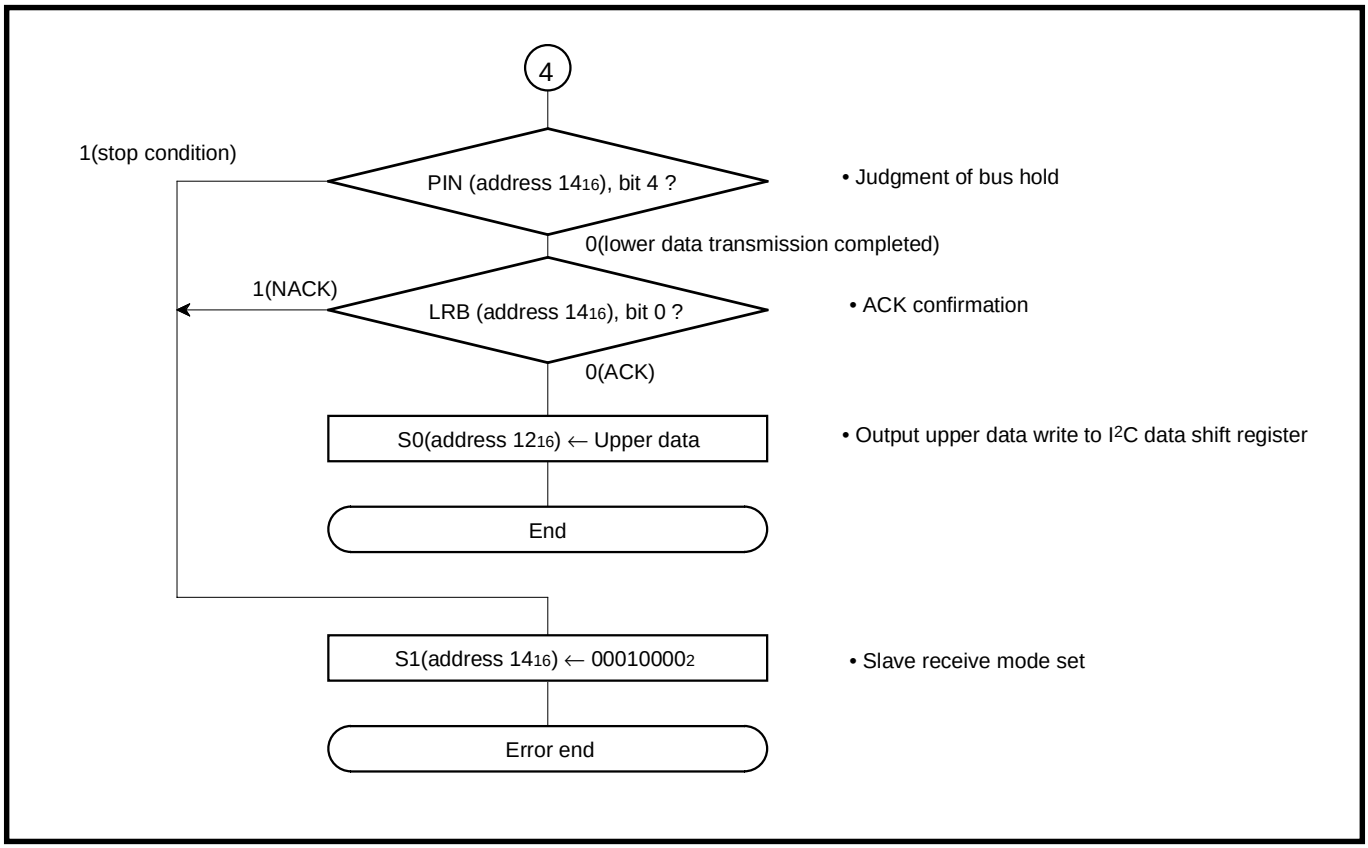
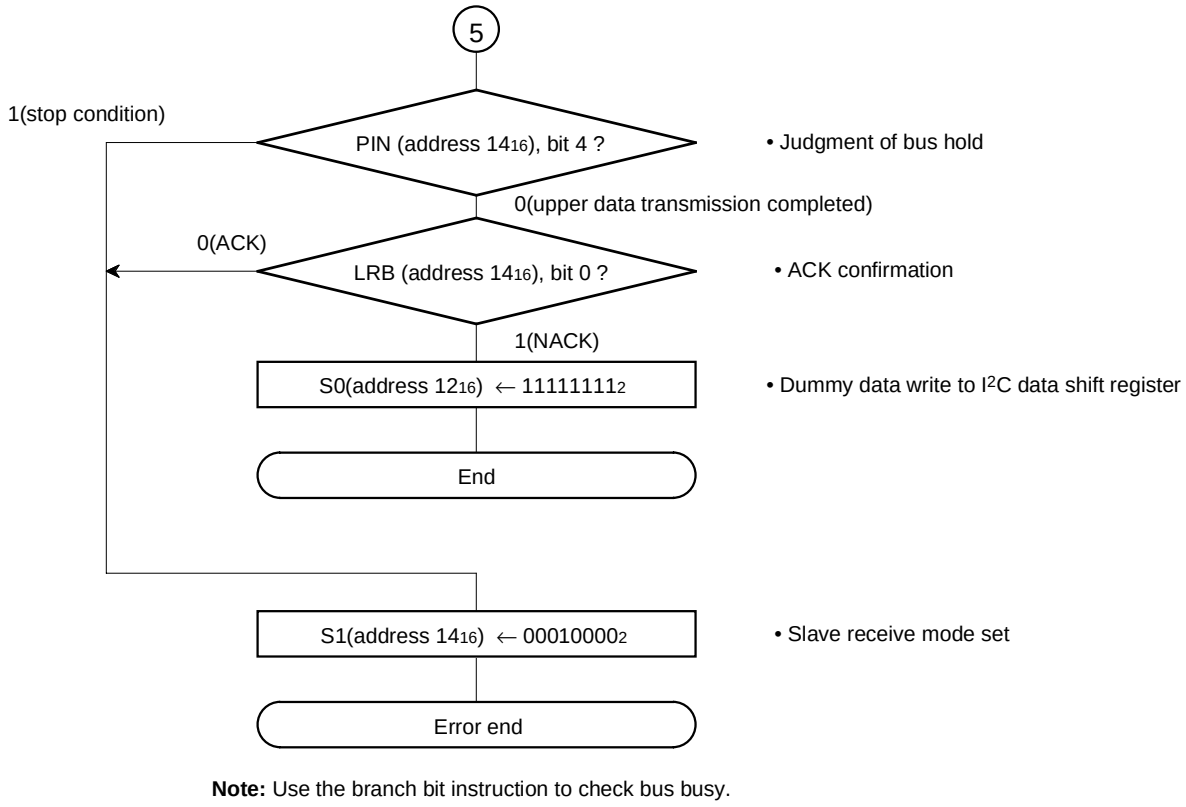


Fig. 2.5.29 Transmission process of lower data

⑤ **Transmission of upper data**

Confirm correct completion of the upper data transmission. The master returns the NACK toward the transmitted second-byte data, the upper data. Accordingly, confirm that the last received bit (LRB bit; bit 0 of address 0014_{16}) is "1".

After that, write dummy data to the I²C data shift register and wait for the interrupt of STOP condition.



Note: Use the branch bit instruction to check bus busy.

Fig. 2.5.30 Transmission process of upper data

⑥ **Reception of STOP condition**

Confirm that the STOP condition is correctly output and the bus is released.

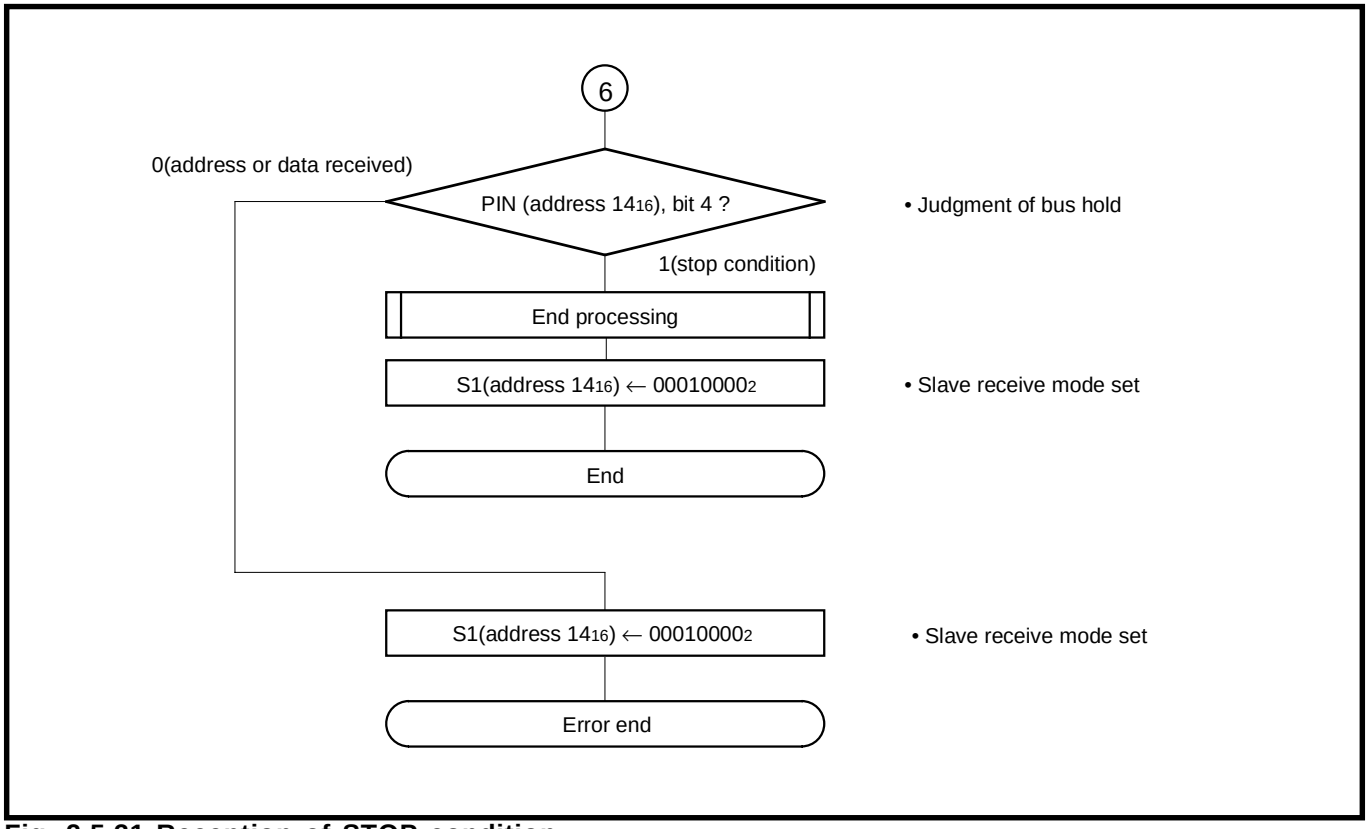


Fig. 2.5.31 Reception of STOP condition

2.5.7 Notes on multi-master I²C-BUS interface

(1) Read-modify-write instruction

Precautions for read-modify-write instructions, such as **SEB** and **CLB**, when used for any of the registers of the multi-master I²C-BUS interface, are described below.

① **I²C data shift register (S0: address 0012_{16})**
When executing the read-modify-write instruction for this register during transfer, data may become an unexpected value.

② **I²C address register (S0D: address 0013_{16})**
When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become an unexpected value.

● **Reason**
Because hardware changes the read/$\overline{\text{write}}$ bit (RWB) at detecting the STOP condition.

③ **I²C status register (S1: address 0014_{16})**
Do not execute the read-modify-write instruction for this register because all bits of this register are changed by hardware.

④ **I²C control register (S1D: address 0015_{16})**
When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become an unexpected value.

● **Reason**
Because hardware changes the bit counter (BC0 to BC2).

⑤ **I²C clock control register (S2: address 0016_{16})**
The read-modify-write instruction can be executed for this register.

⑥ **I²C START/STOP condition control register (S2D: address 0017_{16})**
The read-modify-write instruction can be executed for this register.

2.5 Multi-master I²C-BUS interface

(2) Procedure for generating START condition using multi-master

① Procedure example (The necessary conditions for the procedure are described in Items ② to ⑤ below).

```
        LDA #SLADR          (Take out slave address value)
        SEI                 (Disable interrupt)
        BBS 5, S1, BUSBUSY  (BB flag confirmation and branch process)
BUSFREE:
        STA S0              (Write slave address value)
        LDM #$F0, S1        (Trigger START condition generation)
        CLI                 (Enable interrupt)
         :
         :
BUSBUSY:
        CLI                 (Enable interrupt)
         :
         :
```

② Use "Branch on Bit Set" of "BBS 5, S1, –" for the BB flag confirmation and branch process.

③ Use "STA", "STX" or "STY" of the zero page addressing instruction for writing the slave address value to the I²C data shift register (S0: address 0012_{16}).

④ Execute the branch instruction of above ② and the store instruction of above ③ continuously shown the above procedure example.

⑤ Disable interrupts during the following three process steps:

- BB flag confirmation
- Write slave address value
- Trigger START condition generation

When the BB flag is in bus busy state, enable interrupts immediately.

(3) Procedure for generating RESTART condition

This procedure cannot be applied to M38867M8A and M38867E8A when the external memory is used and the bus cycle is extended by $\overline{\text{ONW}}$ function.

① Procedure example (The necessary conditions for the procedure are described in Items ② to ④ below). Execute the following procedure when the PIN bit is "0".

```
        LDM #$00, S1        (Select slave receive mode)
        LDA #SLADR          (Take out slave address value)
        SEI                 (Disable interrupt)
        STA S0              (Write slave address value)
        LDM #$F0, S1        (Trigger RESTART condition generation)
        CLI                 (Enable interrupt)
         :
         :
```

② Select the slave receive mode when the PIN bit is "0". Do not write "1" to the PIN bit. Neither "0" nor "1" is specified as input to the BB bit. The TRX bit becomes "0" and the SDA pin is released.

③ The SCL pin is released by writing the slave address value to the I²C data shift register.

④ Disable interrupts during the following two process steps:

- Write slave address value
- Trigger RESTART condition generation

(4) Writing to I²C status register

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

(5) Process of after STOP condition generating

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

(6) STOP condition input at 7th clock pulse

The SDA line may be held at LOW even if flag BB is set to "0" when all the following conditions are satisfied:

•The STOP condition is input at the 7th clock pulse while receiving a slave address or data.
•The clock pulse is continuously input.
•In the slave mode

Countermeasure:

Write dummy data to the I²C data shift register or reset the ES0 bit in the S1D register (ES0 = "L" → ES0 = "H") during a stop condition interrupt routine with flag BB = "0".

Note: Do not use the read-modify-write instruction at this time. Furthermore, when the ES0 bit is set to "0", the SDA pin becomes a general-purpose port ; so that the port must be set to input mode or output "H".

(7) ES0 bit switch

In standard clock mode when SSC = "00010_2" or in high-speed clock mode, flag BB may switch to "1" if ES0 bit is set to "1" when SDA is "L".

Countermeasure:

Set ES0 to "1" when SDA is "H".

2.6 PWM

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

2.6.1 Memory map

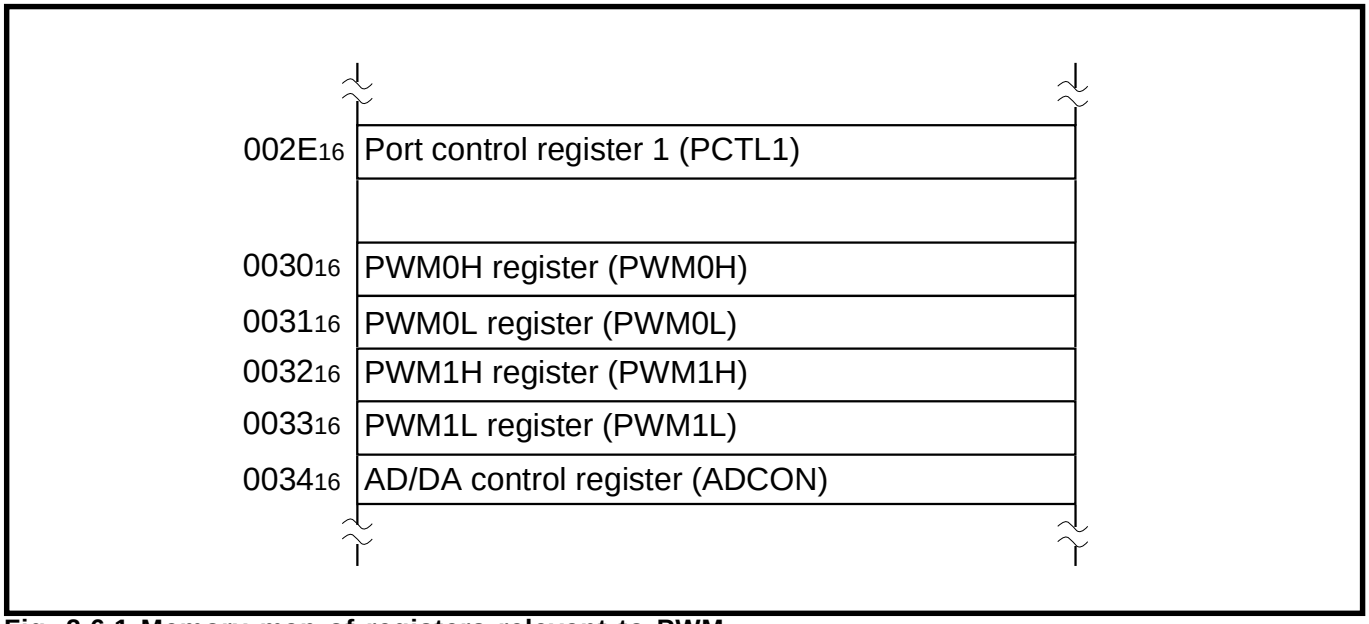


Fig. 2.6.1 Memory map of registers relevant to PWM

2.6.2 Relevant registers

Port control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Port control register 1 (PCTL1) [Address 2E 16]

B	Name	Function	At reset	R	W
0	$P0_0$–$P0_3$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	×	○
1	$P0_4$–$P0_7$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	×	○
2	$P1_0$–$P1_3$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	×	○
3	$P1_4$–$P1_7$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	×	○
4	$P3_0$–$P3_3$ pull-up control bit	0: No pull-up 1: Pull-up	0	×	○
5	$P3_4$–$P3_7$ pull-up control bit	0: No pull-up 1: Pull-up	0	×	○
6	PWM_0 enable bit	0: PWM_0 output disabled 1: PWM_0 output enabled	0	×	○
7	PWM_1 enable bit	0: PWM_1 output disabled 1: PWM_1 output enabled	0	×	○

Fig. 2.6.2 Structure of Port control register 1

PWM0H register

b7 b6 b5 b4 b3 b2 b1 b0

PWM0H register (PWM0H) [Address : 30_{16}]

B	Function	At reset	R	W
0	• The high-order 8 bits of the PWM0H output data are set. • At writing: A written data is transferred to PWM0 latch at every sub-period (64 μs). ($f(X_{IN})$ = 8 MHz) • At reading: The contents of the PWM0H register are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 2.6.3 Structure of PWM0H register

PWM0L register

b7 b6 b5 b4 b3 b2 b1 b0

PWM0L register (PWM0L) [Address : 31_{16}]

B	Function	At reset	R	W
0	• The low-order 6 bits of the PWM0L output data are set. • At writing: A written data is transferred to PWM0 latch at every repetitive period (4096 μs). ($f(X_{IN})$ = 8 MHz) • At reading: The low-order 6 bits of PWM0 latch are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is “0”.	?	○	×
7	• The completion of transfer to the PWM0 latch is indicated. 0: Transfer completed. 1: Not transferred. • At writing: This bit is set to “1”.	?	○	×

Fig. 2.6.4 Structure of PWM0L register

PWM1H register

b7 b6 b5 b4 b3 b2 b1 b0

PWM1H register (PWM1H) [Address : 32_{16}]

B	Function	At reset	R	W
0	• The high-order 8 bits of the PWM1H output data are set. • At writing: A written data is transferred to PWM1 latch at every sub-period (64 μs). ($f(X_{IN})$ = 8 MHz) • At reading: The contents of the PWM1H register are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 2.6.5 Structure of PWM1H register

PWM1L register

b7 b6 b5 b4 b3 b2 b1 b0

PWM1L register (PWM1L) [Address : 33_{16}]

B	Function	At reset	R	W
0	• The low-order 6 bits of the PWM1L output data are set. • At writing: A written data is transferred to PWM1 latch at every repetitive period (4096 μs). ($f(X_{IN})$ = 8 MHz) • At reading: The low-order 6 bits of PWM1 latch are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is “0”.	?	○	×
7	• The completion of transfer to the PWM1 latch is indicated. 0: Transfer completed. 1: Not transferred. • At writing: This bit is set to “1”.	?	○	×

Fig. 2.6.6 Structure of PWM1L register

2.6 PWM

AD/DA control register

b7 b6 b5 b4 b3 b2 b1 b0

AD/DA control register (ADCON) [Address : 34_{16}]

B	Name	Function	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					
2					
3	AD conversion completion bit	0 : Conversion in progress 1 : Conversion completed	1	○	○
4	PWM_0 output pin selection bit	0: $P5_6/PWM_{01}$ 1: $P3_0/PWM_{00}$	0	○	○
5	PWM_1 output pin selection bit	0: $P5_7/PWM_{11}$ 1: $P3_1/PWM_{10}$	0	○	○
6	DA_1 output enable bit	0: DA_1 output disabled 1: DA_1 output enabled	0	○	○
7	DA_2 output enable bit	0: DA_2 output disabled 1: DA_2 output enabled	0	○	○

Fig. 2.6.7 Structure of AD/DA control register

2.6.3 PWM application example

(1) Control of VS tuner

Figure 2.6.8 shows a connection diagram, and Figure 2.6.9 shows the setting of relevant registers.

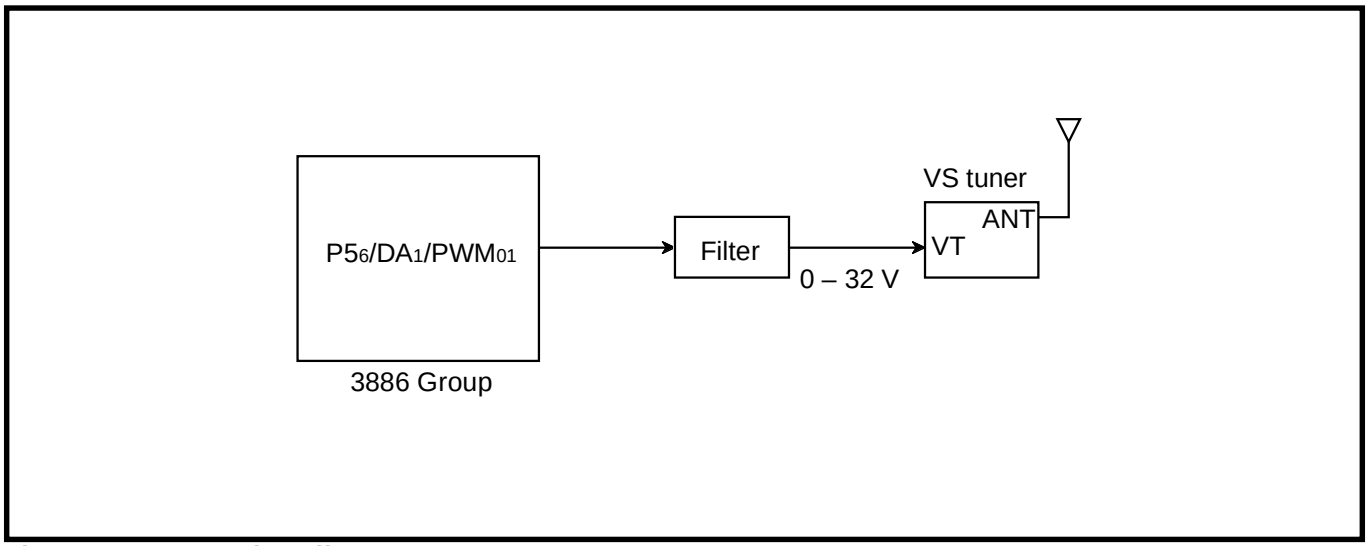


Fig. 2.6.8 Connection diagram

APPLICATION

2.6 PWM

Outline: • Control of VS tuner by using the 14-bit resolution PWM_0 output function
• $f(X_{IN})$ = 8 MHz

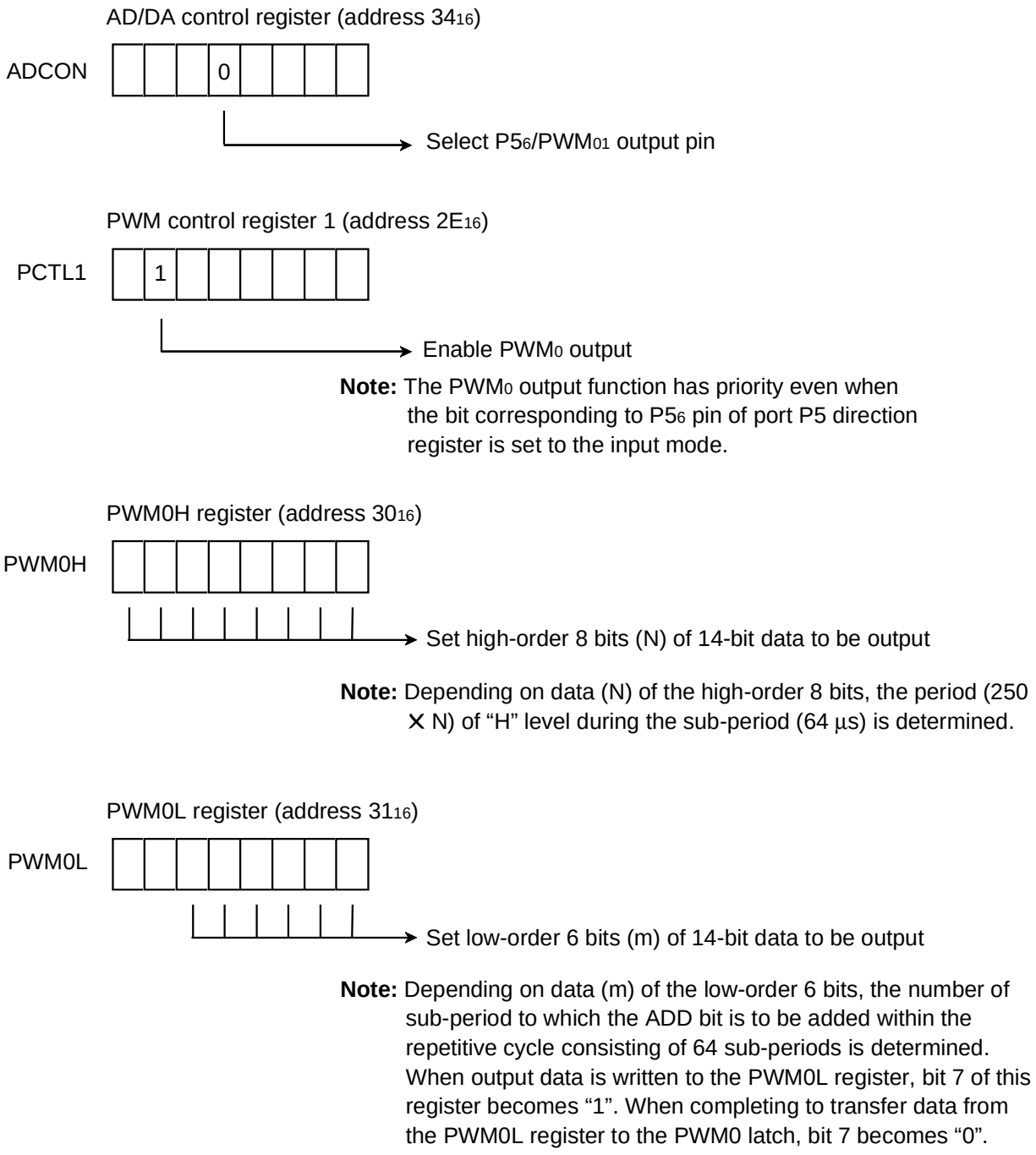


Fig. 2.6.9 Relevant registers setting

Control procedure: PWM waveform is output to the external by setting relevant registers shown Figure 2.6.9. This PWM_0 output is integrated through the low pass filter and converted into DC signals for control of the VS tuner.
Figure 2.6.10 shows the control procedure.

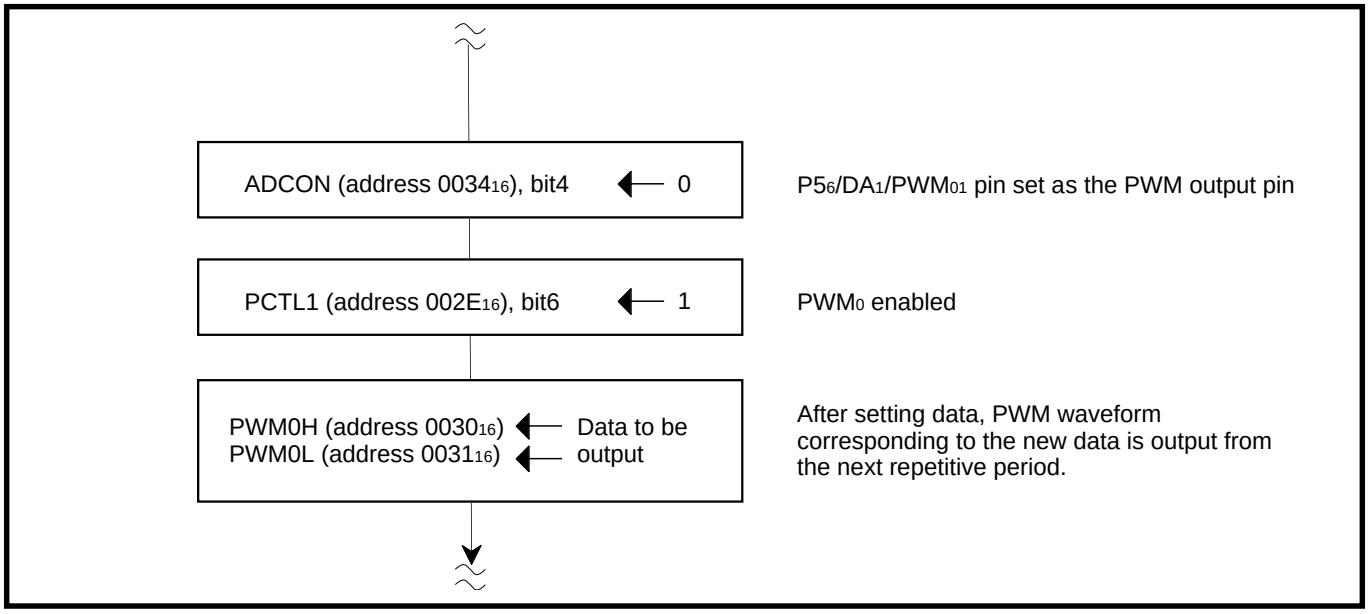


Fig. 2.6.10 Control procedure

2.6.4 Notes on PWM

- For PWM_0 output, "L" level is output first.
- After data is set to the PWM0L and the PWM0H registers, PWM waveform corresponding to the new data is output from next repetitive period.

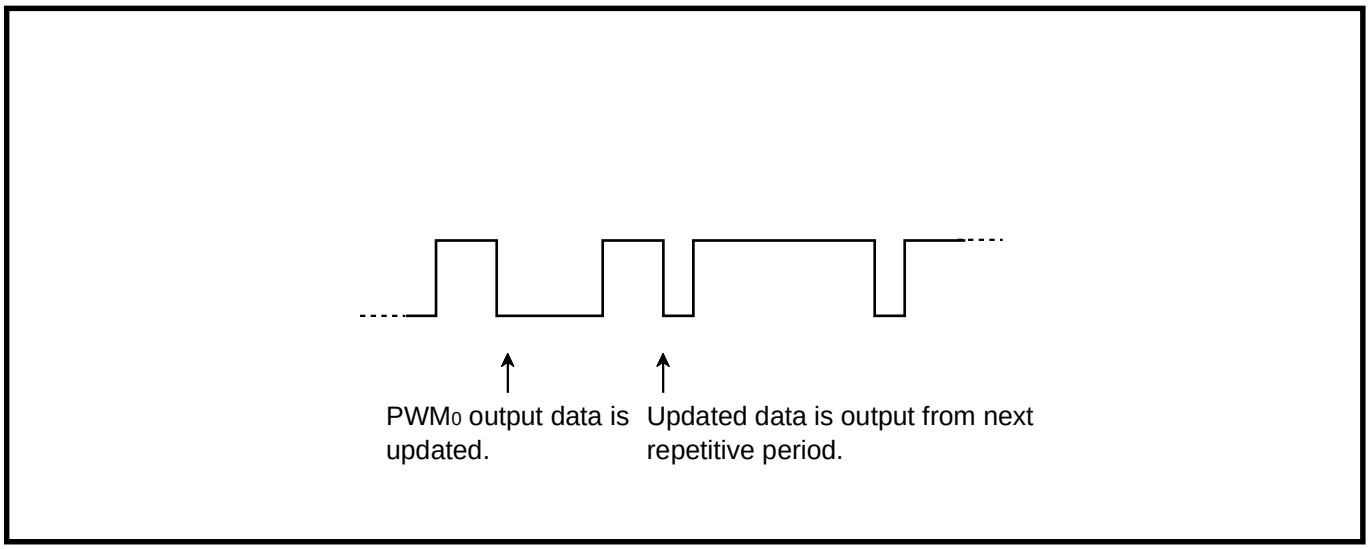


Fig. 2.6.11 PWM_0 output

2.7 A-D converter

This paragraph explains the registers setting method and the notes relevant to the A-D converter.

2.7.1 Memory map

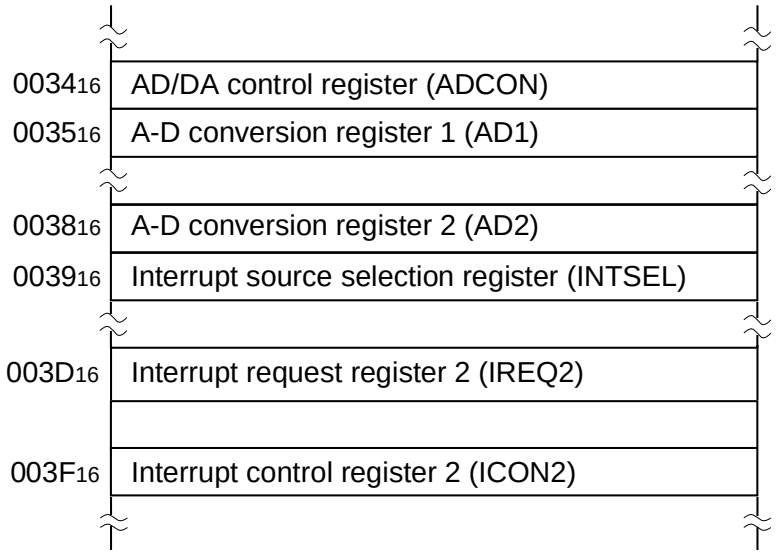


Fig. 2.7.1 Memory map of registers relevant to A-D converter

2.7.2 Relevant registers

AD/DA control register

b7 b6 b5 b4 b3 b2 b1 b0

AD/DA control register (ADCON) [Address : 34_{16}]

B	Name	Function	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					
2					
3	AD conversion completion bit	0 : Conversion in progress 1 : Conversion completed	1	○	○
4	PWM_0 output pin selection bit	0: $P5_6/PWM_{01}$ 1: $P3_0/PWM_{00}$	0	○	○
5	PWM_1 output pin selection bit	0: $P5_7/PWM_{11}$ 1: $P3_1/PWM_{10}$	0	○	○
6	DA_1 output enable bit	0: DA_1 output disabled 1: DA_1 output enabled	0	○	○
7	DA_2 output enable bit	0: DA_2 output disabled 1: DA_2 output enabled	0	○	○

Fig. 2.7.2 Structure of AD/DA control register

A-D conversion register 1

b7 b6 b5 b4 b3 b2 b1 b0

A-D conversion register 1 (AD1) [Address : 35_{16}]

B	Function	At reset	R	W
0	The read-only register in which the A-D conversion's results are stored. < 8-bit read> b7 b0 b9 b8 b7 b6 b5 b4 b3 b2 < 10-bit read> b7 b0 b7 b6 b5 b4 b3 b2 b1 b0	?	○	×
1		?	○	×
2		?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

Fig. 2.7.3 Structure of A-D conversion register 1

A-D conversion register 2

b7 b6 b5 b4 b3 b2 b1 b0

A-D conversion register 2 (AD2) [Address : 38_{16}]

B	Name	Function	At reset	R	W
0		The read-only register in which the A-D conversion's results are stored.	?	○	×
1		< 10-bit read> b7 b0 b9 b8	?	○	×
2		Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".	0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7	Conversion mode selection bit	0: 10-bit A-D mode 1: 8-bit A-D mode	0	○	○

Fig. 2.7.4 Structure of A-D conversion register 2

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register (INTSEL) [Address 39_{16}]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt source selection bit	0 : INT_0 interrupt 1 : Input buffer full interrupt	0	○	○
1	INT_1/output buffer empty interrupt source selection bit	0 : INT_1 interrupt 1 : Output buffer empty interrupt	0	○	○
2	Serial I/O1 transmit/S_{CL},S_{DA} interrupt source selection bit	0 : Serial I/O1 transmit interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
3	$CNTR_0$/S_{CL},S_{DA} interrupt source selection bit	0 : $CNTR_0$ interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
4	Serial I/O2/I^2C interrupt source selection bit	0 : Serial I/O2 interrupt ✻2 1 : I^2C interrupt	0	○	○
5	INT2/I^2C interrupt source selection bit	0 : INT_2 interrupt ✻2 1 : I^2C interrupt	0	○	○
6	$CNTR_1$/key-on wake-up interrupt source selection bit	0 : $CNTR_1$ interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○
7	AD converter/key-on wake-up interrupt source selection bit	0 : A-D converter interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○

✻1: Do not write "1" to these bits simultaneously.
✻2: Do not write "1" to these bits simultaneously.
✻3: Do not write "1" to these bits simultaneously.

Fig. 2.7.5 Structure of Interrupt source selection register

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2 (IREQ2) [Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0/S_{CL}$, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	$CNTR_1$/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT_4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.7.6 Structure of Interrupt request register 2

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 2 (ICON2) [Address : $3F_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0/S_{CL}$, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	$CNTR_1$/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	INT_4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to "0".		0	○	○

Fig. 2.7.7 Structure of Interrupt control register 2

2.7.3 A-D converter application examples

(1) Conversion of analog input voltage

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

Figure 2.7.8 shows a connection diagram, and Figure 2.7.9 shows the relevant registers setting.

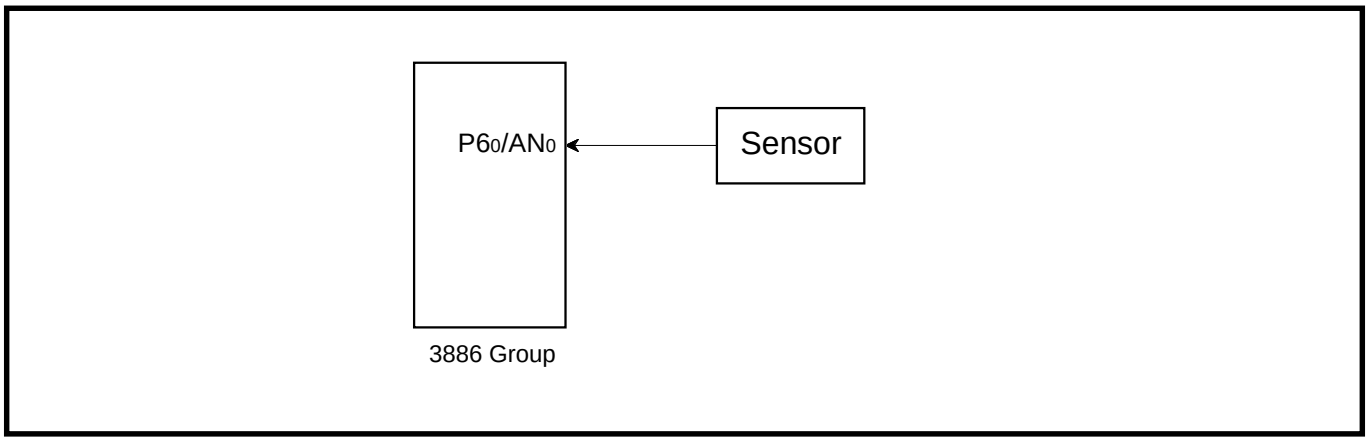


Fig. 2.7.8 Connection diagram

Specifications : •The analog input voltage input from a sensor is converted to digital values.
•P6₀/AN₀ pin is used as an analog input pin.

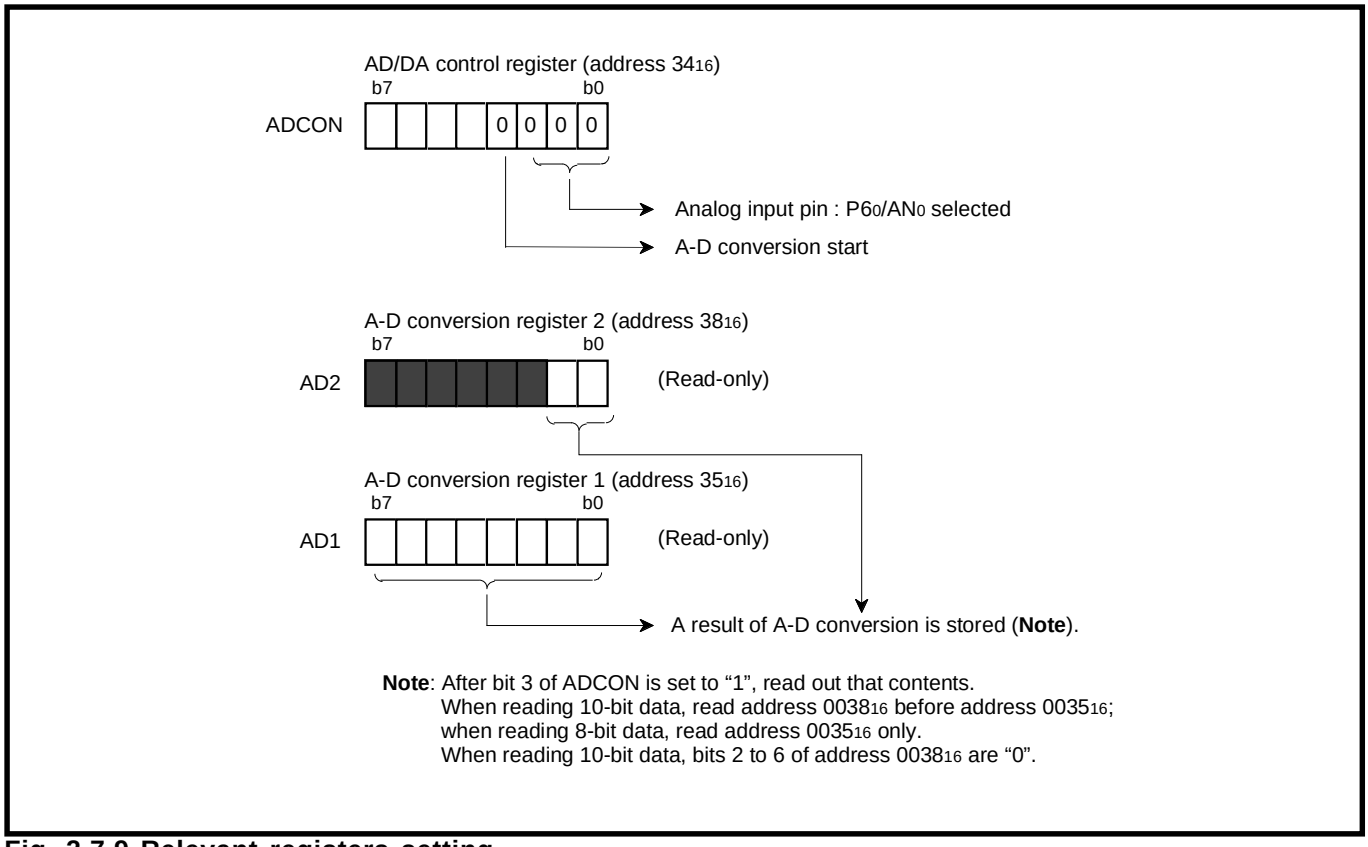


Fig. 2.7.9 Relevant registers setting

An analog input signal from a sensor is converted to the digital value according to the relevant registers setting shown by Figure 2.7.9. Figure 2.7.10 shows the control procedure for 8-bit read, and Figure 2.7.11 shows the control procedure for 10-bit read.

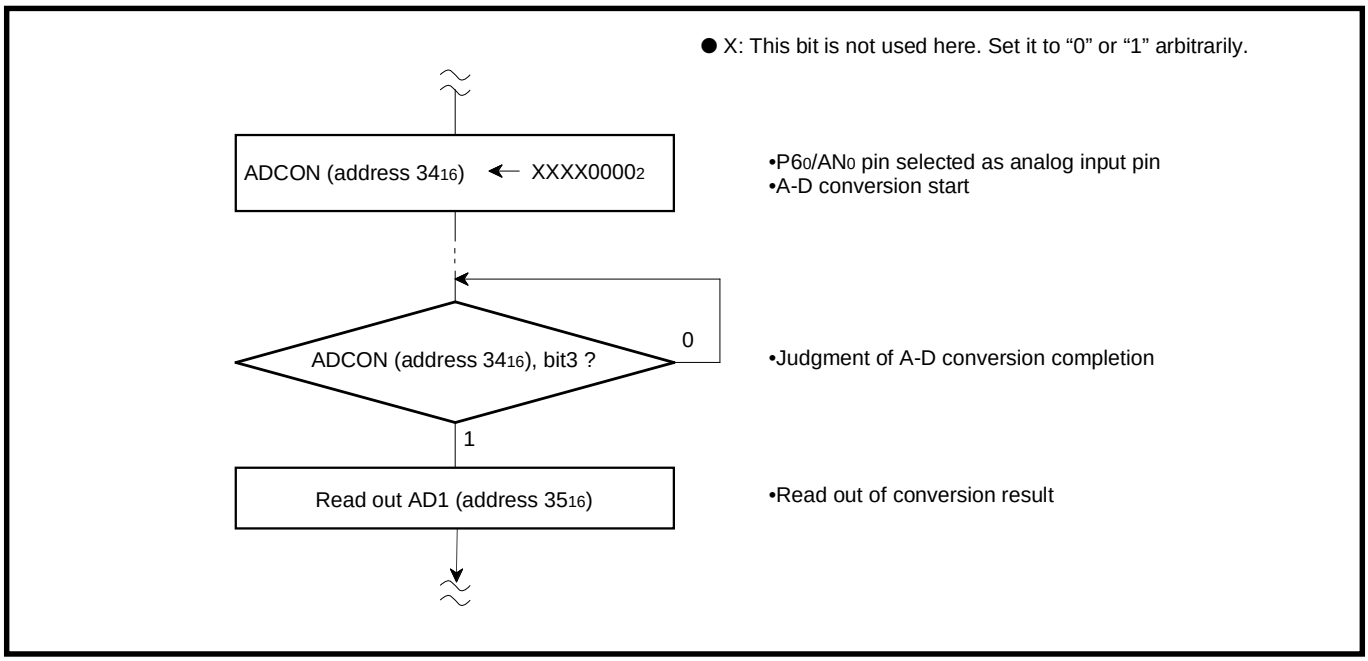


Fig. 2.7.10 Control procedure for 8-bit read

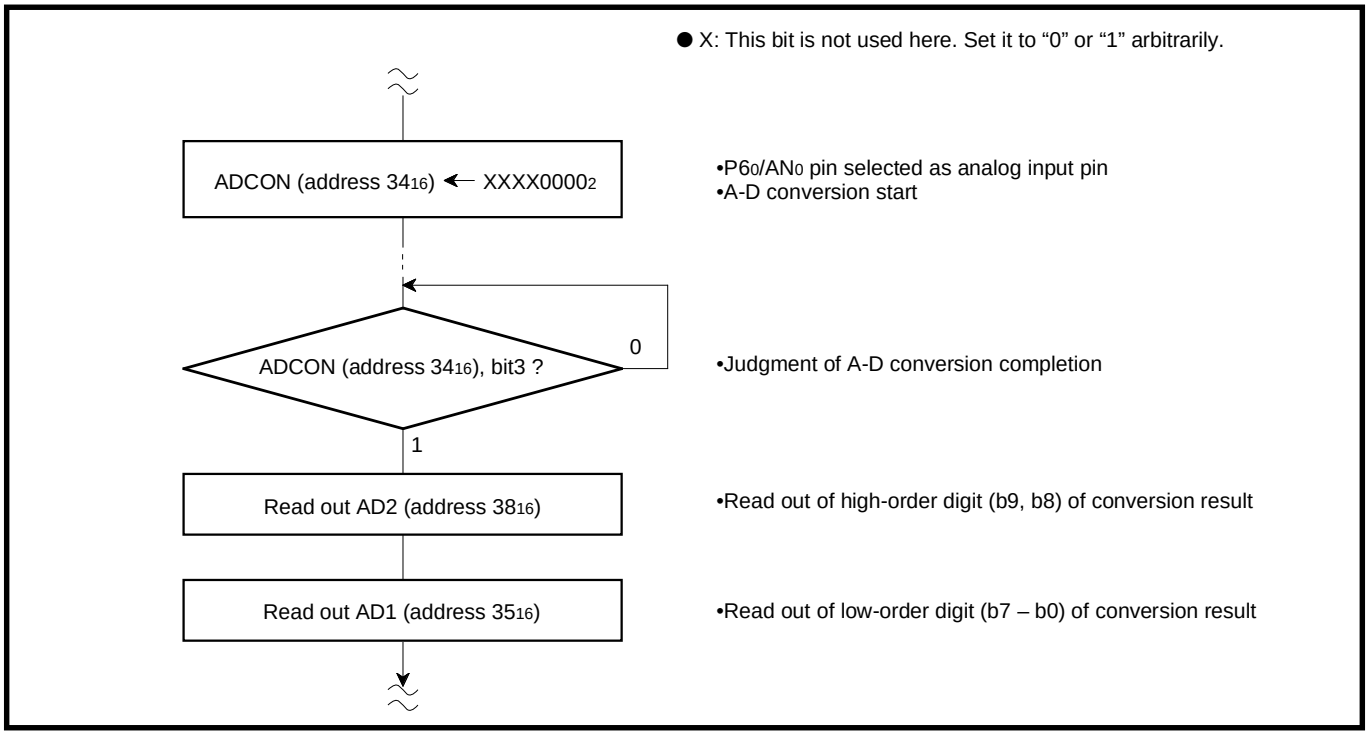


Fig. 2.7.11 Control procedure for 10-bit read

2.7 A-D converter

2.7.4 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 an A-D converter power source pin. Regardless of using the A-D conversion function or not, connect them as following :

• AVSS : Connect to the VSS line

● **Reason**

If the AVSS pin id 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

• Do not execute the **STP** instruction

2.8 D-A Converter

This paragraph explains the registers setting method and the notes relevant to the D-A converter.

2.8.1 Memory map

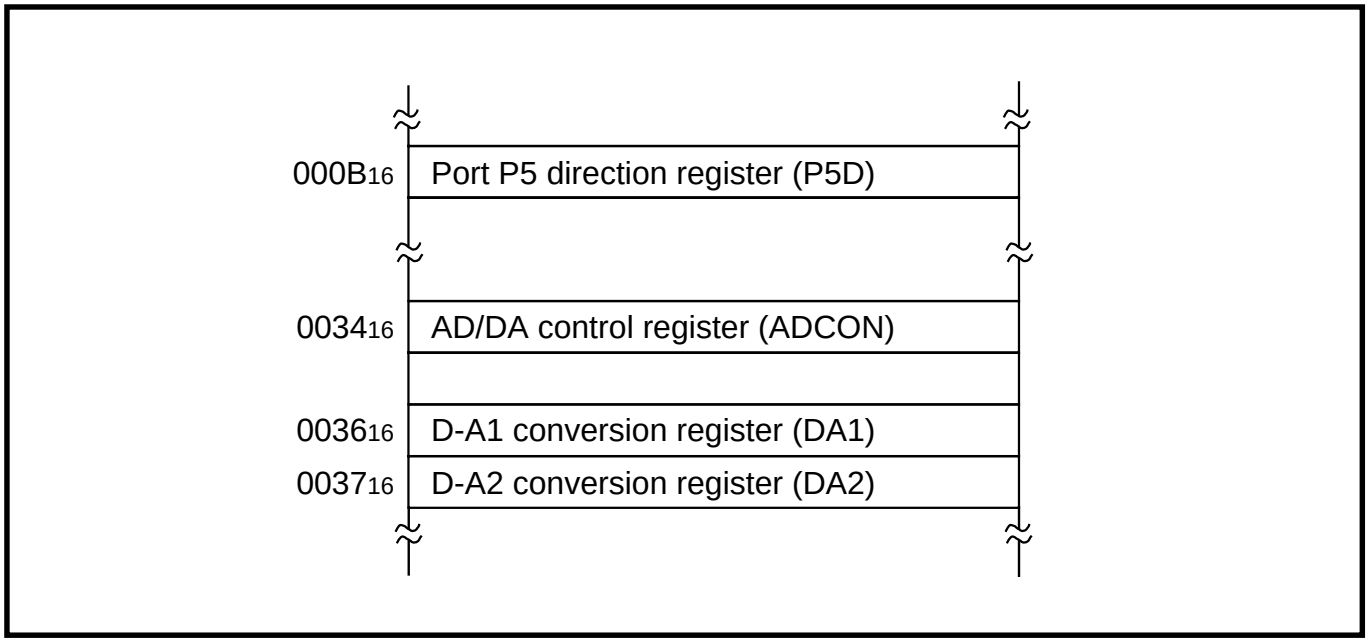


Fig. 2.8.1 Memory map of registers relevant to D-A converter

2.8.2 Relevant registers

Port P5 direction register

b7 b6 b5 b4 b3 b2 b1 b0

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

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

Fig. 2.8.2 Structure of Port P5 direction register

AD/DA control register

b7 b6 b5 b4 b3 b2 b1 b0

AD/DA control register
(ADCON: address 34_{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	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
4	PWM_0 output pin selection bit	0: $P5_6/PWM_{01}$ 1: $P3_0/PWM_{00}$	0	○	○
5	PWM_1 output pin selection bit	0: $P5_7/PWM_{11}$ 1: $P3_1/PWM_{10}$	0	○	○
6	DA_1 output enable bit	0: DA_1 output disabled 1: DA_1 output enabled	0	○	○
7	DA_2 output enable bit	0: DA_2 output disabled 1: DA_2 output enabled	0	○	○

Fig. 2.8.3 Structure of AD/DA control register

D-Ai conversion register

b7 b6 b5 b4 b3 b2 b1 b0

D-Ai conversion register (i = 1, 2)
(DAi: addresses 36_{16}, 37_{16})

b	Functions	At reset	R	W
0	This is D-A output value stored bits. This is write exclusive register.	0	×	○
1		0	×	○
2		0	×	○
3		0	×	○
4		0	×	○
5		0	×	○
6		0	×	○
7		0	×	○

Fig. 2.8.4 Structure of D-Ai converter register

2.8.3 D-A converter application example

(1) Speaker output volume modulation

Outline: The volume of a speaker output is modulated by using D-A converter.

Specifications: •Timer X modulates the period of sound for the pitch interval, so that a fixed pitch ("la": approx. 440 Hz) can be output. Modulating the amplitude with the D-A output value controls the volume.
•Use f(X_{IN}) = 6 MHz.
•Use DA1 ($P5_6/DA_1$ pin) as D-A converter.

Figure 2.8.5 shows a peripheral circuit example and Figure 2.8.6 shows a speaker output example. Figure 2.8.7 shows the relevant registers setting.

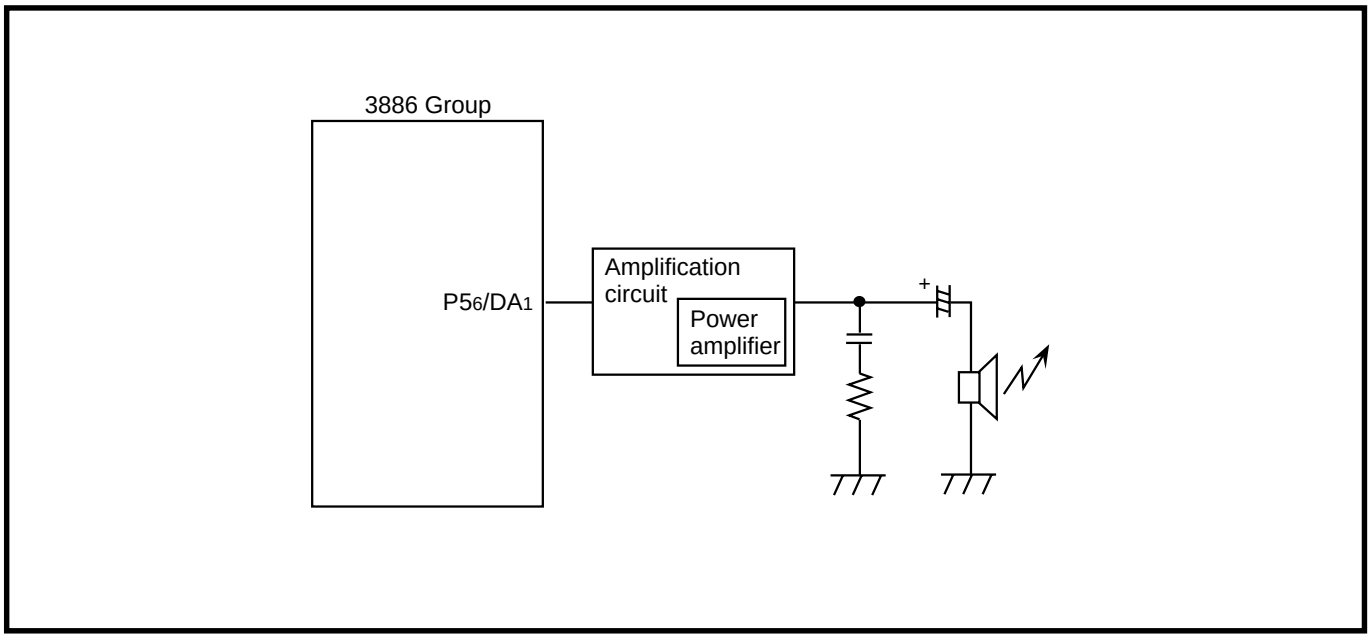


Fig. 2.8.5 Peripheral circuit example

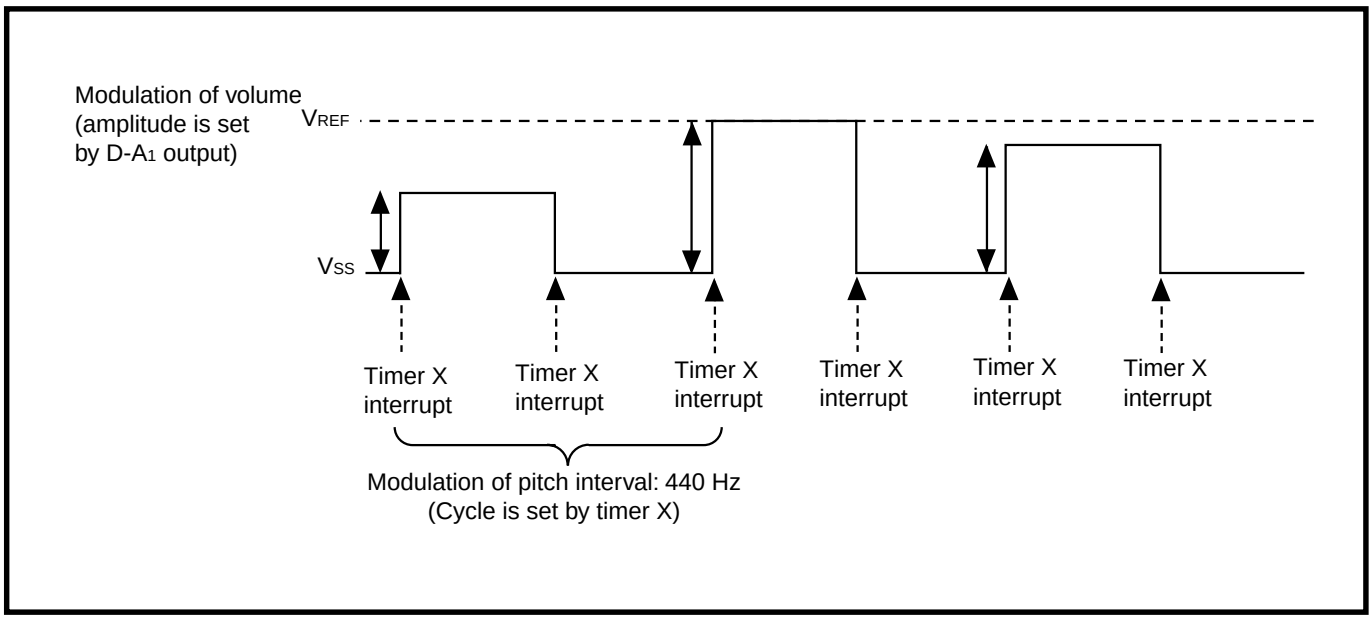


Fig. 2.8.6 Speaker output example

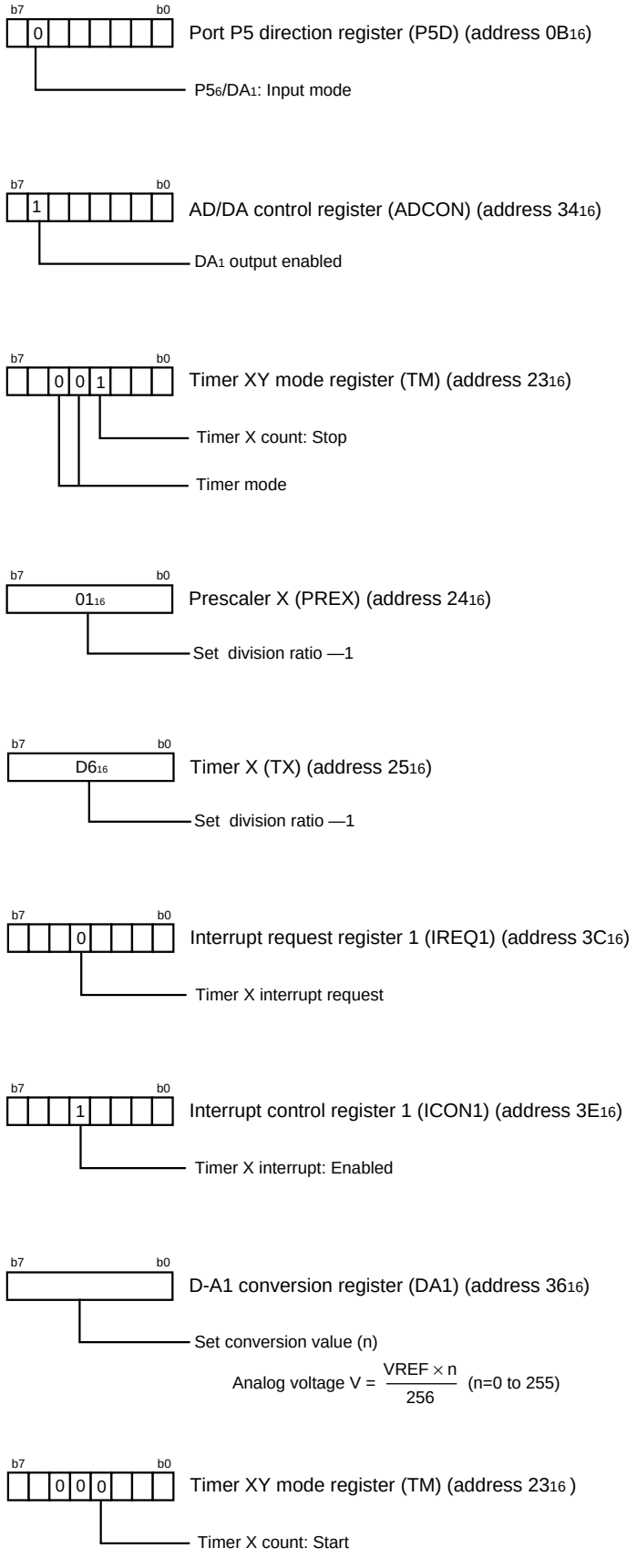


Fig. 2.8.7 Relevant registers setting

2.8 D-A Converter

When the registers are set as shown in Figure 2.8.7, the speaker output volume is modulated by the D-A output value. Figure 2.8.8 shows the control procedure.

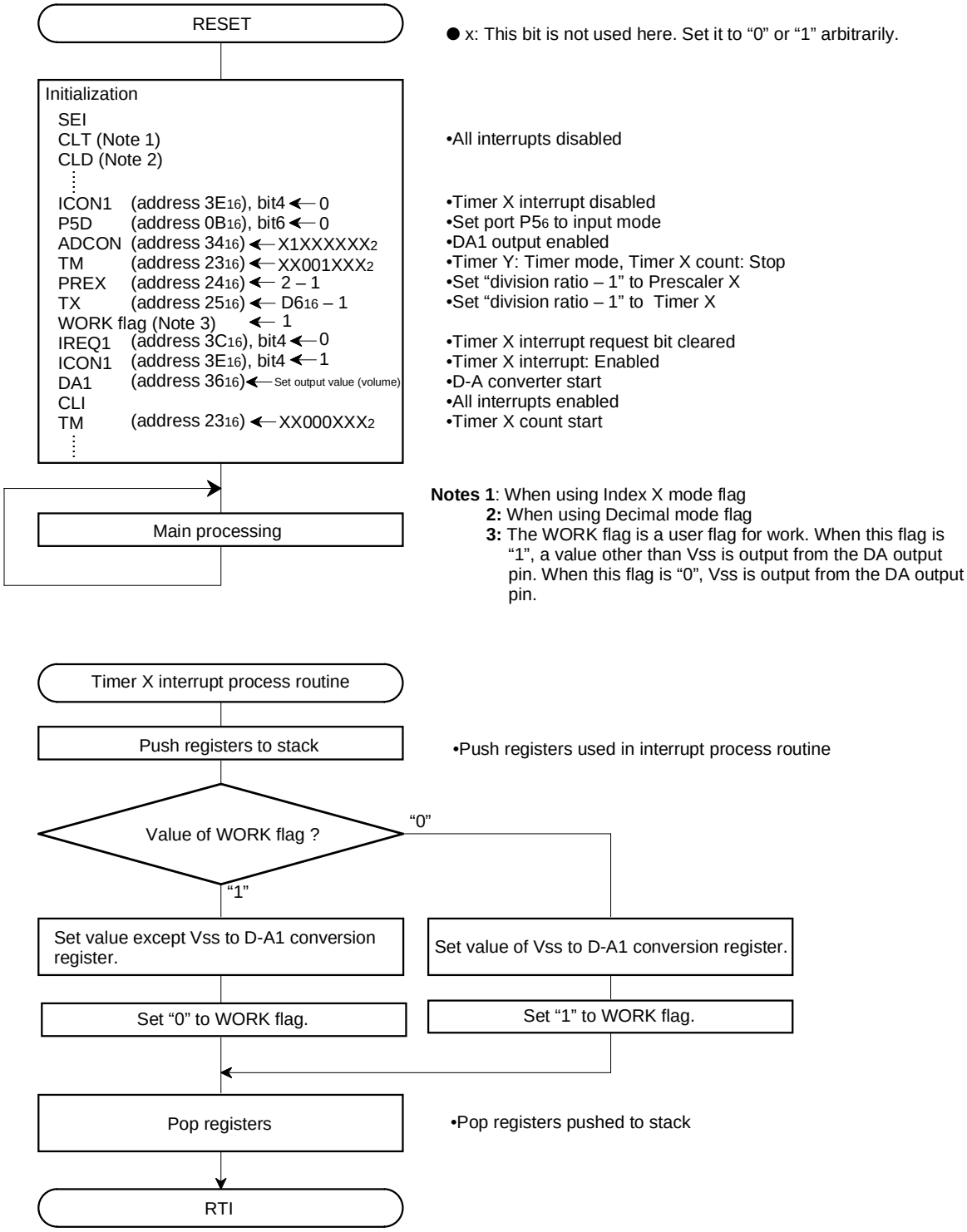


Fig. 2.8.8 Control procedure

2.8.4 Notes on D-A converter

(1) Vcc when using D-A converter

The D-A converter accuracy when Vcc is 4.0 V or less differs from that of when Vcc is 4.0 V or more. When using the D-A converter, we recommend using a Vcc of 4.0 V or more.

(2) D-Ai conversion register when not using D-A converter

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

2.9 Bus interface

This paragraph explains the registers setting method and the programming examples relevant to the bus interface.

2.9.1 Memory map

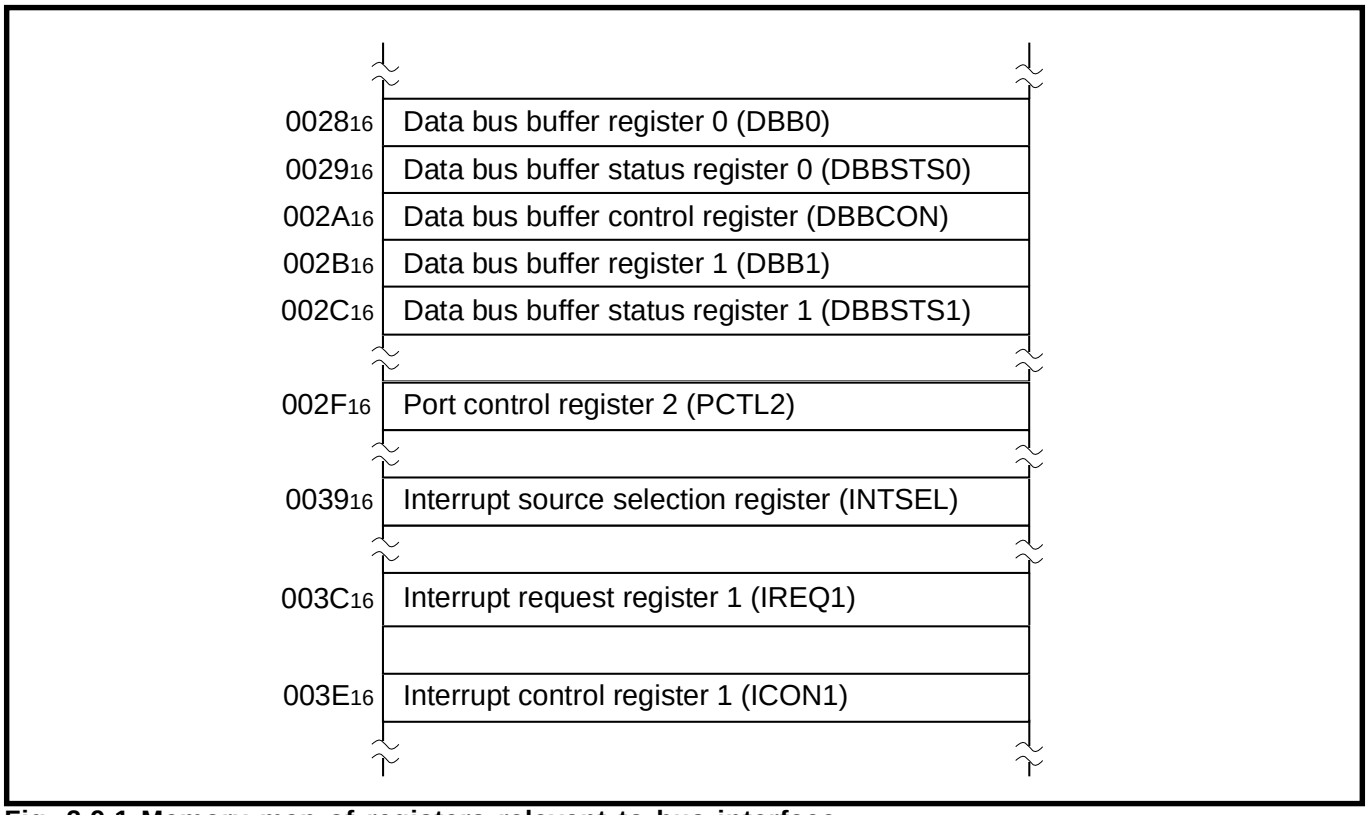


Fig. 2.9.1 Memory map of registers relevant to bus interface

2.9.2 Relevant registers

Data bus buffer register i

b7 b6 b5 b4 b3 b2 b1 b0

Data bus buffer register i (DBBi) (i=0, 1) [Address : $28_{16}/2B_{16}$]

B	Name	Function	At reset	R	W
0	• Buffer register to write output data and read input data. • at write: data is written to output data buffer register. • at read: the contents of input data buffer register are read out.		?	○	○
1			?	○	○
2			?	○	○
3			?	○	○
4			?	○	○
5			?	○	○
6			?	○	○
7			?	○	○

Note: Output data bus buffer and input data bus buffer are assigned to the same address.
The contents of output data bus buffer register cannot be read out.
Writing to input data bus buffer register is disabled.

Fig. 2.9.2 Structure of Data bus buffer register i

Data bus buffer status register i

b7 b6 b5 b4 b3 b2 b1 b0

Data bus buffer status register i (DBBSTSi) (i=0, 1) [Address $29_{16}/2C_{16}$]

B	Name	Function	At reset	R	W
0	Output buffer full flag i	0 : Buffer empty 1 : Buffer full	0	○	×
1	Input buffer full flag i	0 : Buffer empty 1 : Buffer full	0	○	×
2	User definable flags (This flag can be defined by user freely.)		0	○	○
3	A_{0i} flag	This flag indicates the condition of A_{0i} status when the IBF_i flag is set.	0	○	×
4	User definable flags (This flag can be defined by user freely.)		0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Fig. 2.9.3 Structure of Data bus buffer status register i

Data bus buffer control register

b7 b6 b5 b4 b3 b2 b1 b0

Data bus buffer control register (DBBCON) [Address $2A_{16}$]

B	Name	Function	At reset	R	W
0	Data bus buffer enable bit	0 : $P5_0$–$P5_3$, P8 I/O port 1 : Data bus buffer enabled	0	○	○
1	Data bus buffer function selection bit	0 : Single data bus buffer mode ($P4_7$ functions as I/O port.) 1 : Double data bus buffer mode ($P4_7$ functions as $\overline{S_1}$ input.)	0	○	○
2	OBF_0 output selection bit	0 : OBF_{00} valid 1 : OBF_{01} valid	0	○	○
3	OBF_{00} output enable bit	0 : $P4_2$ functions as port I/O pin. 1 : $P4_2$ functions as OBF_{00} output pin.	0	○	○
4	OBF_{01} output enable bit	0 : $P4_3$ functions as port I/O pin. 1 : $P4_3$ functions as OBF_{01} output pin.	0	○	○
5	OBF_{10} output enable bit	0 : $P4_6$ functions as port I/O pin. 1 : $P4_6$ functions as OBF_{10} output pin.	0	○	○
6	Input level selection bit	0 : CMOS level input 1 : TTL level input	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.9.4 Structure of Data bus buffer control register

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register (INTSEL) [Address 39_{16}]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt source selection bit	0 : INT_0 interrupt 1 : Input buffer full interrupt	0	○	○
1	INT_1/output buffer empty interrupt source selection bit	0 : INT_1 interrupt 1 : Output buffer empty interrupt	0	○	○
2	Serial I/O1 transmit/S_{CL},S_{DA} interrupt source selection bit	0 : Serial I/O1 transmit interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
3	$CNTR_0$/S_{CL},S_{DA} interrupt source selection bit	0 : $CNTR_0$ interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
4	Serial I/O2/I^2C interrupt source selection bit	0 : Serial I/O2 interrupt ✻2 1 : I^2C interrupt	0	○	○
5	INT2/I^2C interrupt source selection bit	0 : INT_2 interrupt ✻2 1 : I^2C interrupt	0	○	○
6	$CNTR_1$/key-on wake-up interrupt source selection bit	0 : $CNTR_1$ interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○
7	AD converter/key-on wake-up interrupt source selection bit	0 : A-D converter interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○

✻1: Do not write “1” to these bits simultaneously.
✻2: Do not write “1” to these bits simultaneously.
✻3: Do not write “1” to these bits simultaneously.

Fig. 2.9.5 Structure of Interrupt source selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT_1/output buffer empty interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.9.6 Structure of Interrupt request register 1

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1 (ICON1) [Address : $3E_{16}$]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1/output buffer empty interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.9.7 Structure of Interrupt control register 1

Port control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Port control register 2 (PCTL2) [Address 2F 16]

B	Name	Function	At reset	R	W
0	P4 input level selection bit ($P4_2$–$P4_6$)	0: CMOS level input 1: TTL level input	0	×	○
1	P7 input level selection bit ($P7_0$–$P7_5$)	0: CMOS level input 1: TTL level input	0	×	○
2	P4 output structure selection bit ($P4_2$, $P4_3$, $P4_4$, $P4_6$)	0: CMOS 1: N-channel open-drain	0	×	○
3	P8 function selection bit	0: Port P8/Port P8 direction register 1: Port P4 input register/Port P7 input register	0	×	○
4	INT_2, INT_3, INT_4 interrupt switch bit	0: INT_{20}, INT_{30}, INT_{40} interrupt 1: INT_{21}, INT_{31}, INT_{41} interrupt	0	×	○
5	Timer Y count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ in low-speed mode) 1: $f(X_{CIN})$	0	×	○
6	Oscillation stabilizing time set after STP instruction released bit	0: Automatic set "01_{16}" to timer 1 and "FF_{16}" to prescaler 12 1: No automatic set	0	×	○
7	Port output $P4_2$/$P4_3$ clear function selection bit	0: Only software clear 1: Software clear and output data bus buffer 0 reading (system bus side)	0	×	○

Fig. 2.9.8 Structure of Port control register 2

2.9.3 Bus interface overview

The 3886 group has the built-in bus interface of two bytes to activate itself as a slave microcomputer. A slave microcomputer is the microcomputer which is operated owing to the host CPU's indication. Data is asynchronously transmitted/received between the host CPU and the slave microcomputer, through the bus interface of these two bytes. Accordingly, the slave microcomputer can be treated as well as two general peripheral LSIs on the host CPU side. Consequently, it is easy to change its function by updating the slave's program.

The performance overview of 3886 group's built-in bus interface is as follows:

- 8-bit data bus
- Built-in data bus buffer of two levels for input and output each
- Possible externally to output input/output buffer state as status.

Figure 2.9.9 shows the bus interface block diagram.

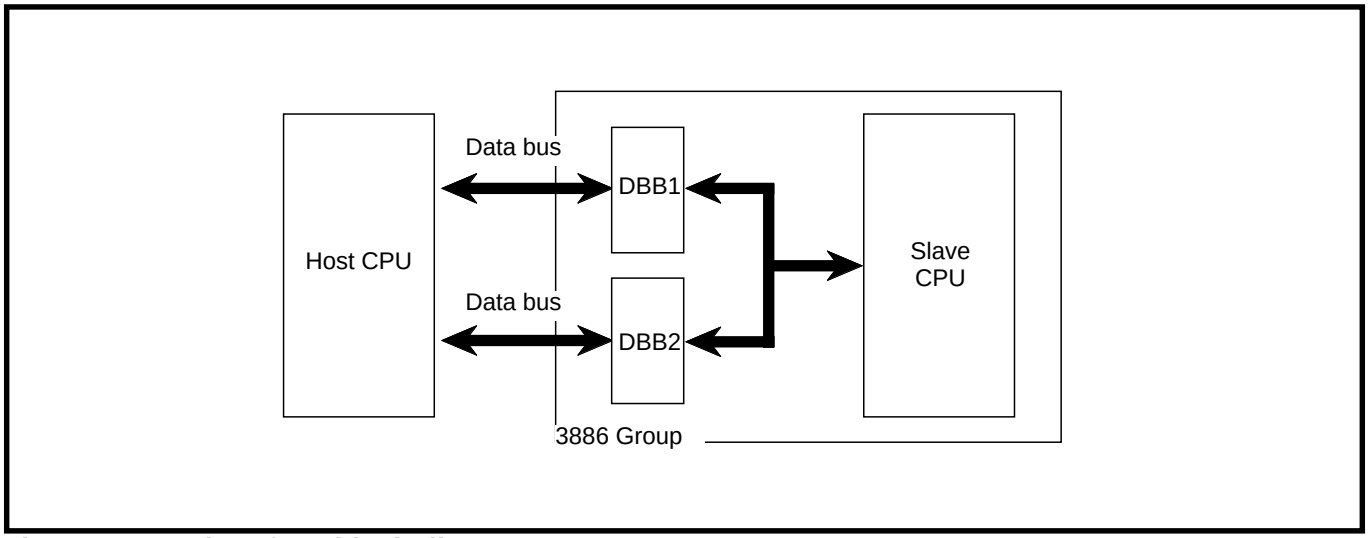


Fig. 2.9.9 Bus interface block diagram

2.9 Bus interface

2.9.4 Input/Output operation

(1) Input operation

The bus interface input operation is explained as the following:

① When the logical OR of $\overline{Si}$ (i = 0, 1) and $\overline{W}$ is "0", the data bus status is latched into the data bus buffer register i (DBBi) at the rising of $\overline{W}$ input signal.

② When the data is latched into the input data bus buffer register i, the IBFi flag of the data bus buffer status register i is simultaneously set to "1".

③ When the IBFi flag is set to "1", the input buffer full interrupt request occurs and the input buffer full interrupt request bit is set to "1".

④ At the timing ③, the A0 level is stored into bit 3 of the data bus buffer status register i. Bit 3 indicates that the contents of the input data bus buffer register i are data or a command.

(2) Output operation

The bus interface output operation is explained as the following:

① Writing data to the DBBi sets the OBFi flag of the data bus status register i to "1".

② When the logical OR of $\overline{Si}$, $\overline{R}$ and A0 is "0", the contents of the output data bus buffer register i are output on the system bus and the OBFi flag is simultaneously cleared to "0".

③ At the rising of the $\overline{R}$ input signal, the output buffer empty interrupt request occurs and the output buffer empty interrupt request bit is set to "1".

Table 2.9.1 Bus control signals and data bus status

$\overline{Si}$	$\overline{R}$	$\overline{W}$	A0	Data bus status	Data on data bus
0	0	1	0	Read	Output data
0	0	1	1	Read	Status information
0	1	0	0	Write	Input data (Data)
0	1	0	1	Write	Input data (Command)
1	X	X	X	High impedance	—

2.9.5 Relevant registers setting

Figure 2.9.10 shows the relevant registers setting.

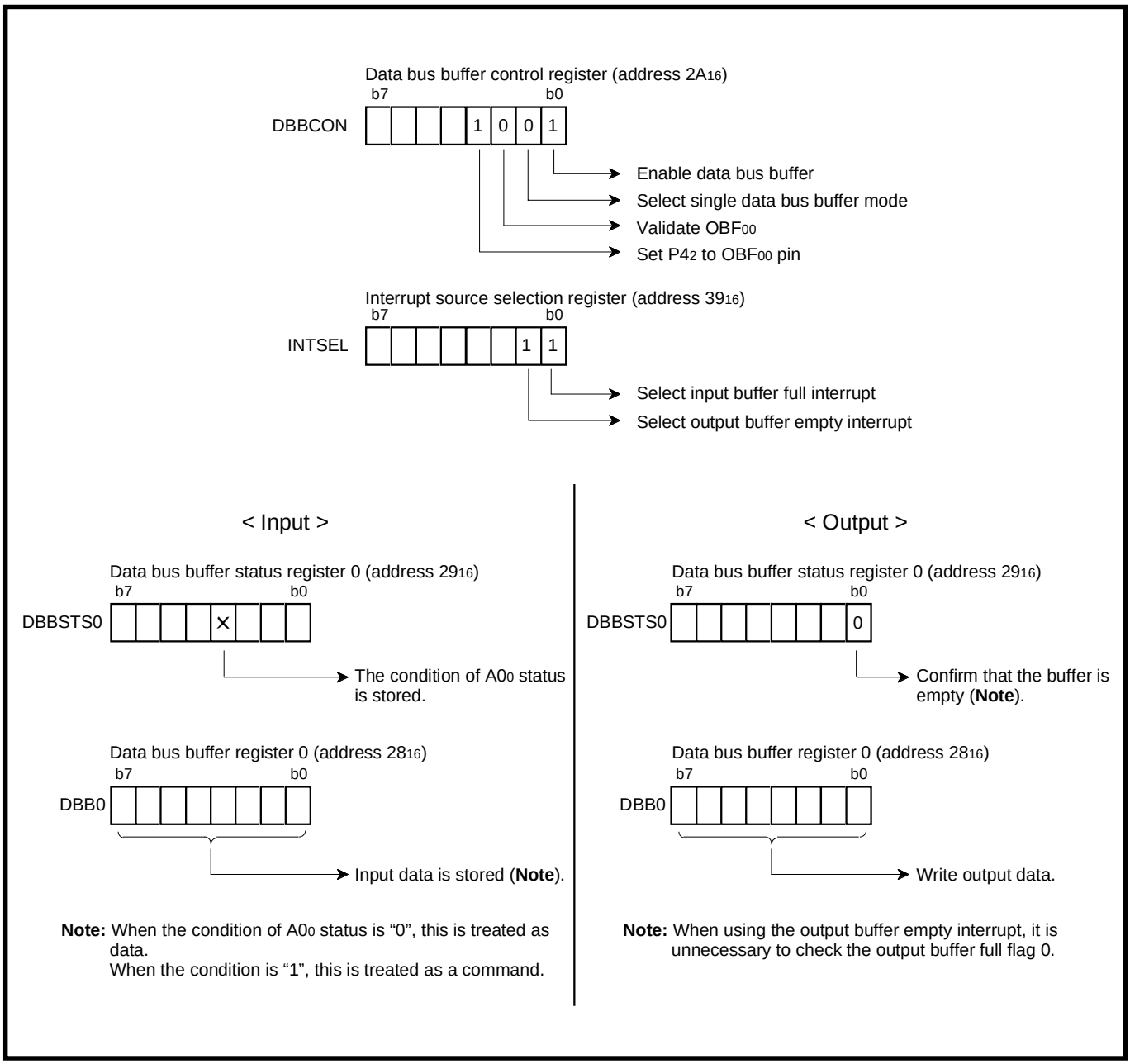


Fig. 2.9.10 Relevant registers setting

2.9 Bus interface

Figure 2.9.11 shows the control procedure using the interrupt.

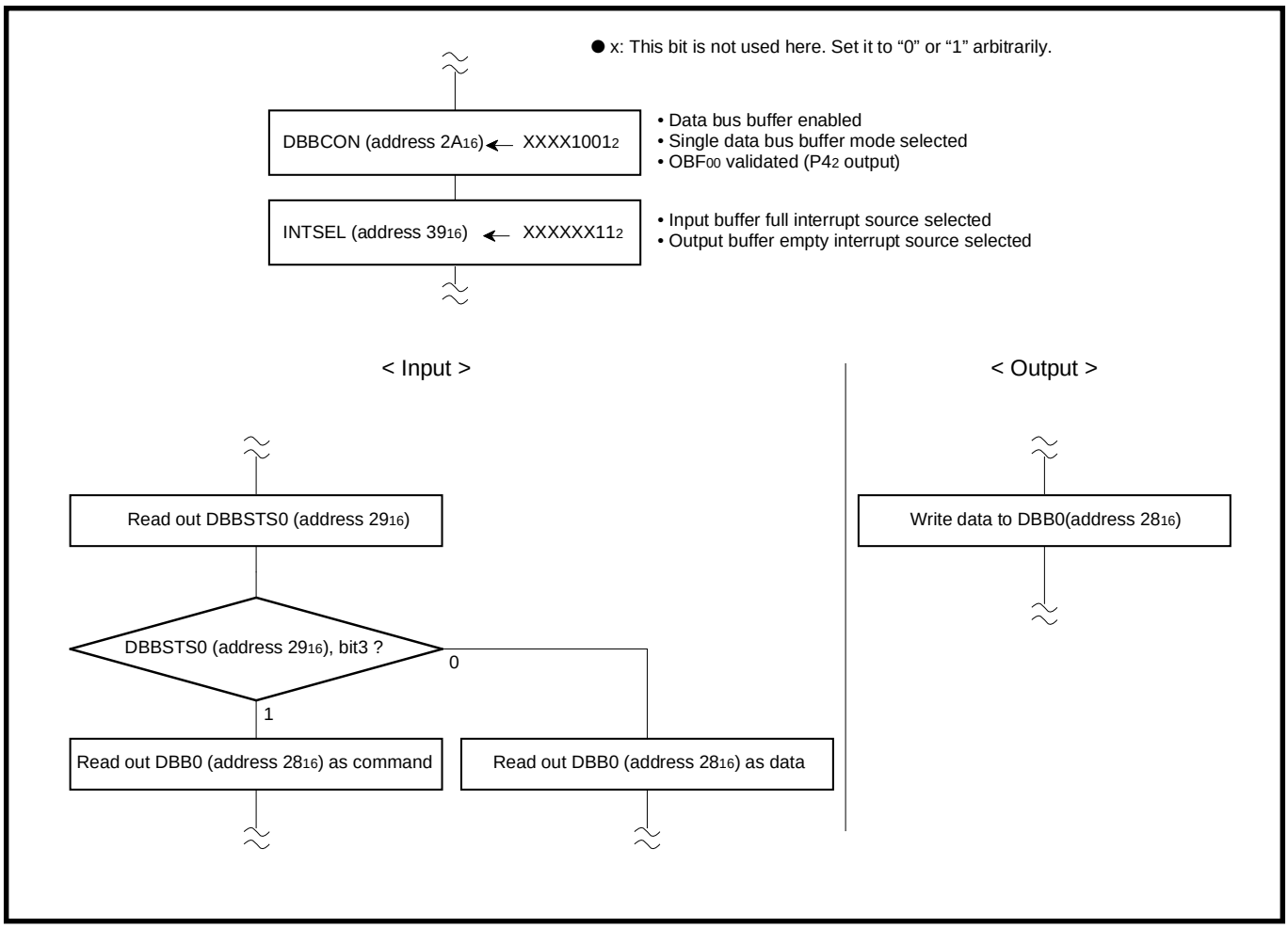


Fig. 2.9.11 Control procedure using interrupt

2.10 Watchdog timer

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

2.10.1 Memory map

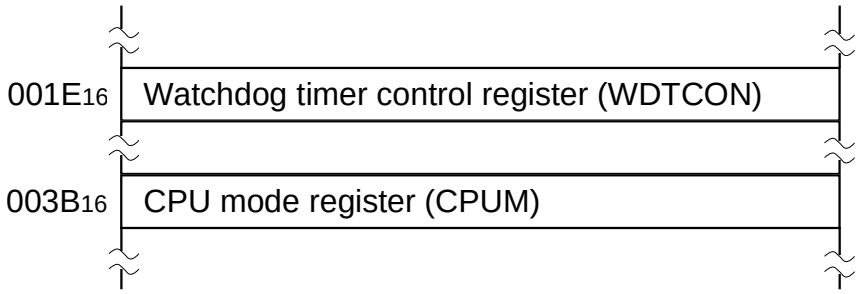


Fig. 2.10.1 Memory map of registers relevant to watchdog timer

2.10.2 Relevant registers

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0

Watchdog timer control register (WDTCON) [Address : $1E_{16}$]

B	Name	Function	At reset	R	W
0	Watchdog timer H (for read-out of high-order 6 bits)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0 : STP instruction enabled 1 : STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0 : Watchdog timer L underflow 1 : f(X_{IN})/16 or f(X_{CIN})/16	0	○	○

Fig. 2.10.2 Structure of Watchdog timer control register

CPU mode register

b7	b6	b5	b4	b3	b2	b1	b0
				1			

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

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : Memory expansion mode **(Note)** 10 : Microprocessor mode **(Note)** 11 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to “1”.		1	○	○
4	Port Xc switch bit	0: I/O port function (oscillation stopped) 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 bits	b7 b6 0 0: ϕ=f(X_{IN})/2 (high-speed mode) 0 1: ϕ=f(X_{IN})/8 (middle-speed mode) 1 0: ϕ=f(X_{CIN})/2 (low-speed mode) 1 1: Not available	1	○	○
7			0	○	○

*: The initial value of bit 1 depends on the CNVss level.
Note: This mode is not available for M38869M8A/MCA/MFA or the flash memory version.

Fig. 2.10.3 Structure of CPU mode register

2.10.3 Watchdog timer application examples

(1) Detection of program runaway

Outline: If program runaway occurs, let the microcomputer reset, using the internal timer for detection of program runaway.

Specifications:
- An underflow of watchdog timer H is judged to be program runaway, and the microcomputer is returned to the reset status.
- Before the watchdog timer H underflows, "0" is set into bit 7 of the watchdog timer control register at every cycle in a main routine.
- High-speed mode is used as a main clock division ratio.
- An underflow signal of the watchdog timer L is supplied as the count source of watchdog timer H.

Figure 2.10.4 shows a watchdog timer connection and division ratio setting; Figure 2.10.5 shows the relevant registers setting; Figure 2.10.6 shows the control procedure.

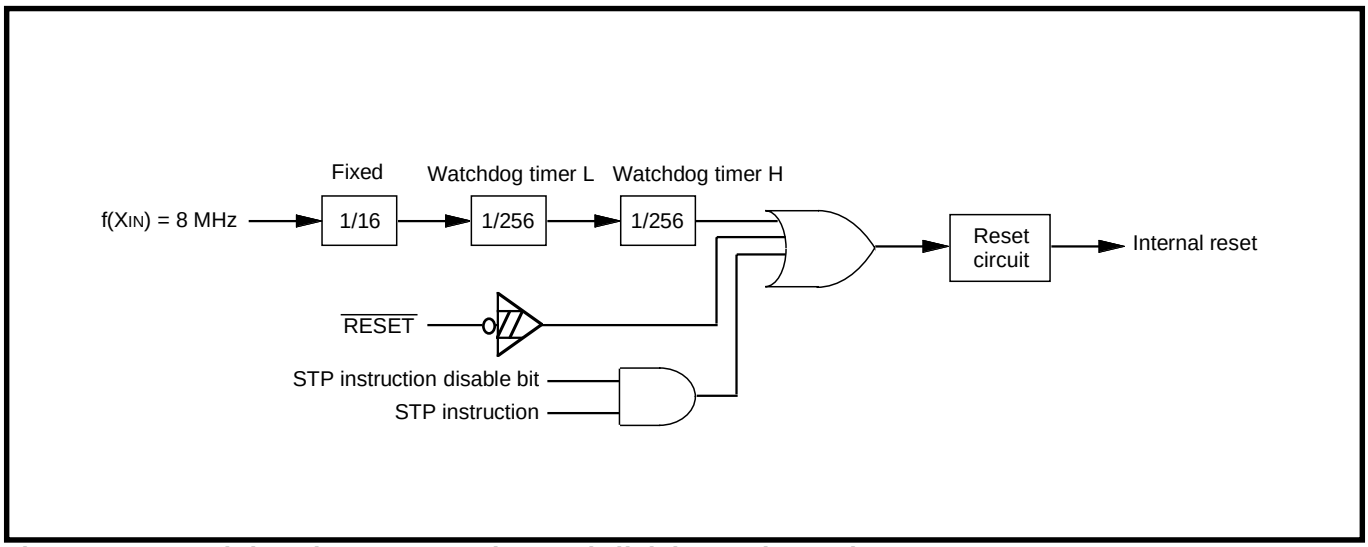


Fig. 2.10.4 Watchdog timer connection and division ratio setting

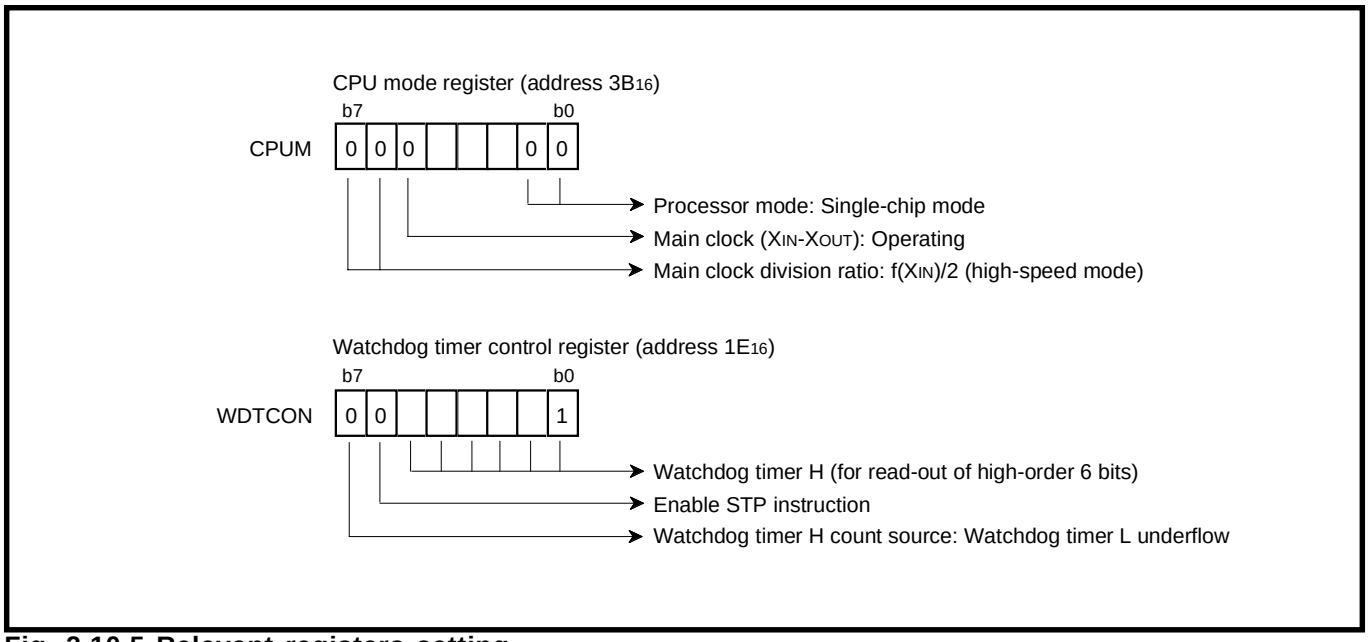


Fig. 2.10.5 Relevant registers setting

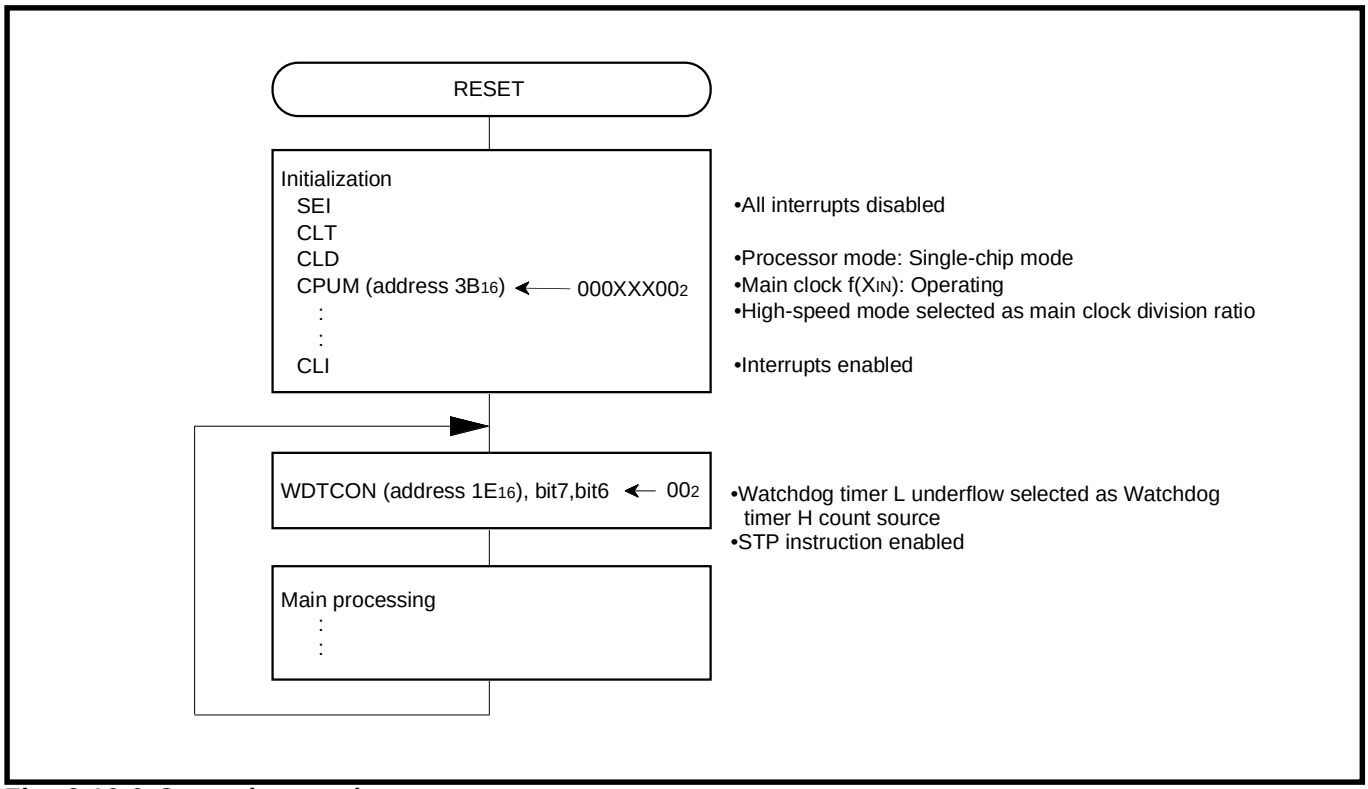


Fig. 2.10.6 Control procedure

2.10.4 Notes on watchdog timer

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

2.11 Reset

2.11.1 Connection example of reset IC

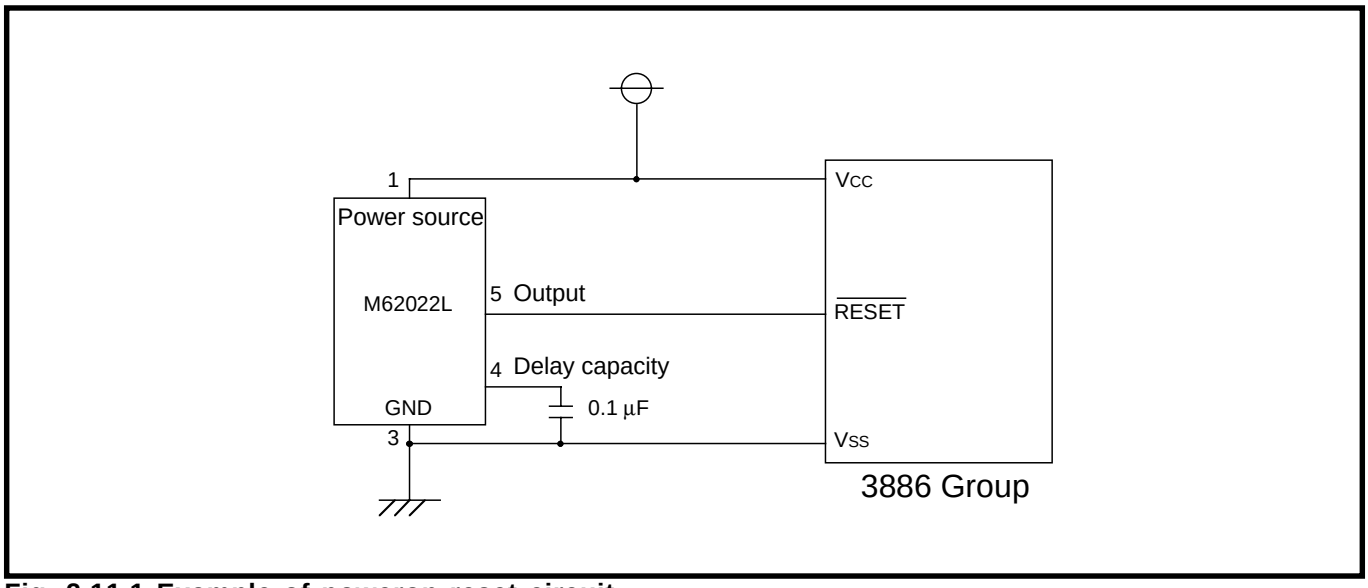


Fig. 2.11.1 Example of poweron reset circuit

Figure 2.11.2 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.

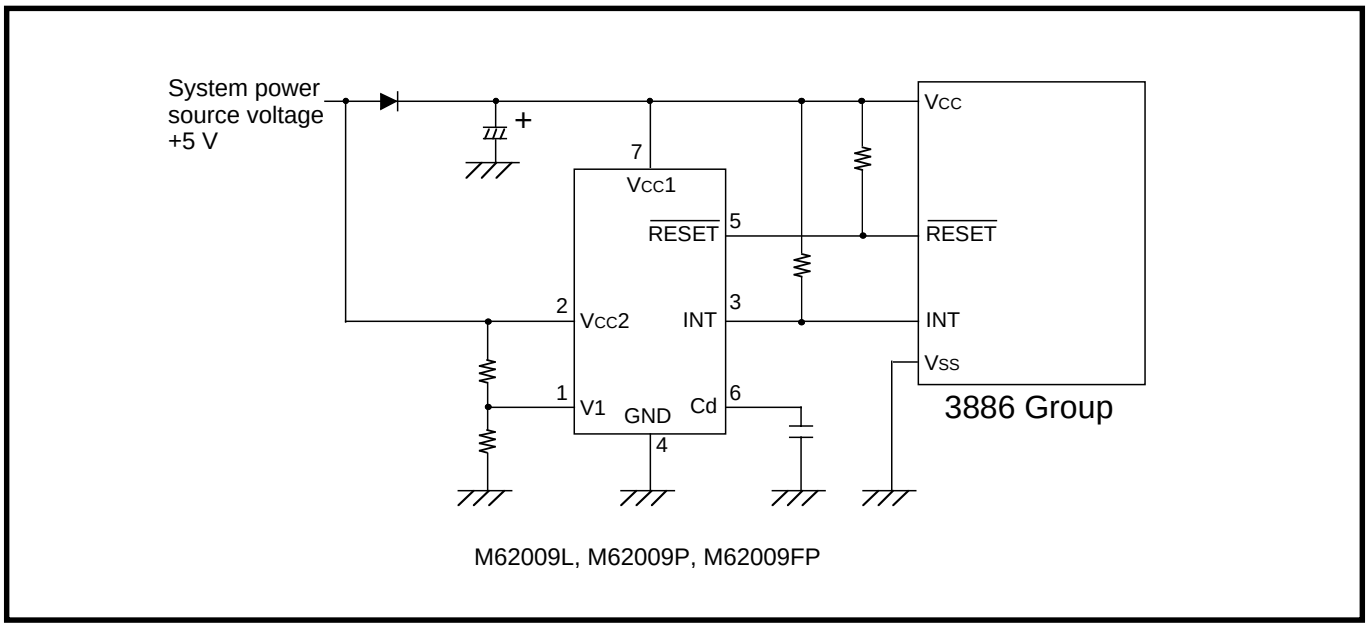


Fig. 2.11.2 RAM backup system

2.11 Reset

2.11.2 Notes on $\overline{\text{RESET}}$ pin

Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the Vss pin. Use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

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

● **Reason**

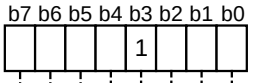
If the several nanosecond or several ten nanosecond impulse noise enters the $\overline{\text{RESET}}$ pin, it may cause a microcomputer failure.

2.12 Clock generating circuit

This paragraph explains how to set the registers relevant to the clock generating circuit and describes an application example.

2.12.1 Relevant registers

CPU mode register



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

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : Memory expansion mode **(Note)** 10 : Microprocessor mode **(Note)** 11 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to "1".		1	○	○
4	Port Xc switch bit	0: I/O port function (oscillation stopped) 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 bits	b7 b6 0 0: ϕ=f(X_{IN})/2 (high-speed mode) 0 1: ϕ=f(X_{IN})/8 (middle-speed mode) 1 0: ϕ=f(X_{CIN})/2 (low-speed mode) 1 1: Not available	1	○	○
7			0	○	○

*: The initial value of bit 1 depends on the CNVss level.
Note: This mode is not available for M38869M8A/MCA/MFA or the flash memory version.

Fig. 2.12.1 Structure of CPU mode register

APPLICATION

2.12 Clock generating circuit

2.12.2 Clock generating circuit application example

(1) Status transition during power failure

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

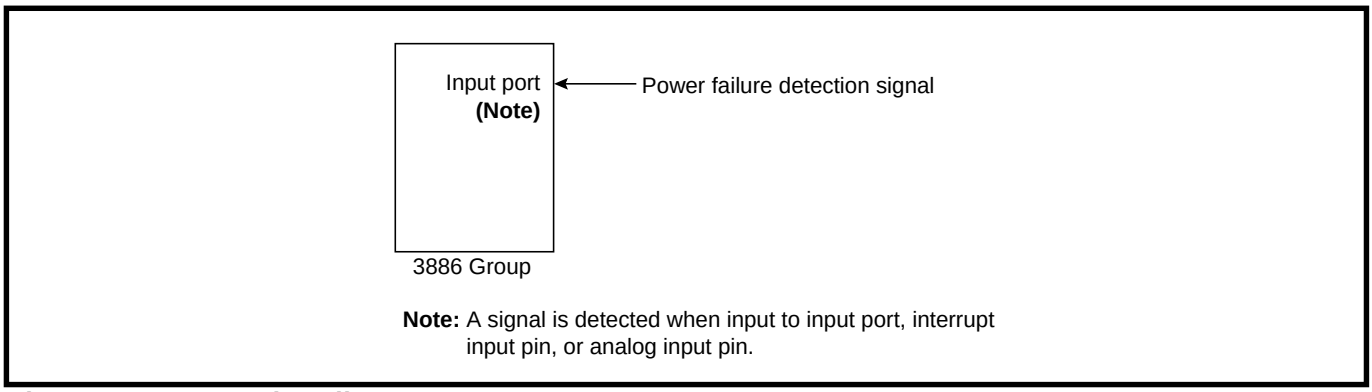


Fig. 2.12.2 Connection diagram

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 externally.
Output port: Fixed to output level that does not cause current flow to the external.
(Example) Fix to "H" for an LED circuit that turns on at "L" output level.
I/O port: Input port → Fixed to "H" or "L" level externally.
Output port → Output of data that does not consume current
V_{REF} pin: Terminate A-D conversion operation
Stop V_{REF} current dissipation by setting value of D-Ai conversion register to "00_{16}".

Figure 2.12.3 shows the status transition diagram during power failure and Figure 2.12.4 shows the setting of relevant registers.

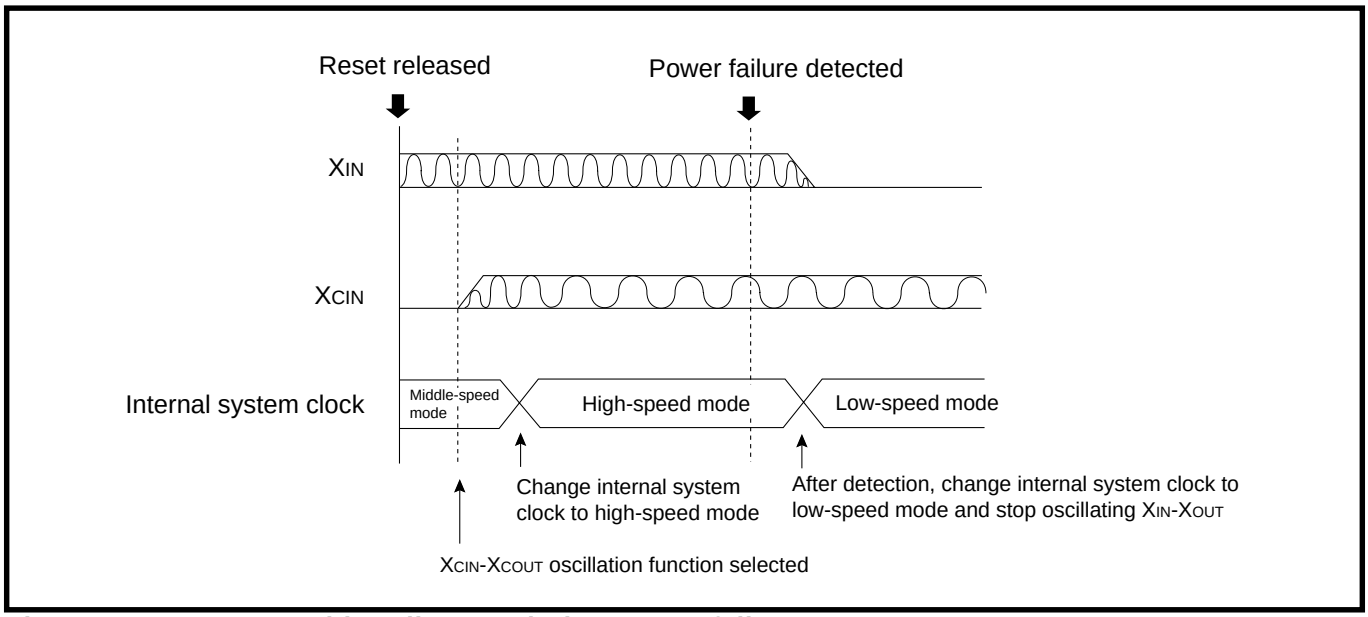
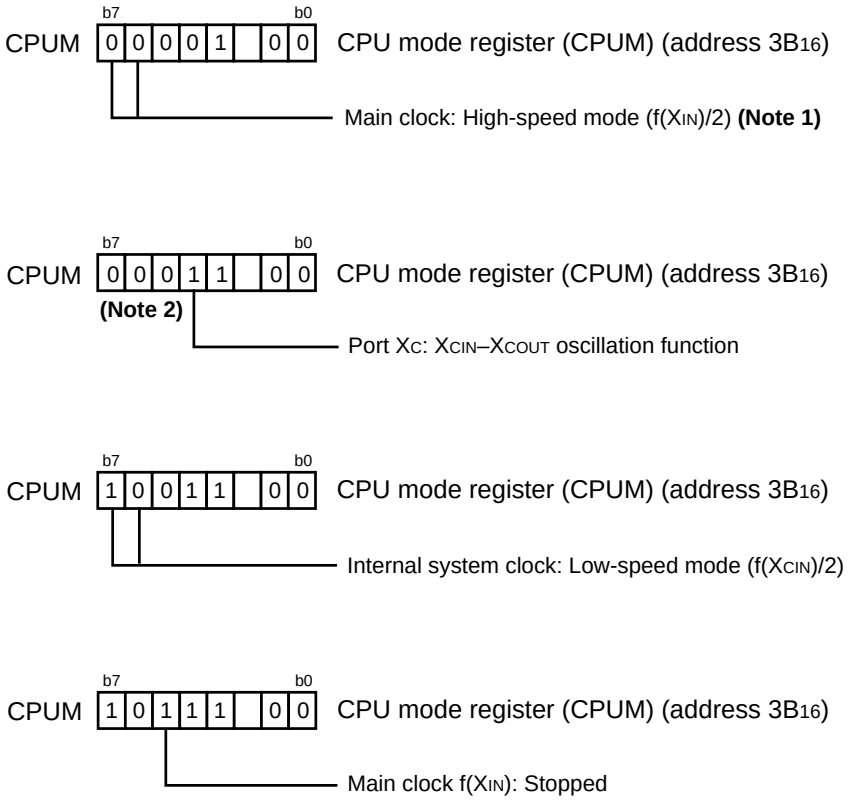


Fig. 2.12.3 Status transition diagram during power failure



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. 2.12.4 Setting of relevant registers

2.12 Clock generating circuit

Control procedure: To prepare for a power failure, set the relevant registers in the order shown below.

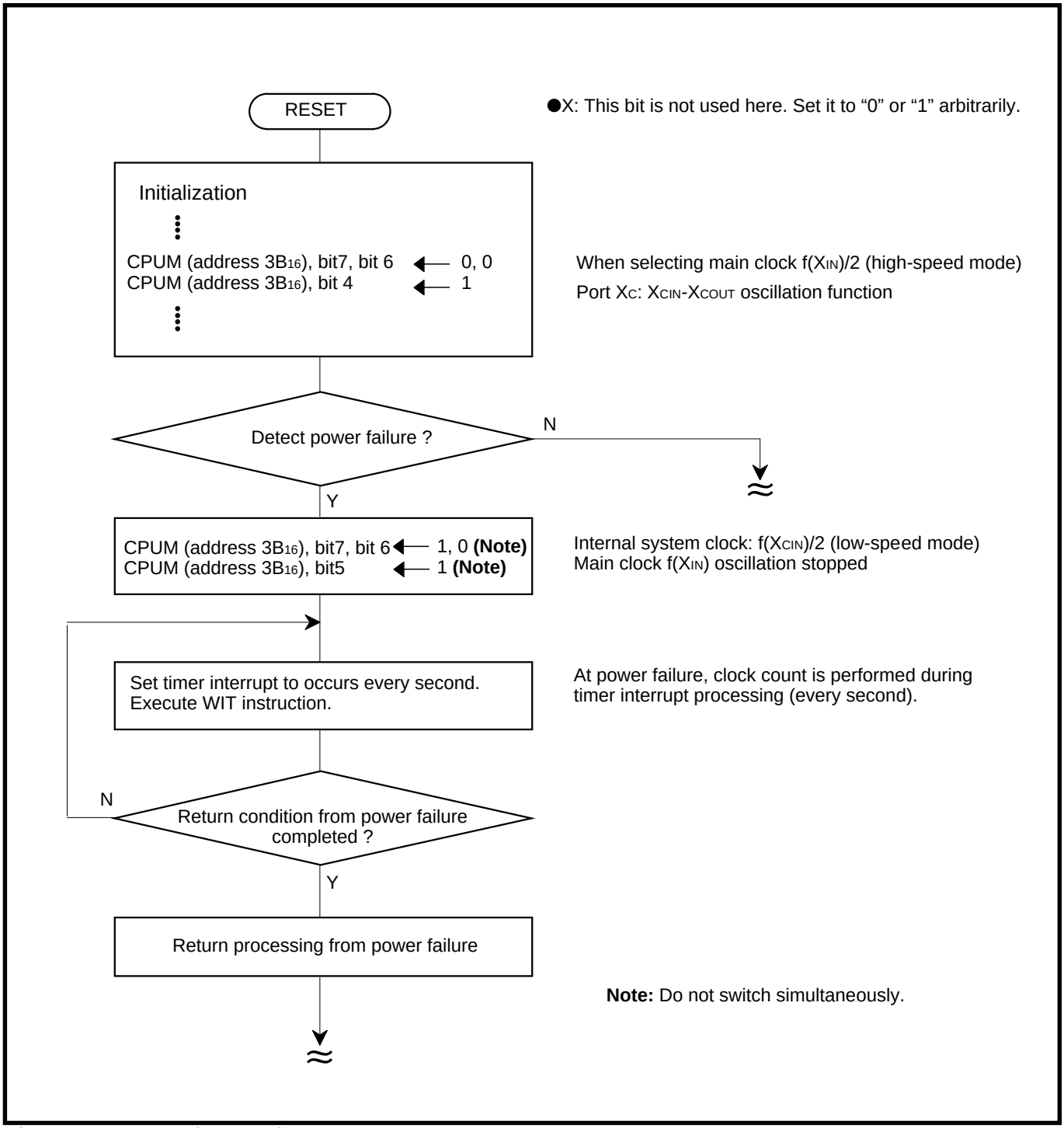


Fig. 2.12.5 Control procedure

2.13 Standby function

The 3886 group is provided with standby functions to stop the CPU by software and put the CPU into the low-power operation.
The following two types of standby functions are available.
•Stop mode using STP instruction
•Wait mode using WIT instruction

2.13.1 Stop mode

The stop mode is set by executing the STP instruction. In the stop mode, the oscillation of both clocks (X_{IN}–X_{OUT}, X_{CIN}–X_{COUT}) stop and the internal clock ϕ stops at the "H" level. The CPU stops and peripheral units stop operating. As a result, power dissipation is reduced.

(1) State in stop mode

Table 2.13.1 shows the state in the stop mode.

Table 2.13.1 State in stop mode

Item	State in stop mode
Oscillation	Stopped.
CPU	Stopped.
Internal clock ϕ	Stopped at "H" level.
I/O ports P0–P8	Retains the state at the STP instruction execution.
Timer	Stopped. (Timers 1, 2, X, Y) However, Timers X and Y can be operated in the event counter mode.
PWM0, PWM1	Stopped.
Watchdog timer	Stopped.
Serial I/O1, Serial I/O2	Stopped. However, these can be operated only when an external clock is selected.
I^2C-BUS interface	Stopped.
A-D converter	Stopped.
D-A converter	Retains output voltage.
Comparator	Stopped.
Bus interface	Operating.

2.13 Standby function

(2) Release of stop mode

The stop mode is released by a reset input or by the occurrence of an interrupt request. Note the differences in the restoration process according to reset input or interrupt request, as described below.

■Restoration by reset input

The stop mode is released by holding the $\overline{\text{RESET}}$ pin to the "L" input level during the stop mode. Oscillation is started when all ports are in the input state and the stop mode of the main clock (X_{IN}-X_{OUT}) is released.

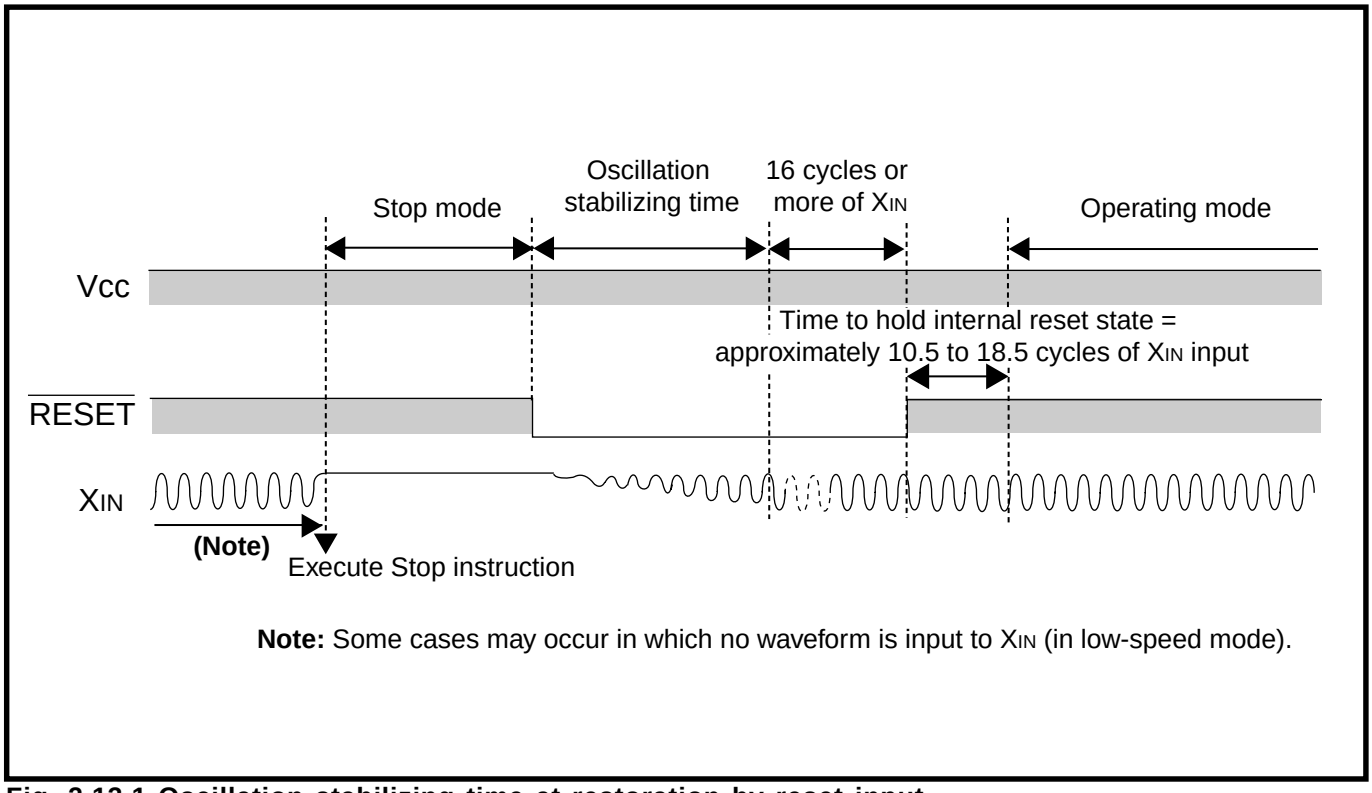
Oscillation is unstable when restarted. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) is required. The input of the $\overline{\text{RESET}}$ pin should be held at the "L" level until oscillation stabilizes.

When the $\overline{\text{RESET}}$ pin is held at the "L" level for 16 cycles or more of X_{IN} after the oscillation has stabilized, the microcomputer will go to the reset state. After the input level of the $\overline{\text{RESET}}$ pin is returned to "H", the reset state is released in approximately 10.5 to 18.5 cycles of the X_{IN} input.

Figure 2.13.1 shows the oscillation stabilizing time at restoration by reset input.

At release of the stop mode by reset input, the internal RAM retains its contents previous to the reset. However, the previous contents of the CPU register and SFR are not retained.

For more details concerning reset, refer to "2.11 Reset".



Note: Some cases may occur in which no waveform is input to X_{IN} (in low-speed mode).

Fig. 2.13.1 Oscillation stabilizing time at restoration by reset input

■Restoration by interrupt request

The occurrence of an interrupt request in the stop mode releases the stop mode. As a result, oscillation is resumed. The interrupts available for restoration are:

•INT_0–INT_4
•$CNTR_0$, $CNTR_1$
•Serial I/O (1, 2) using an external clock
•Timer X, Y using an external event count
•Key input (key-on wake-up)
•Bus interface
•S_{CL}/S_{DA}

However, when using any of these interrupt requests for restoration from the stop mode, in order to enable the selected interrupt, you must execute the STP instruction after setting the following conditions.

[Necessary register setting]

① Interrupt disable flag I = "0" (interrupt enabled)
② Timer 1 interrupt enable bit = "0" (interrupt disabled)
③ Interrupt request bit of interrupt source to be used for restoration = "0" (no interrupt request issued)
④ Interrupt enable bit of interrupt source to be used for restoration = "1" (interrupts enabled)

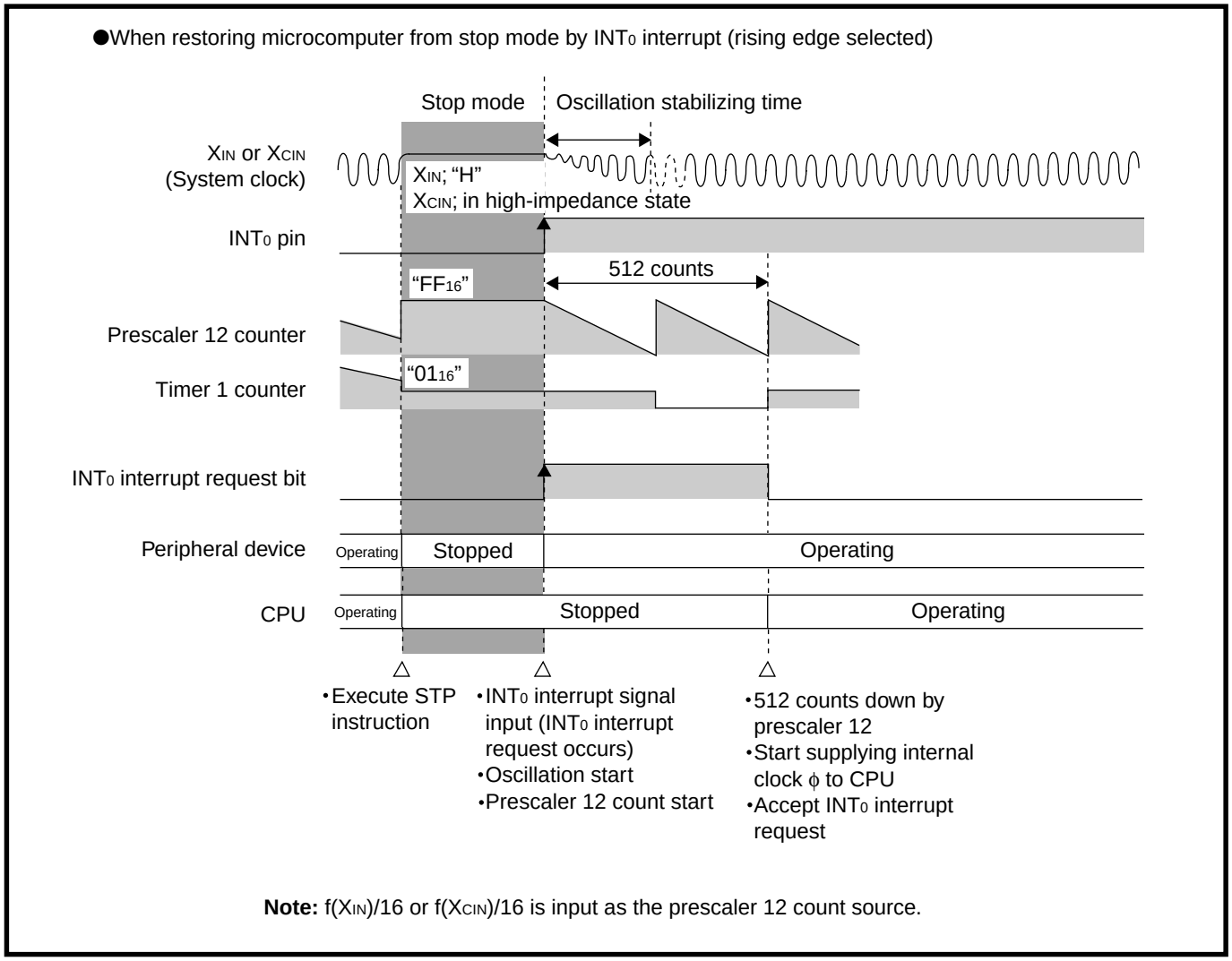
For more details concerning interrupts, refer to "2.2 Interrupts".

Oscillation is unstable when restarted. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) is required. For restoration by an interrupt request, waiting time prior to supplying internal clock ϕ to the CPU is automatically generated*2 by Prescaler 12 and Timer 1*1. This waiting time is reserved as the oscillation stabilizing time on the system clock side. The supply of internal clock ϕ to the CPU is started at the Timer 1 underflow.

Figure 2.13.2 shows an execution sequence example at restoration by the occurrence of an INT_0 interrupt request.

✻1: If the STP instruction is executed when the oscillation stabilizing time set after STP instruction released bit is "0", "FF_{16}" and "01_{16}" are automatically set in the Prescaler 12 counter/latch and Timer 1 counter/latch, respectively. When the oscillation stabilizing time set after STP instruction released bit is "1", nothing is automatically set to either Prescaler 12 or Timer 1. For this reason, any suitable value can be set to Prescaler 12 and Timer 1 for the oscillation stabilizing time.
✻2: Immediately after the oscillation is started, the count source is supplied to the prescaler 12 so that a count operation is started.

2.13 Standby function



Note: f(XIN)/16 or f(XCIN)/16 is input as the prescaler 12 count source.

Fig. 2.13.2 Execution sequence example at restoration by occurrence of INT_0 interrupt request

(3) Notes on using stop mode

■Restarting oscillation

Usually, when the MCU stops the clock oscillation by STP instruction and the STP instruction has been released by an external interrupt source, the fixed values of Timer 1 and Prescaler 12 (Timer 1 = 01_{16}, Prescaler 12 = FF_{16}) are automatically reloaded in order for the oscillation to stabilize. The user can inhibit the automatic setting by writing "1" to bit 6 of the port control register 2 (address $002F_{16}$).

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

•Reason

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

■**Clock restoration**

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

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

2.13.2 Wait mode

The wait mode is set by execution of the WIT instruction. In the wait mode, oscillation continues, but the internal clock ϕ stops at the "H" level.

The CPU stops, but most of the peripheral units continue operating.

(1) State in wait mode

The continuation of oscillation permits clock supply to the peripheral units except I^2C-BUS interface. Table 2.13.2 shows the state in the wait mode.

Table 2.13.2 State in wait mode

Item	State in wait mode
Oscillation	Operating.
CPU	Stopped.
Internal clock ϕ	Stopped at "H" level.
I/O ports P0–P8	Retains the state at the WIT instruction execution.
Timer	Operating.
PWM0, PWM1	Operating.
Watchdog timer	Operating.
Serial I/O1, Serial I/O2	Operating.
I^2C-BUS interface	Stopped. However, this operates when the system clock stop selection bit (bit 6 of address 15_{16}) is "1".
A-D converter	Operating.
D-A converter	Retains output voltage.
Comparator	Operating.
Bus interface	Operating.

2.13 Standby function

(2) Release of wait mode

The wait mode is released by reset input or by the occurrence of an interrupt request. Note the differences in the restoration process according to reset input or interrupt request, as described below.

In the wait mode, oscillation is continued, so an instruction can be executed immediately after the wait mode is released.

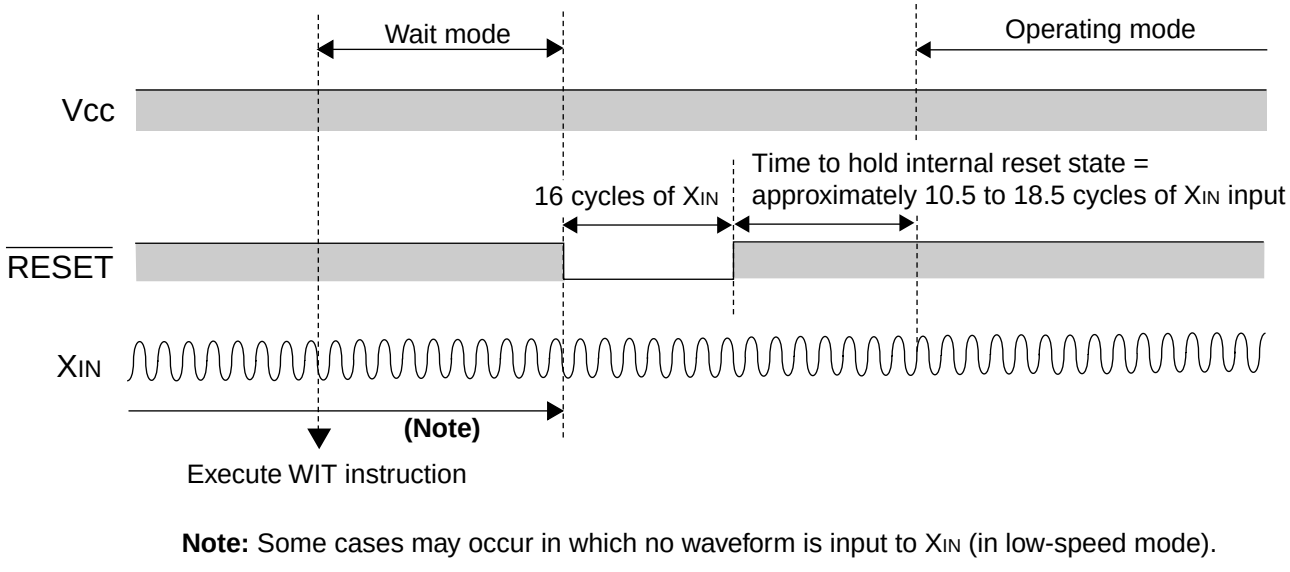
■Restoration by reset input

The wait mode is released by holding the input level of the $\overline{\text{RESET}}$ pin at "L" in the wait mode. Upon release of the wait mode, all ports are in the input state, and supply of the internal clock ϕ to the CPU is started. To reset the microcomputer, the $\overline{\text{RESET}}$ pin should be held at an "L" level for 16 cycles or more of X_{IN}. The reset state is released in approximately 10.5 cycles to 18.5 cycles of the X_{IN} input after the input of the $\overline{\text{RESET}}$ pin is returned to the "H" level.

At release of wait mode, the internal RAM retains its contents previous to the reset. However, the previous contents of the CPU register and SFR are not retained.

Figure 2.13.3 shows the reset input time.

For more details concerning reset, refer to "2.11 Reset".



Note: Some cases may occur in which no waveform is input to X_{IN} (in low-speed mode).

Fig. 2.13.3 Reset input time

■Restoration by interrupt request

In the wait mode, the occurrence of an interrupt request releases the wait mode and supply of the internal clock ϕ to the CPU is started. At the same time, the interrupt request used for restoration is accepted, so the interrupt processing routine is executed.

However, when using an interrupt request for restoration from the wait mode, in order to enable the selected interrupt, you must execute the STP instruction after setting the following conditions.

[Necessary register setting]

① Interrupt disable flag I = "0" (interrupt enabled)

② Interrupt request bit of interrupt source to be used for restoration = "0" (no interrupt request issued)

③ Interrupt enable bit of interrupt source to be used for restoration = "1" (interrupts enabled)

For more details concerning interrupts, refer to "2.2 Interrupts".

(3) Notes on wait mode

■Clock restoration

If the wait mode is released by a reset when X_{CIN} is set as the system clock and X_{IN} oscillation is stopped during execution of the WIT instruction, X_{CIN} oscillation stops, X_{IN} oscillations starts, and X_{IN} is set as the system clock.

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

2.14 Processor mode

This paragraph explains usage examples and others relevant to the processor mode. (Support product: M38867M8A/E8A)

2.14.1 Memory map

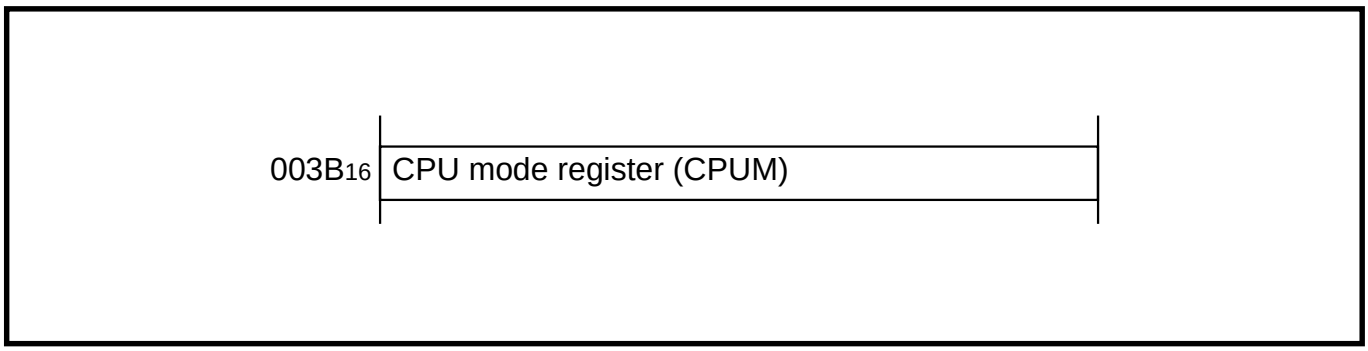


Fig. 2.14.1 Memory map of registers relevant to processor mode

2.14.2 Relevant registers

CPU mode register

b7	b6	b5	b4	b3	b2	b1	b0
				1			

CPU mode register
(CPUM: address 3B16)

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : Memory expansion mode **(Note)** 10 : Microprocessor mode **(Note)** 11 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to “1”.		1	○	○
4	Port Xc switch bit	0: I/O port function (oscillation stopped) 1: XCIN-XCOUT oscillation function	0	○	○
5	Main clock (XIN-XOUT) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bits	b7 b6 0 0: ϕ=f(XIN)/2 (high-speed mode) 0 1: ϕ=f(XIN)/8 (middle-speed mode) 1 0: ϕ=f(XCIN)/2 (low-speed mode) 1 1: Not available	1	○	○
7			0	○	○

*: The initial value of bit 1 depends on the CNVss level.

Note: This mode is not available for M38869M8A/MCA/MFA or the flash memory version.

Fig. 2.14.2 Structure of CPU mode register

2.14.3 Processor mode usage examples

(1) External memory connection example unusing $\overline{\text{ONW}}$ (one wait) function

Outline: An external memory is accessed, using the microprocessor mode. The RAM which meets the following conditions can be used at $f(X_{IN})$ = 8 MHz:

- •$\overline{\text{OE}}$ access time: $t_{a(OE)} \leq 50$ ns
- •Data set up time at write: $t_{su(D)} \leq 65$ ns.

For example, the M5M5256BP-10, whose address access time is 100 ns, can be used. Figure 2.14.3 shows the expansion example of 32-Kbytes ROM and RAM.

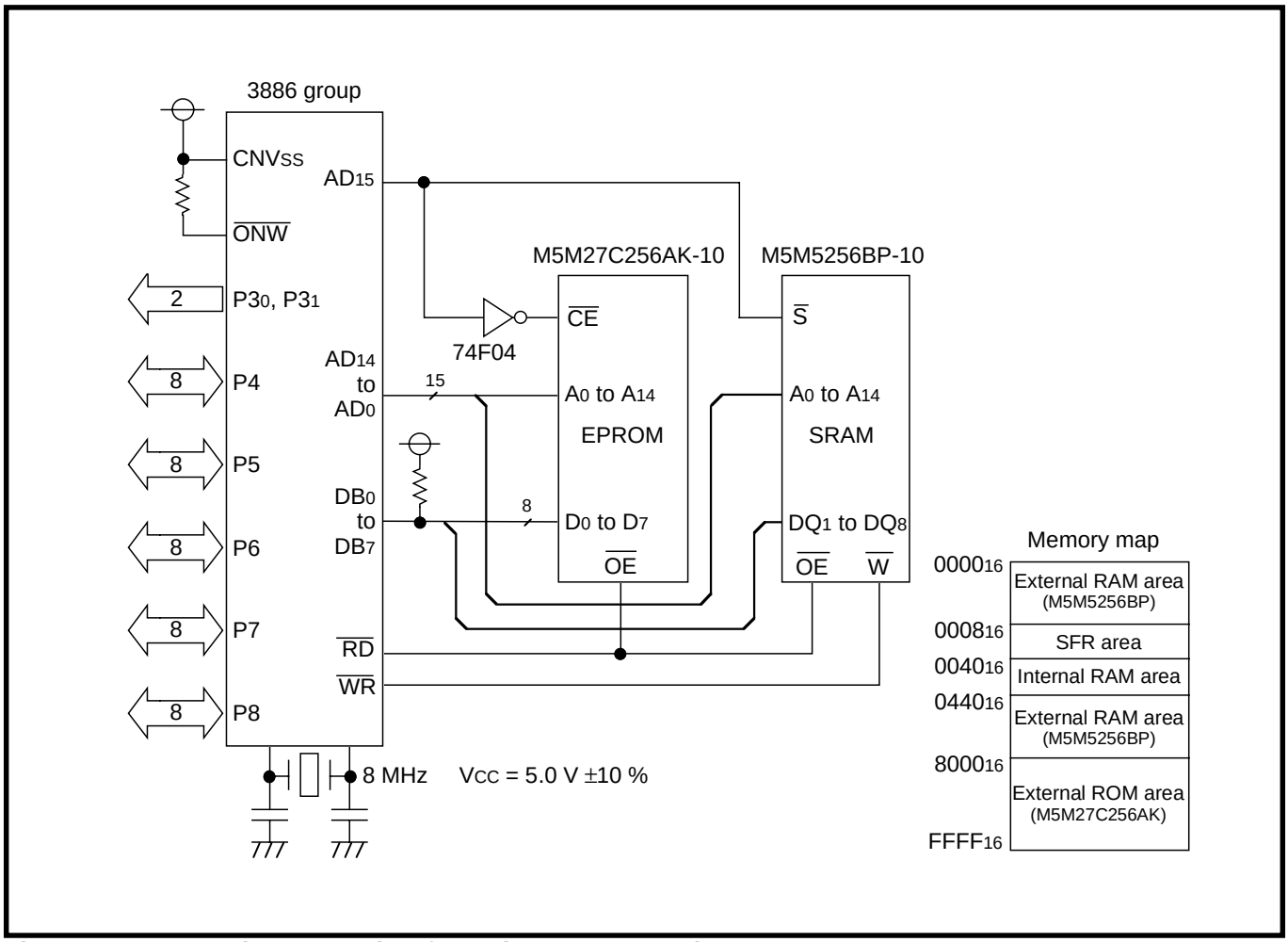


Fig. 2.14.3 Expansion example of 32-Kbytes ROM and RAM

Figures 2.14.4 to 2.14.6 show the basic timing at 8 MHz (no wait) operating.

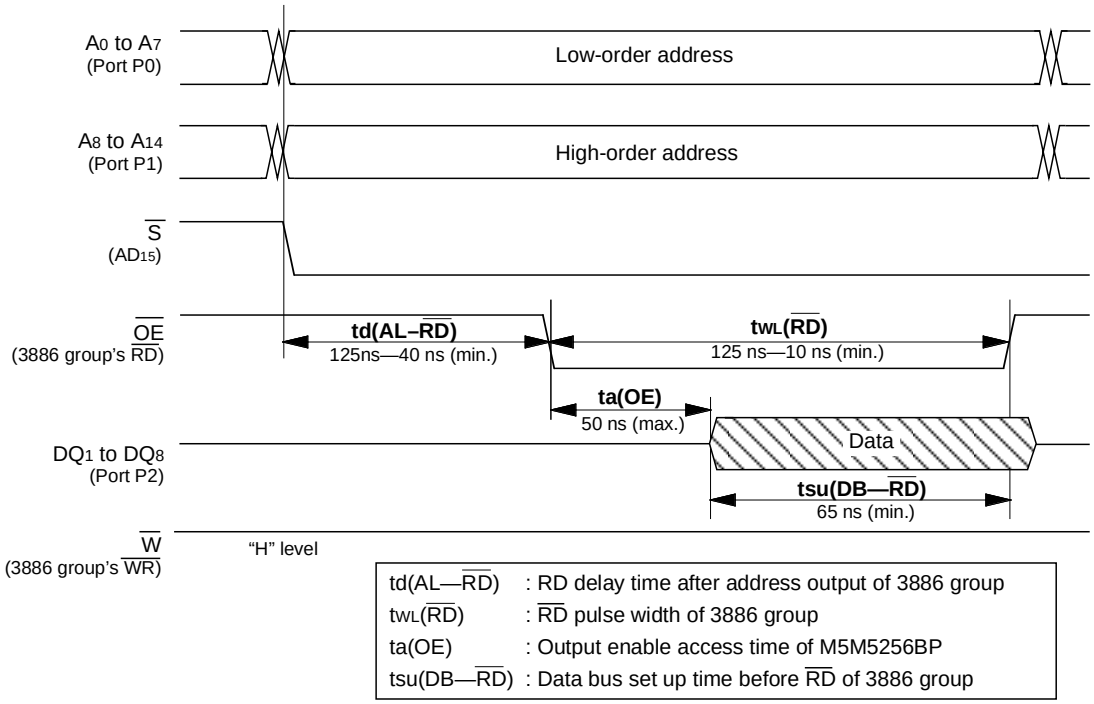


Fig. 2.14.4 Read cycle (OE access, SRAM)

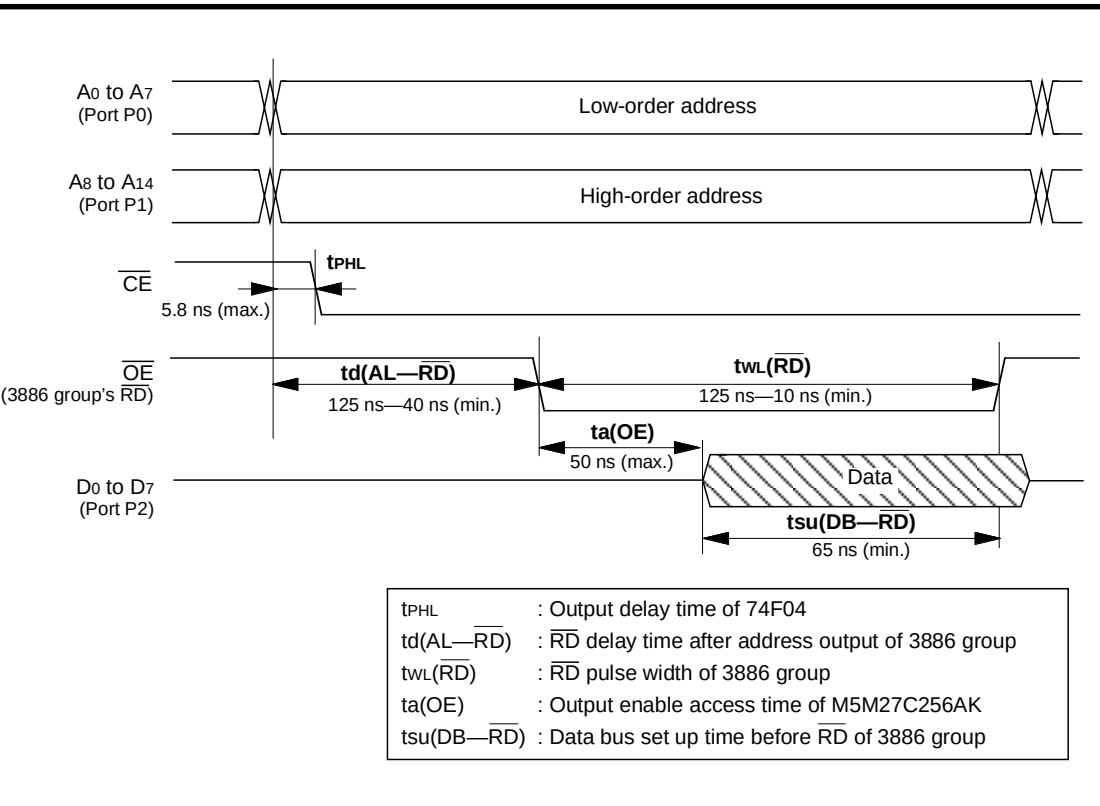


Fig. 2.14.5 Read cycle (OE access, EPROM)

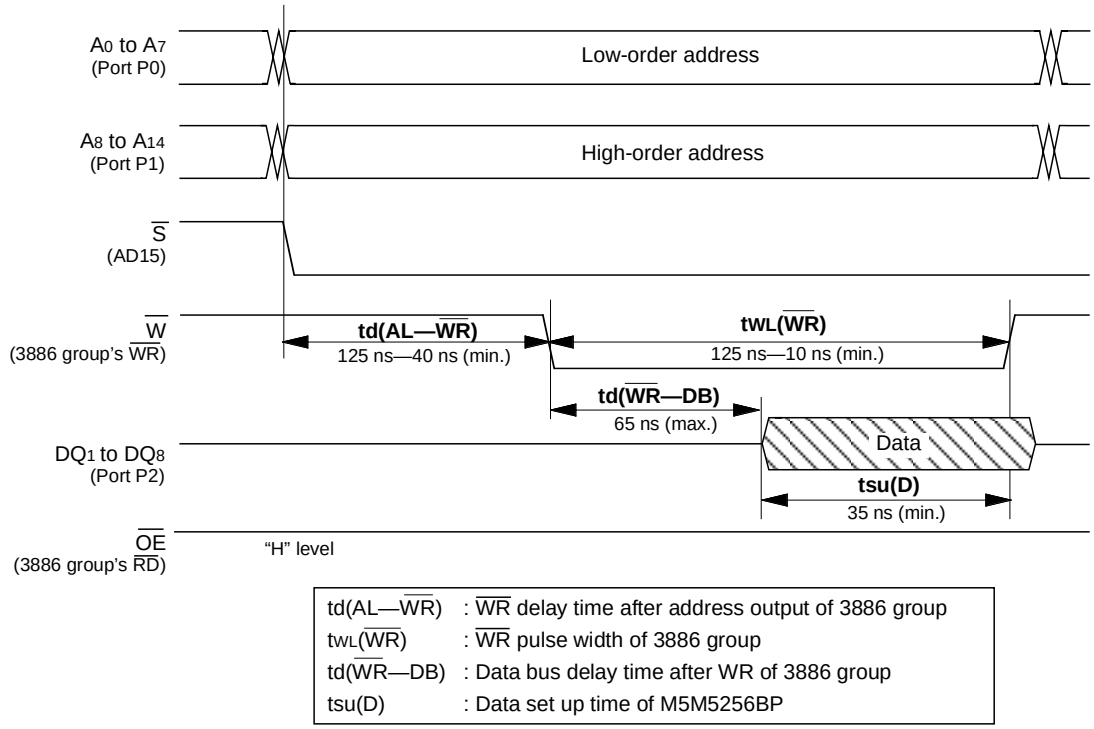


Fig. 2.14.6 Write cycle (W control, SRAM)

(2) External memory connection example using $\overline{\text{ONW}}$ (one wait) function

Outline: When the access time of an external memory is slow, the $\overline{\text{ONW}}$ function is used.

When "L" level is input to the P3$_2$/$\overline{\text{ONW}}$ pin in the state that the CPU reads or writes, a read or write cycle is extended by one cycle of ϕ. The $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal retains "L" level during the extended time.

The $\overline{\text{ONW}}$ function is valid for read or write to addresses 0000_{16} to 0007_{16} and 0440_{16} to $FFFF_{16}$.

Figure 2.14.7 shows the usage example using the $\overline{\text{ONW}}$ function.

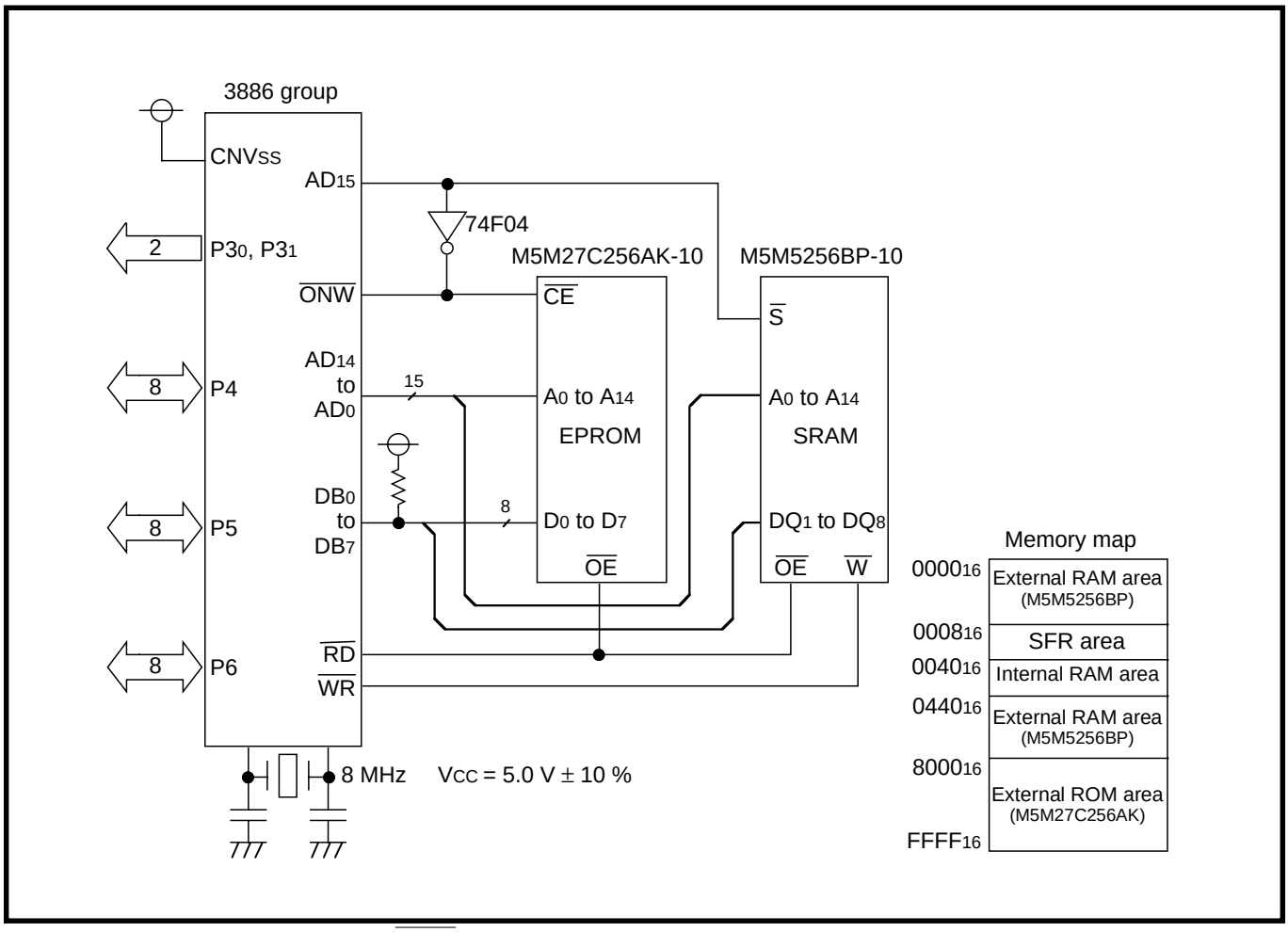


Fig. 2.14.7 Usage example of $\overline{\text{ONW}}$ function

(3) External memory connection example at f(X_{IN}) = 8 MHz or more

Outline: When the access time of an external memory is fast, it is possible to use at f(X_{IN}) = 8 MHz or more. The RAM which meets the following conditions can be used at f(X_{IN}) = 9 MHz:

- •$\overline{OE}$ access time: $t_{a(OE)} \leq 35$ ns
- •Data set up time at write: $t_{su(D)} \leq 50$ ns.

For example, the M5M5256BP-70, whose address access time is 70 ns, can be used. Figure 2.14.8 shows the expansion example of 32-Kbytes ROM and RAM.

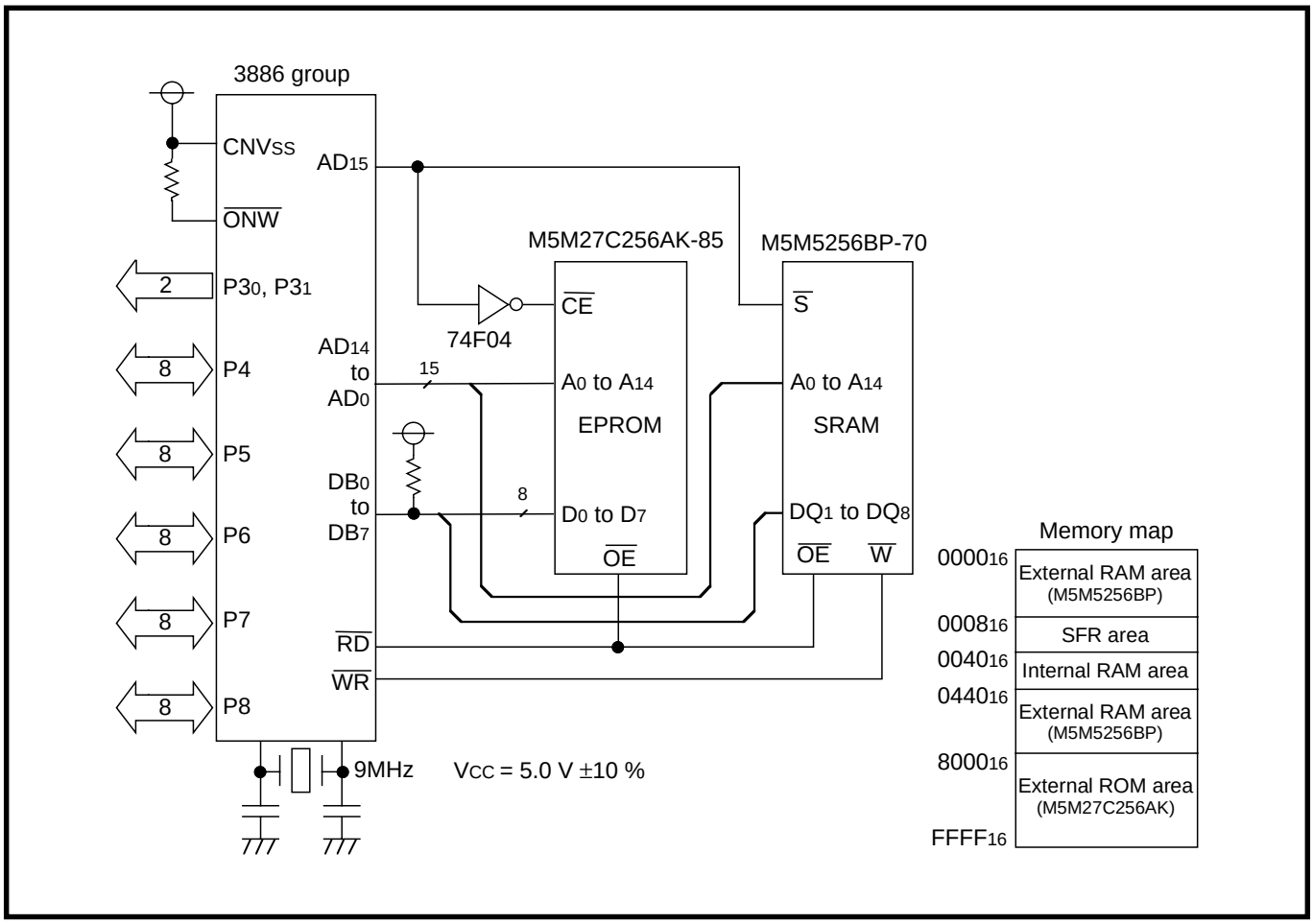


Fig. 2.14.8 Expansion example of 32-Kbytes ROM and RAM at f(X_{IN}) = 8 MHz or more

APPLICATION

2.14 Processor mode

Figures 2.14.9 to 2.14.11 show the basic timing at 9 MHz (no wait) operating.

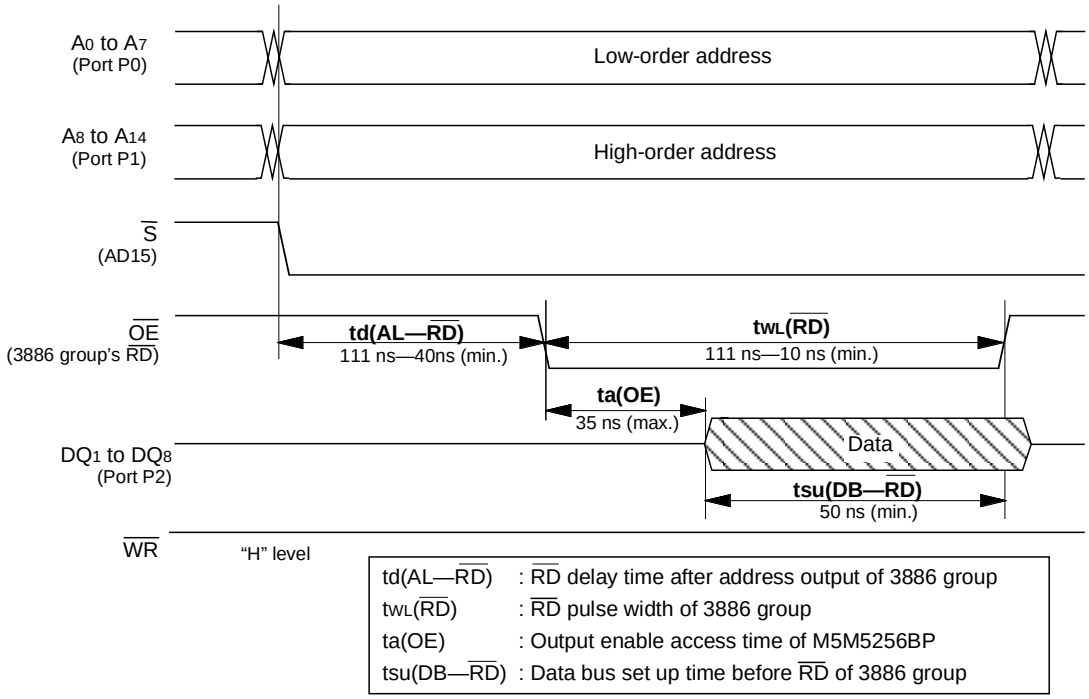


Fig. 2.14.9 Read cycle (OE access, SRAM)

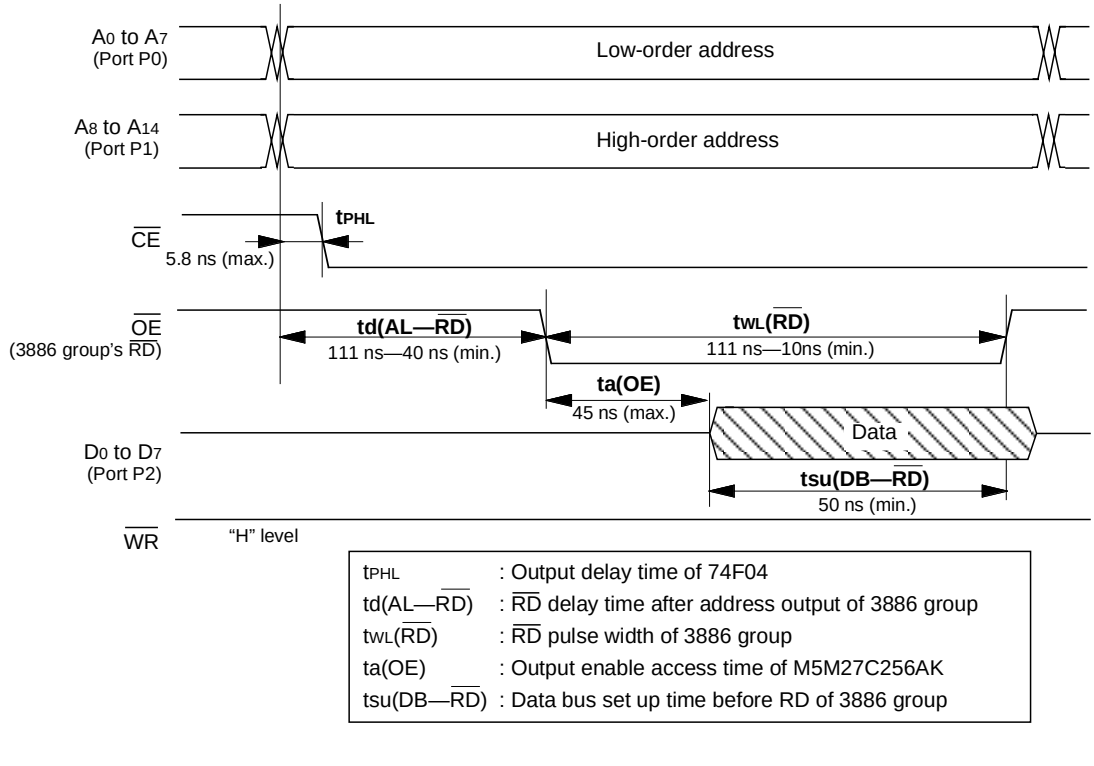


Fig. 2.14.10 Read cycle (OE access, EPROM)

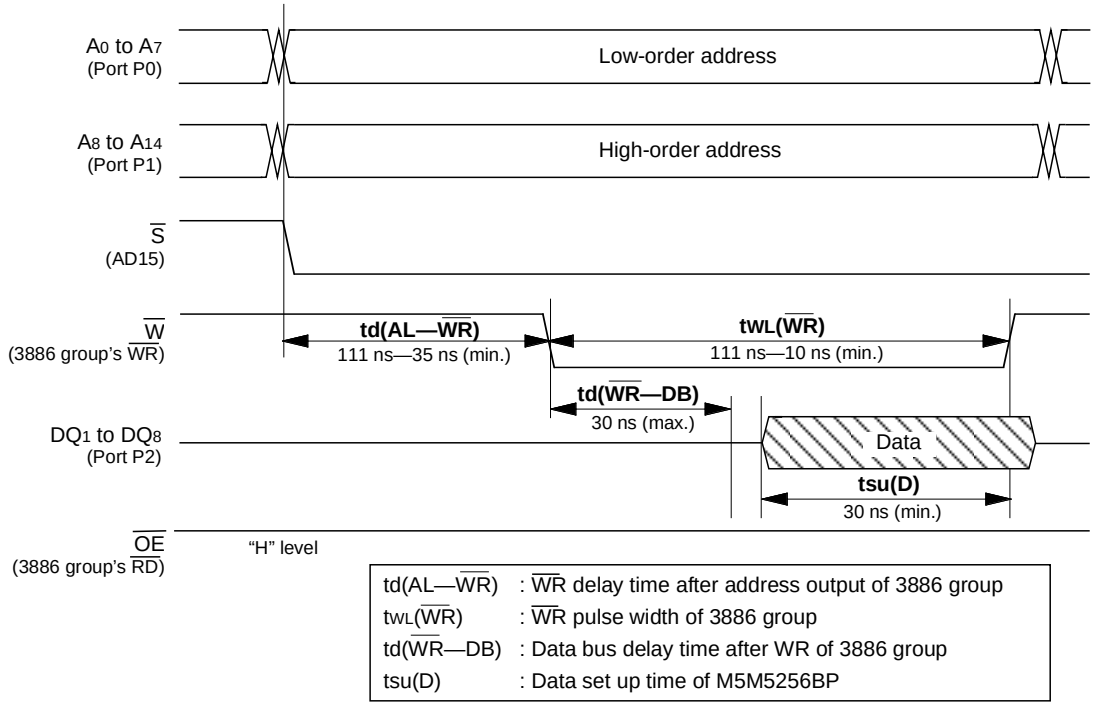


Fig. 2.14.11 Write cycle (W control, SRAM)

2.15 Flash memory

This paragraph explains the registers setting method and the notes relevant to the flash memory version.

2.15.1 Overview

The functions of the flash memory version are similar to those of the mask ROM version except that the flash memory is built-in and some of the SFR area differ from that of the mask ROM version (refer to "2.15.2 Memory map").

In the flash memory version, the built-in flash memory can be programmed or erased by using the following three modes.

- CPU reprogramming mode
- Parallel input/output mode
- Serial input/output mode

2.15.2 Memory map

M38869FFAHP/GP have 60 Kbytes of built-in flash memory.

Figure 2.15.1 shows the memory map of the flash memory version.

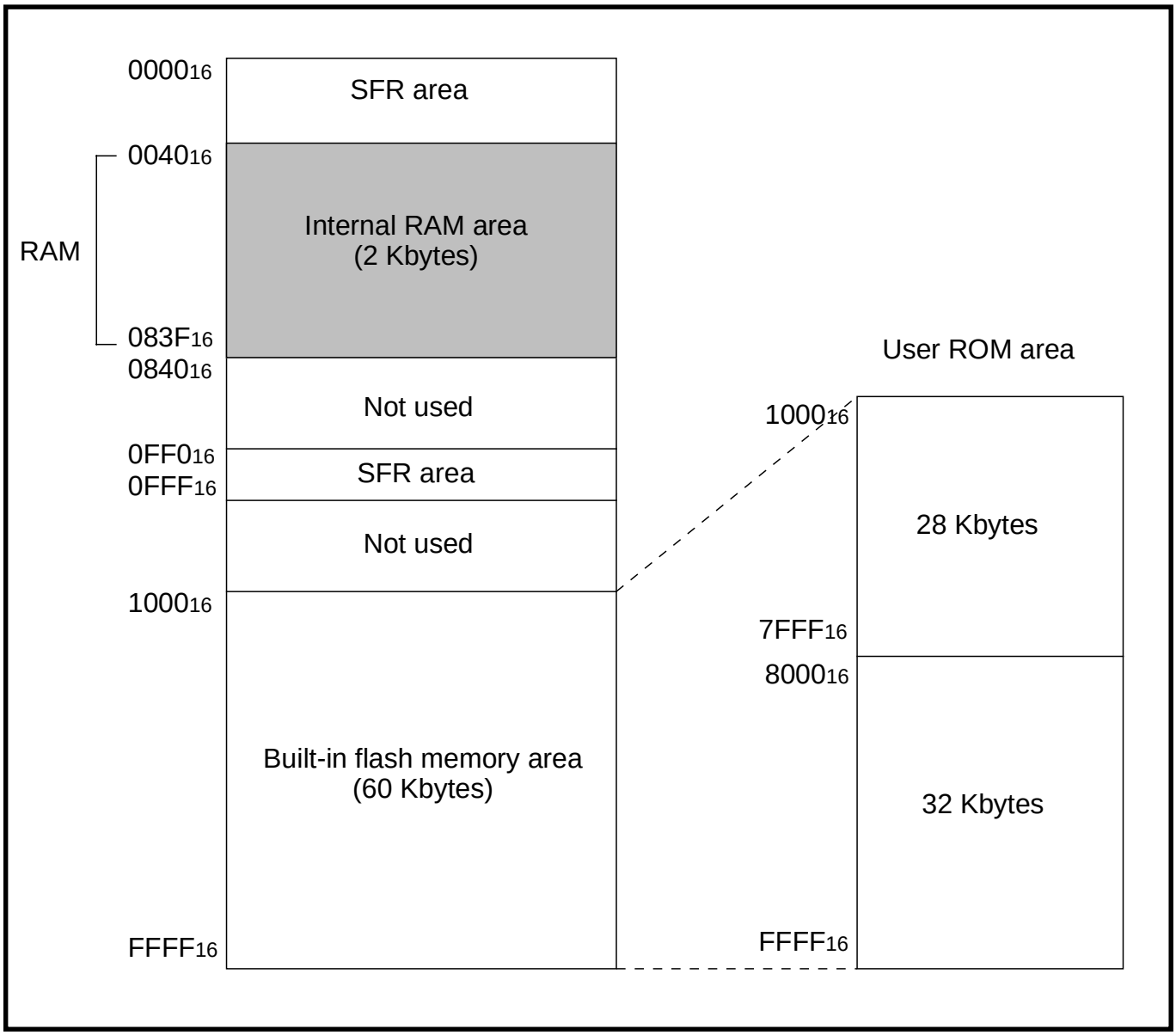


Fig. 2.15.1 Memory map of flash memory version for 3886 Group

2.15.3 Relevant registers

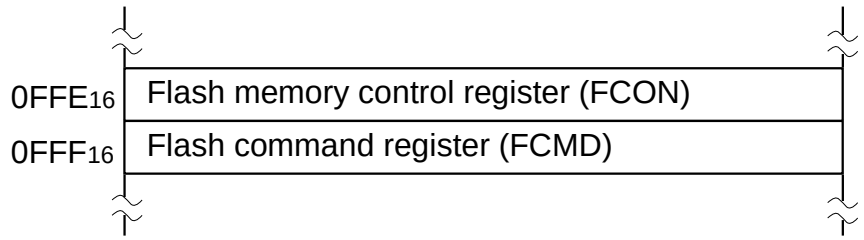


Fig. 2.15.2 Memory map of registers relevant to flash memory

Flash memory control register

b7 b6 b5 b4 b3 b2 b1 b0
0 0

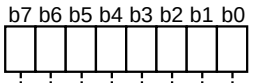
Flash memory control register
(FCON : address FFE16)

b	Name	Functions	At reset	R	W
0	CPU reprogramming mode select bit **(Note)**	0 : CPU reprogramming mode is invalid. (Normal operation mode) 1 : When applying 0 V to CNVSS/VPP pin, CPU reprogramming mode is invalid. When applying VPPH to CNVSS/VPP pin, CPU reprogramming mode is valid.	0	○	○
1	Erase/Program busy flag	0 : Erase and program are completed or have not been executed. 1 : Erase/program is being executed.	0	○	○
2	CPU reprogramming mode monitor flag	0 : CPU reprogramming mode is invalid 1 : CPU reprogramming mode is valid	0	○	○
3	Fix this bit to “0”.		0	○	○
4	Erase/Program area select bits	b5 b4 0 0: Address 100016 to FFFF16 (total 60Kbytes) 0 1: Address 100016 to 7FFF16 (total 28Kbytes) 1 0: Address 800016 to FFFF16 (total 32Kbytes) 1 1: not available	0	○	○
5			0	○	○
6	Fix this bit to “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: Bit 0 can be reprogrammed only when 0 V is applied to the CNVSS/VPP pin.

Fig. 2.15.3 Structure of Flash memory control register

Flash command register



Flash command register
(FCMD: address FFF_{16})

b	Functions		At reset	R	W
0	Writing of software command		0	×	○
1	<Software command name>	<Command code>	0	×	○
2	• Read command	• “00_{16}”	0	×	○
3	• Program command	• “40_{16}”	0	×	○
4	• Program verify command	• “$C0_{16}$”	0	×	○
5	• Erase command	• “20_{16}”+ “20_{16}”	0	×	○
6	• Erase verify command	• “$A0_{16}$”	0	×	○
7	• Reset command	• “FF_{16}” + “FF_{16}”	0	×	○

Note: The flash command register is a write-only register.

Fig. 2.15.4 Structure of Flash command register

2.15.4 Parallel I/O mode

In the parallel I/O mode, program/erase to the built-in flash memory can be performed by a general EPROM programmer.

Set the programming mode of the EPROM programmer to M5M28F101 and set the memory area of program/erase from 01000_{16} to $0FFFF_{16}$. Be especially careful when erasing; if the memory area is not set correctly, the products will be damaged eternally.

Table 2.15.1 shows the setting of EPROM programmers when programming in the parallel I/O mode.

Table 2.15.1 Setting of EPROM programmers when parallel programming

Products	Programming adapter	Programming mode	Memory area
M38869FFAHP	PCA4738HF-80	M5M28F101	01000_{16} to $0FFFF_{16}$
M38869FFAGP	PCA4738GF-80		

2.15.5 Serial I/O mode

Table 2.15.2 shows a pin connection example using MSP-I/MSP-II* between the programmer and the microcomputer when programming in the serial I/O mode.

*MSP-I/MSP-II provided by Suisei Electronics System Co., Ltd. (http://www.suisei.co.jp/index_e.htm) (product available in Asia and Oceania only)

Table 2.15.2 Connection example to programmer when serial programming

MSP-I/MSP-II		3886 Group flash memory version	
Signal name	Target connector Line number	Pin name	Pin number
BUSY	1	$P4_7/\overline{S_{RDY1}}$	18
VPP (**Note 1**)	2	CNV_{SS} (**Note 1**)	24
VDD (**Note 3**)	3	V_{CC} (**Note 3**)	71
SCL	4	$P4_6/S_{CLK1}$	19
SDA	5	$P4_4$/RxD	21
PGM/OE	6	$P3_7$	55
RESET	7	$\overline{RESET}$	25
GND (**Note 2**)	8	V_{SS}, AV_{SS} (**Note 2**)	30, 73

Notes 1: Connect an approximate 0.01 μF capacitor between CNV_{SS}/V_{PP} and GND for noise elimination.

2: When connecting a serial programmer, first connect both GNDs to the same GND level.

3: When the V_{CC} power is already supplied to the target board, do not connect the VDD supply pin of the serial programmer to V_{CC} of the target board.

2.15 Flash memory

2.15.6 CPU reprogramming mode

In the CPU reprogramming mode, issuing software commands through the Central Processing Unit (CPU) can reprogram the built-in flash memory. Accordingly, the contents of the built-in flash memory can be reprogrammed with the microcomputer itself mounted on board, without using the EPROM programmer. Store the reprogramming control program to the built-in flash memory in advance. The built-in flash memory cannot be read in the CPU reprogramming mode. Accordingly, after transferring the reprogramming control program to the internal RAM, execute it on the RAM.
The following commands can be used in the CPU reprogramming mode: read, program, program verify, erase, erase verify, and reset. For details concerning each command, refer to “CHAPTER 1 Flash memory mode 3 (CPU reprogramming mode)”.

(1) CPU reprogramming mode beginning/release procedures

Operation procedure in the CPU reprogramming mode for the built-in flash memory is described below.
As for the control example, refer to “2.15.7 (2) Control example in the CPU reprogramming mode.”

[Beginning procedure]

① Apply 0 V to the CNV_{SS}/V_{PP} pin for reset release.
② After CPU reprogramming mode control program is transferred to internal RAM, jump to this control program on RAM. (The following operations are controlled by this control program).
③ Set “1” to the CPU reprogramming mode select bit (bit 0 of address $0FFE_{16}$).
④ Apply $V_{PP}H$ to the CNV_{SS}/V_{PP} pin.
⑤ Wait until CNV_{SS}/V_{PP} pin becomes 12 V.
⑥ Read the CPU reprogramming mode monitor flag (bit 2 of address $0FFE_{16}$) to confirm that the CPU reprogramming mode is valid.
⑦ Flash memory operations are executed by writing software-commands to the flash command register (address $0FFF_{16}$).

Note: The following procedures are also necessary.
- Control for data which is input from the external (serial I/O etc.) and to be programmed to the flash memory.
- Initial setting for ports, etc.
- Writing to the watchdog timer

[Release procedure]

① Apply 0 V to the CNV_{SS}/V_{PP} pin.
② Wait until CNV_{SS}/V_{PP} pin becomes 0 V.
③ Set the CPU reprogramming mode select bit (bit 0 of address $0FFE_{16}$) to “0”.

Also, execute the following processing before the CPU reprogramming mode is selected so that interrupts will not occur during the CPU reprogramming mode.

• Set the interrupt disable flag (I) to "1"

In the CPU reprogramming mode, write to the watchdog timer control register (address $1E_{16}$) periodically to prevent the generation of a reset by the underflow of the watchdog timer H.

In the program state (programming time: max. 9.5 μs), watchdog timer H and L are set to "FF_{16}", and the count stop. The count is started again after the program state or the erase state is completed. Accordingly, the write period of the watchdog timer control register is calculated except for the program time and erase time.

When the interrupt request or reset occurs in the CPU reprogramming mode, the microcomputer enters the following states;

(1) Interrupt

This may cause a program runaway because the flash memory that has an interrupt vector area cannot be read.

(2) Underflow of watchdog timer H, reset

This may cause a microcomputer reset; the built-in flash memory control circuit and the flash memory control register are reset.

Also, note that, when the interrupt or reset occurs during program/erase, error data may still exist after reset release because the reprogramming of the flash memory has not been completed. In this case, setting the proper program code to the flash memory in the parallel I/O mode or serial I/O mode is required.

2.15.7 Flash memory mode application examples

The control pin processing example on the system board in the serial I/O mode and the control example in the CPU reprogramming mode are described below.

(1) Control pin connection example on the system board in serial I/O mode

As shown in Figure 2.15.5, in the serial I/O mode, the built-in flash memory can be reprogrammed with the microcomputer mounted on board. Connection examples of control pins ($P3_7$, $P4_4$, $P4_6$, $P4_7$, CNV_{SS} and $\overline{RESET}$ pin) in the serial I/O mode are described below.

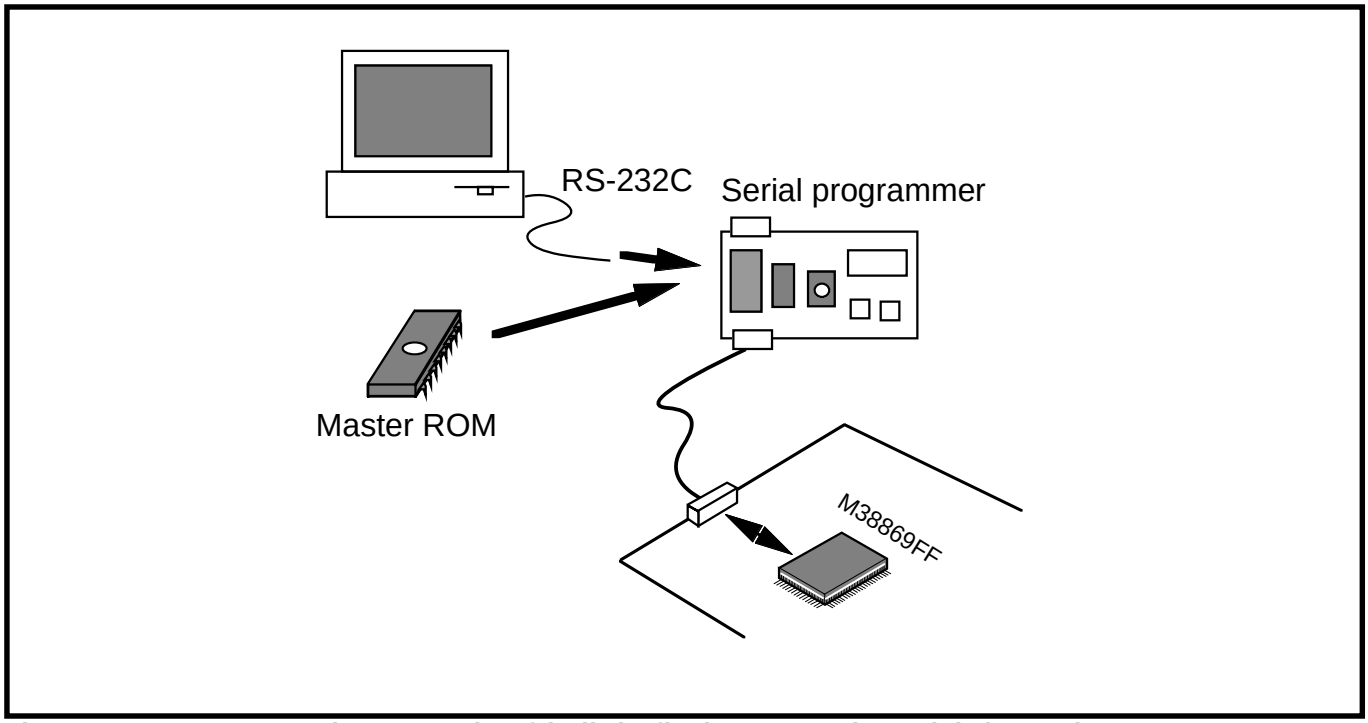


Fig. 2.15.5 Reprogramming example of built-in flash memory in serial I/O mode

① **When control signals are not affected to user system circuit**

When the control signals in the serial I/O mode are not used or not affected to the user system circuit, they can be connected as shown in Figure 2.15.6.

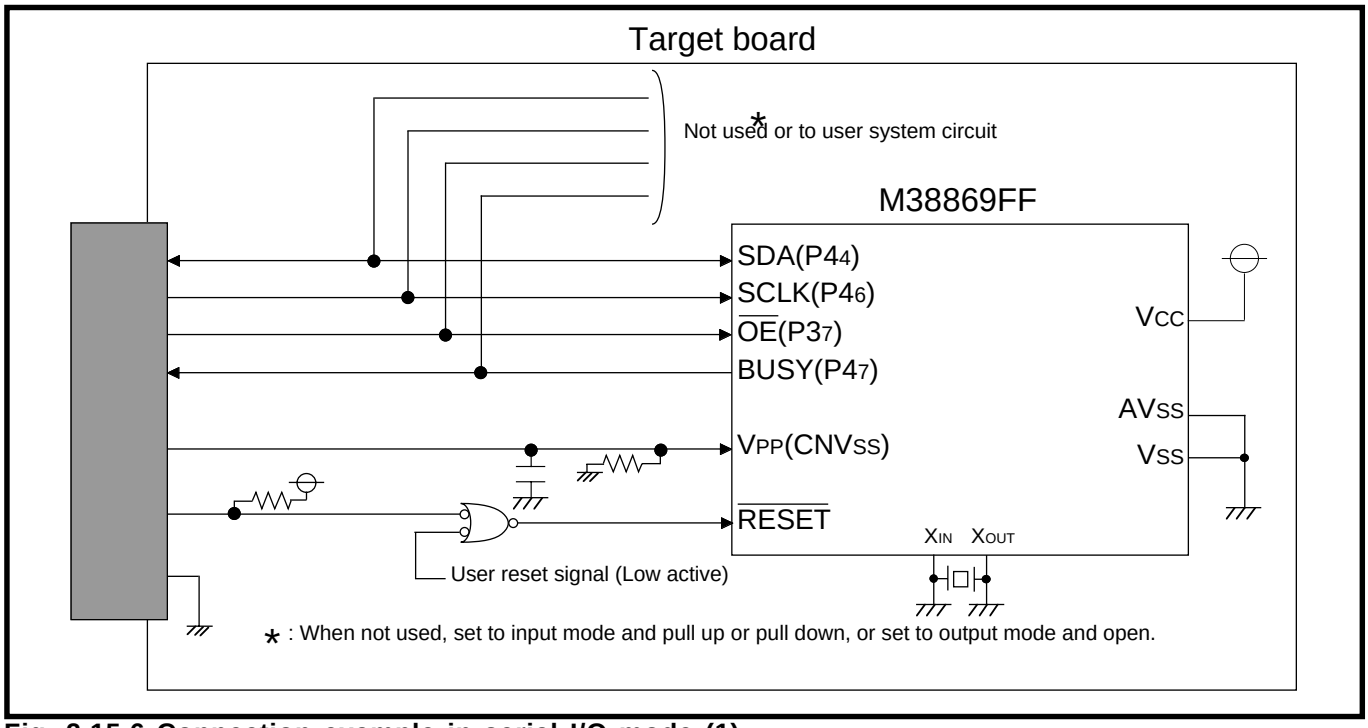


Fig. 2.15.6 Connection example in serial I/O mode (1)

② **When control signals are affected to user system circuit-1**

Figure 2.15.7 shows an example that the jumper switch cut-off the control signals not to supply to the user system circuit in the serial I/O mode.

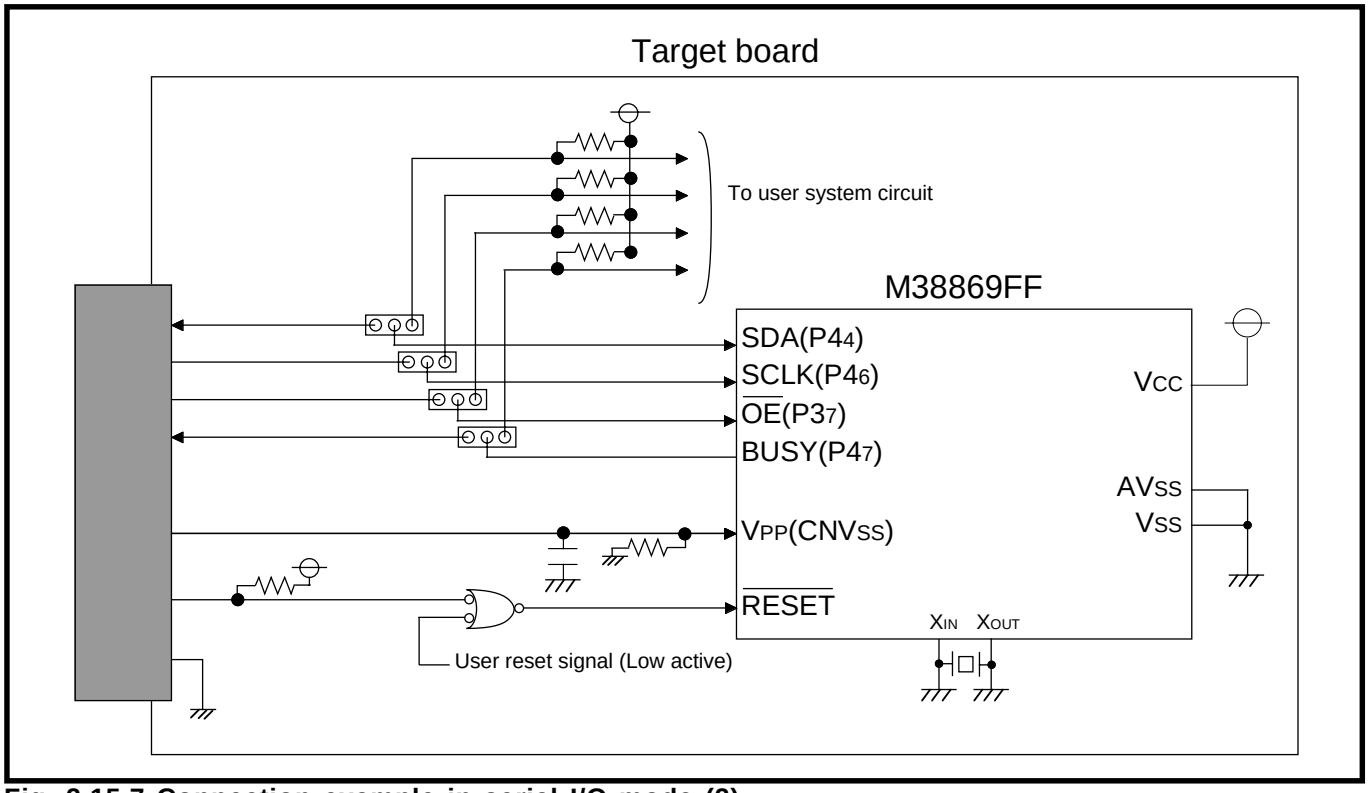


Fig. 2.15.7 Connection example in serial I/O mode (2)

③ **When control signals are affected to user system circuit-2**

Figure 2.15.8 shows an example that the analog switch (74HC4066) cut-off the control signals not to supply to the user system circuit in the serial I/O mode.

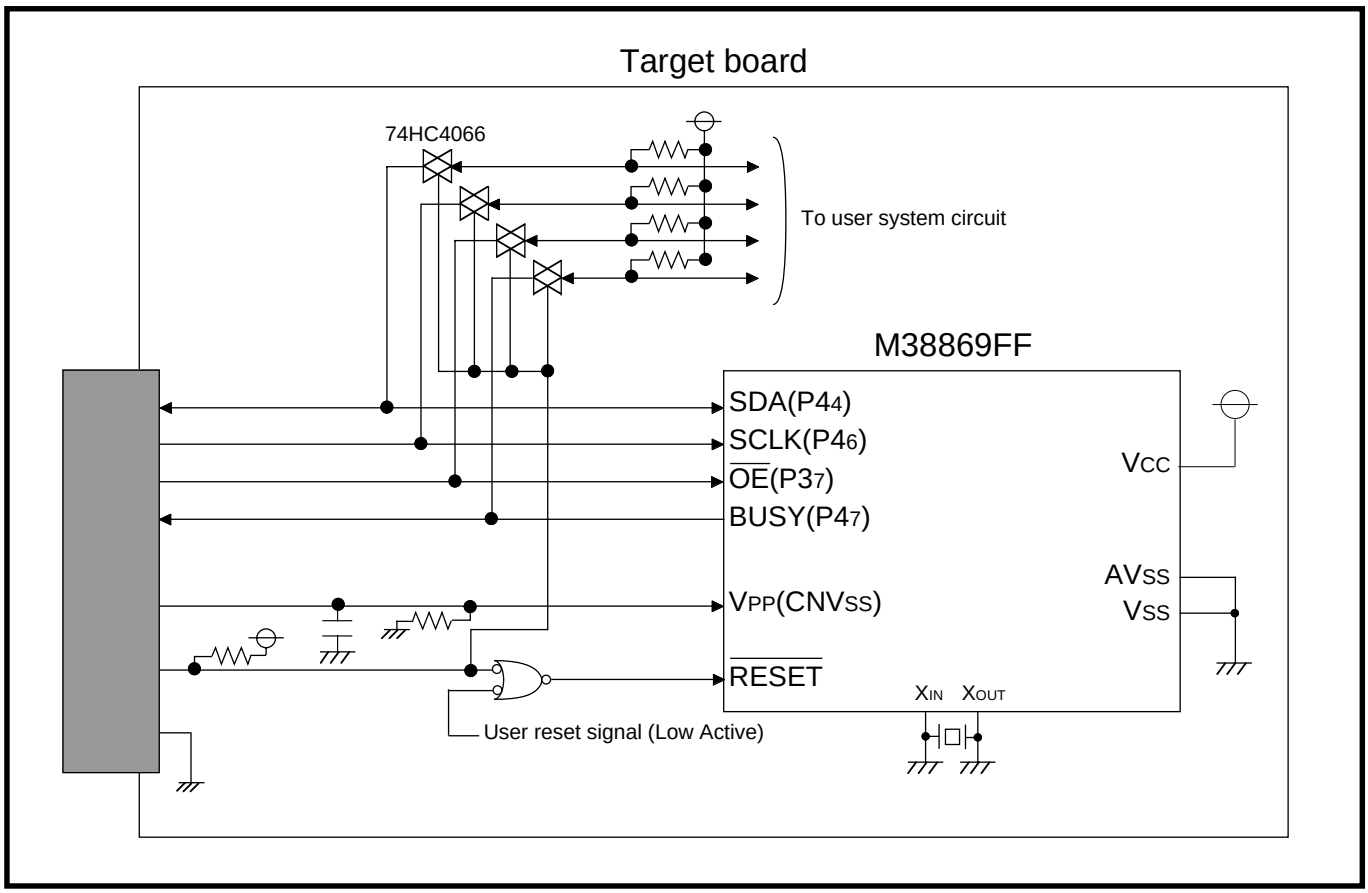


Fig. 2.15.8 Connection example in serial I/O mode (3)

(2) Control example in CPU reprogramming mode

In this example, the built-in flash memory is reprogrammed in the CPU reprogramming mode by serial I/O, receiving the reprogramming data (updated data).

Figure 2.15.9 shows an example of the reprogramming system for the built-in flash memory in the CPU reprogramming mode.

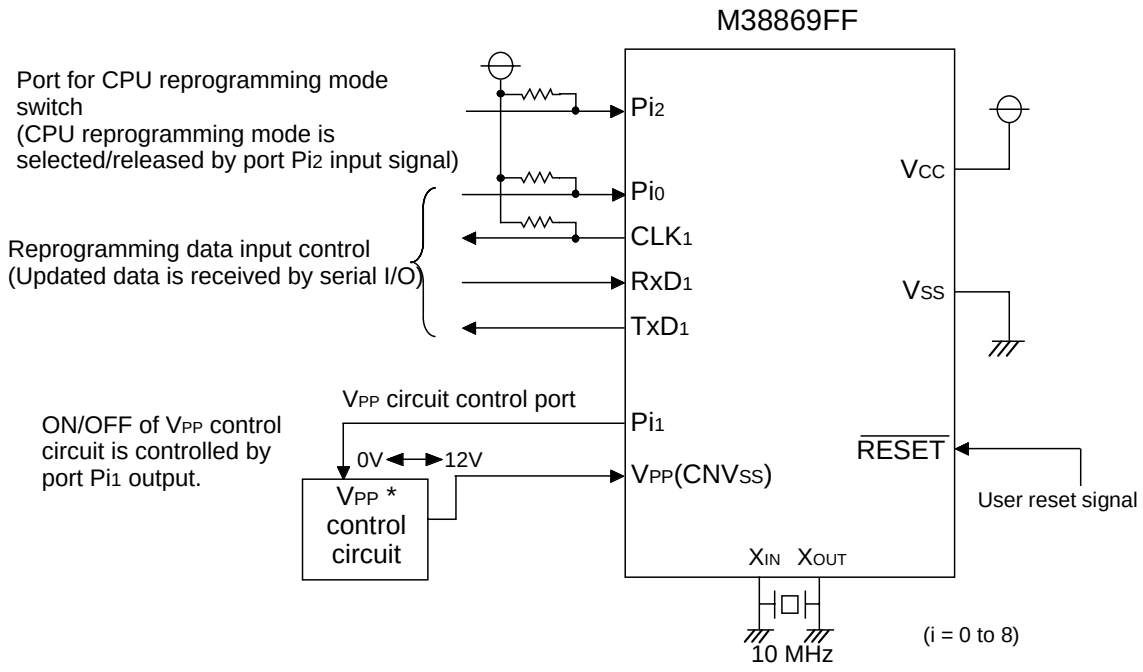


Fig. 2.15.9 Example of reprogramming system for built-in flash memory in CPU reprogramming mode

● **Specifications**

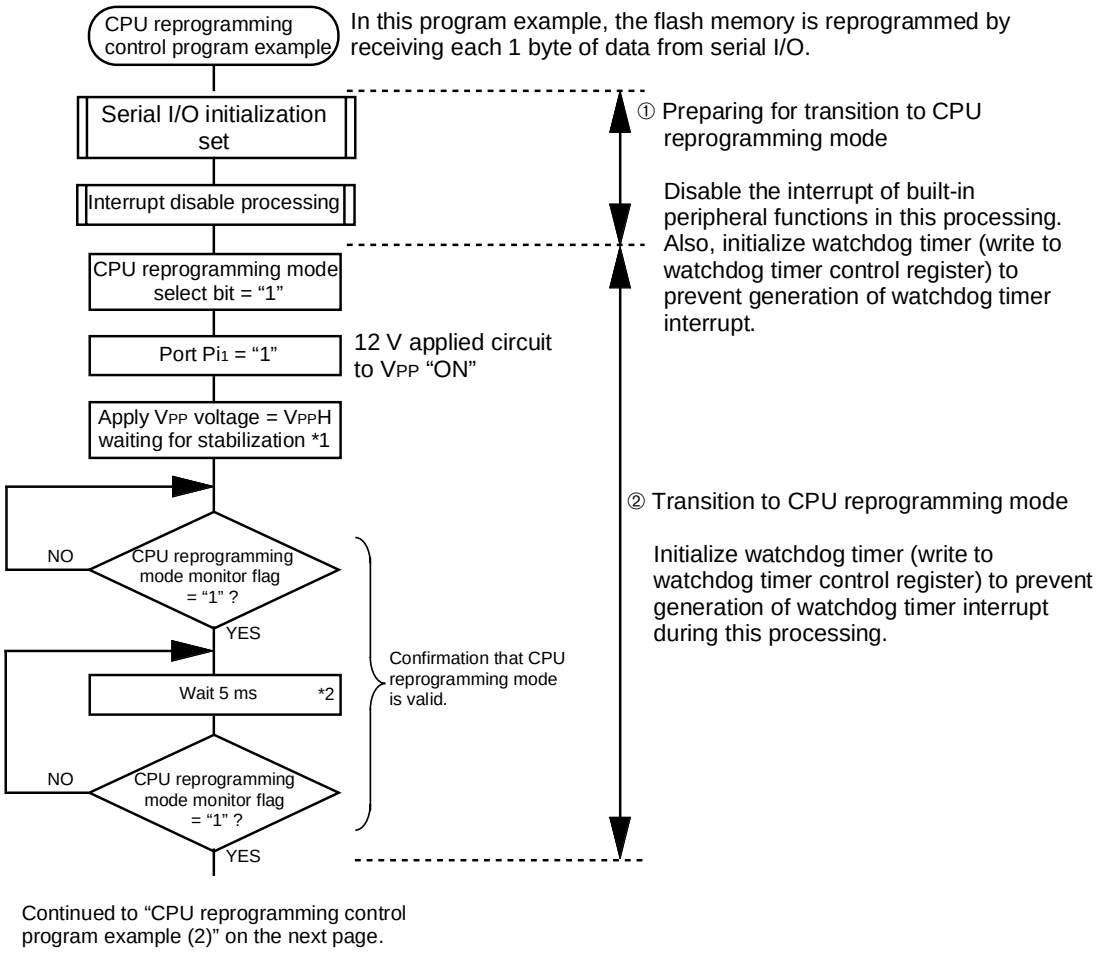
① CPU reprogramming mode is selected/released by the input signal to Pi_2.

② Updated data is received by serial I/O.

③ The transfer enable state of serial transmit side is judged by "L" level input to Pi_0.

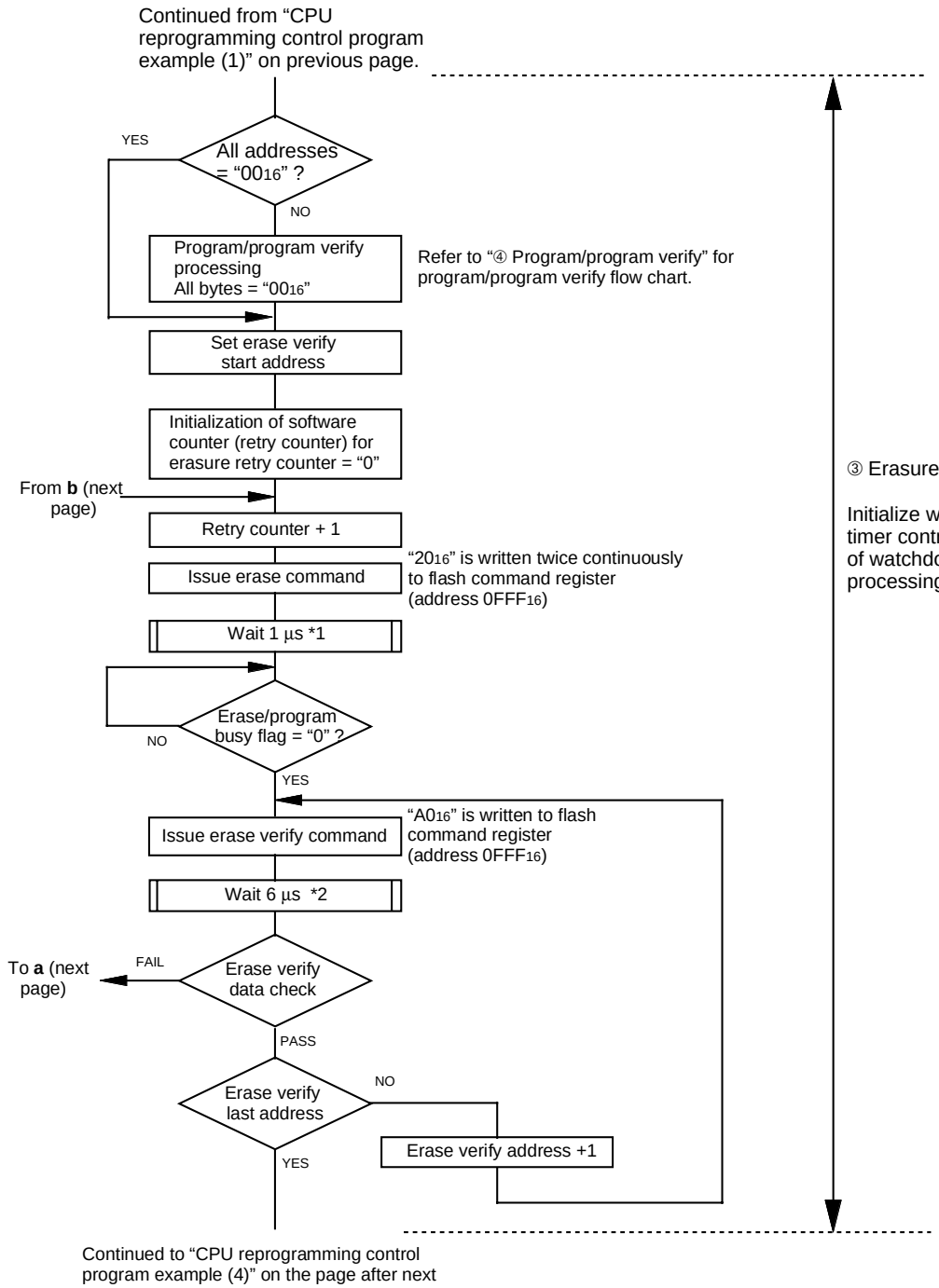
④ V_{PP} control circuit is turned ON/OFF by the output from Pi_1 (refer to Figure 2.15.14 and Figure 2.15.15).

Note: In this example, the following program is transferred to and executed on the internal RAM.



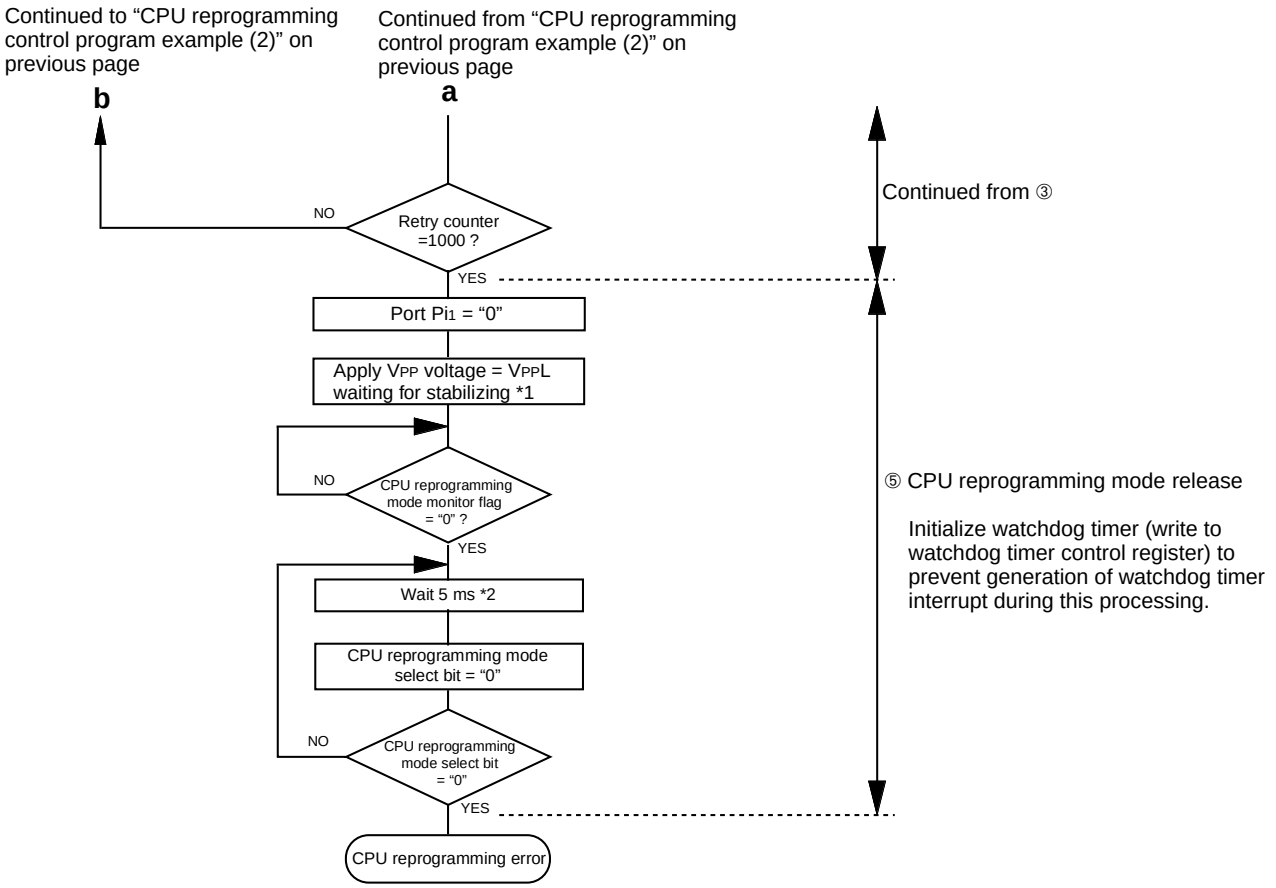
*1: Waiting by software until V_{PP} input voltage is stabilized at V_{PPH} is recommended. (Refer to Figures 2.15.14 and 2.15.15 V_{PP} voltage control timing Ⓐ.)
*2: The waiting time depends on V_{PP} control circuit (Refer to Figures 2.15.14 and 2.15.15 V_{PP} voltage control timing Ⓒ.)

Fig. 2.15.10 CPU reprogramming control program example (1)



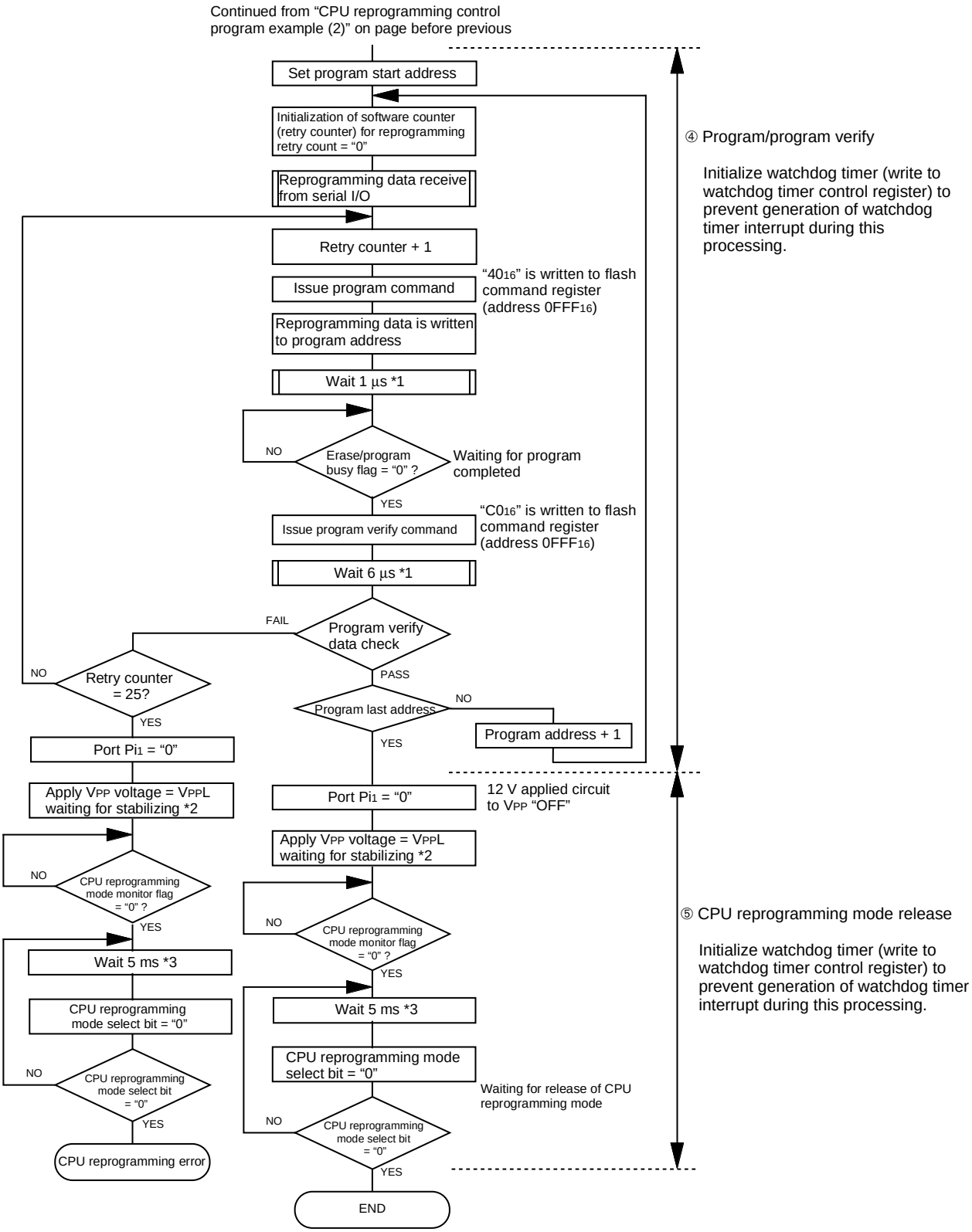
*1: The wait processing time shown in the flow chart is required regardless of the external clock input frequency.
*2: The waiting time depends on V_{PP} control circuit (Refer to Figure 2.15.14 and Figure 2.15.15 V_{PP} voltage control timing©.)

Fig. 2.15.11 CPU reprogramming control program example (2)



*1: Waiting by software until VPP input voltage is stabilized at VPPL is recommended. (Refer to Figure 2.15.14 and Figure 2.15.15 VPP voltage control timing Ⓑ)
*2: The wait time depends on VPP control circuit (Refer to Figure 2.15.14 and Figure 2.15.15 VPP voltage control timing Ⓒ)

Fig. 2.15.12 CPU reprogramming control program example (3)



*1: The wait processing time shown in the flow chart is required regardless of the external clock input frequency.
*2: Waiting by software until VPP input voltage is stabilized at VPPL is recommended. (Refer to Figure 2.15.14 and Figure 2.15.15 VPP voltage control timing Ⓑ)
*3: The waiting time depends on VPP control circuit (Refer to Figure 2.15.14 and Figure 2.15.15 VPP voltage control timing Ⓒ)

Fig. 2.15.13 CPU reprogramming control program example (4)

● When 12 V voltage is supplied to target system

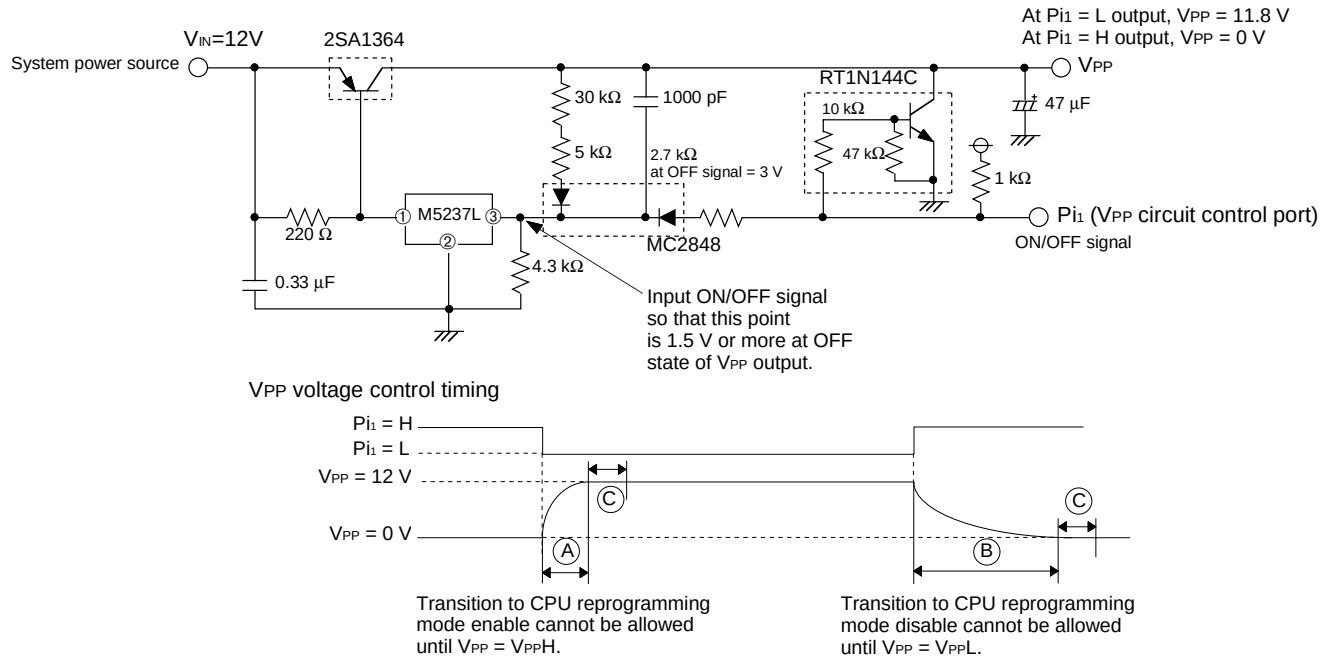


Fig. 2.15.14 V_{PP} control circuit example (1)

● When only 5 V voltage is supplied to target system

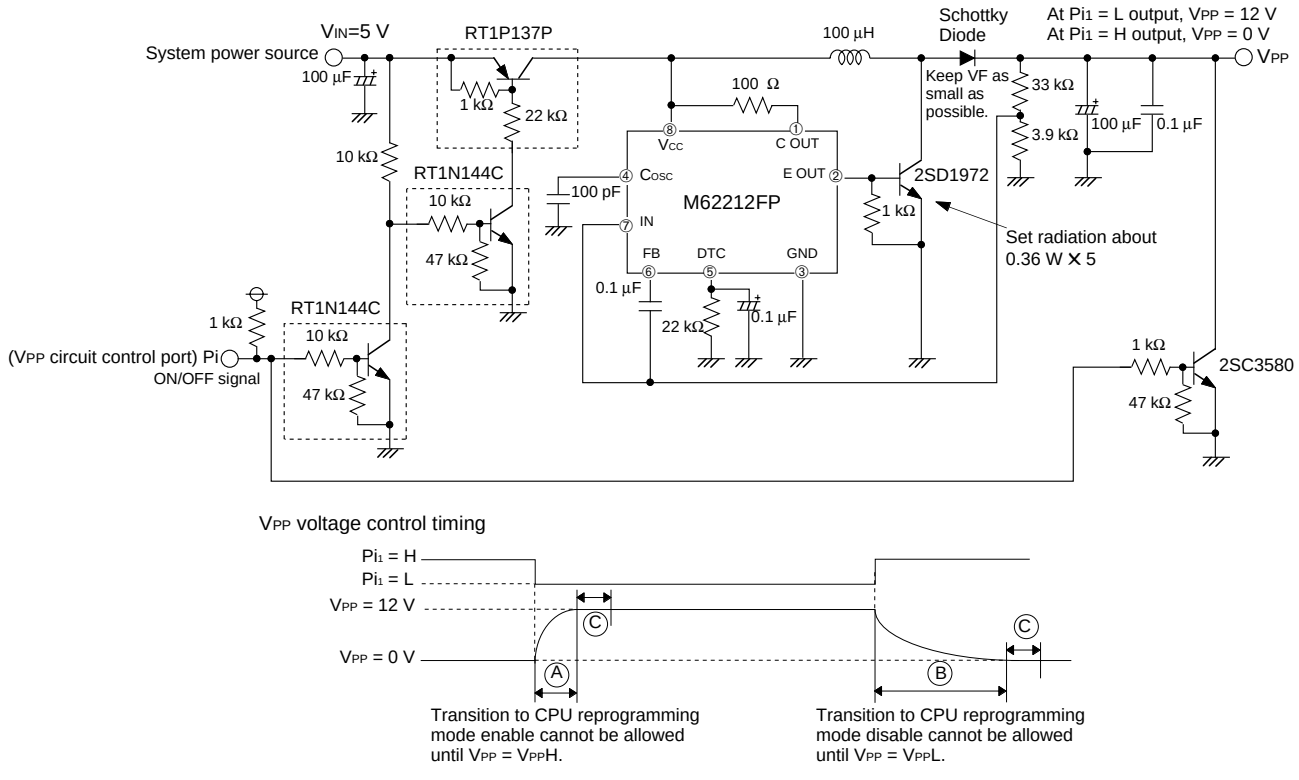


Fig. 2.15.15 V_{PP} control circuit example (2)

2.15 Flash memory

2.15.8 Notes on CPU reprogramming mode

(1) Transfer the CPU reprogramming mode control program to the internal RAM before selecting the CPU reprogramming mode, and then, execute it on the internal RAM. Additionally, when the subroutine or stack operation instruction is used in the control program, make sure the control program is not destroyed by the stack operation.

(2) Make sure each instruction description (specified address etc.) is correct, because the CPU reprogramming mode control program is transferred to the internal RAM and executed on the internal RAM.

(3) In order to avoid generation of a watchdog timer reset, write to the watchdog timer control register periodically during the CPU reprogramming mode control program (refer to “2.7 Watchdog timer”).

(4) Notes on flash memory version

The CNV_{SS} pin is connected to the internal memory circuit block by a low-ohmic resistance, since it works as a program power source pin (V_{PP} pin), as well.

To improve the noise margin, connect the CNV_{SS} pin to V_{SS} through 1 to 10 kΩ resistor.

When the CNV_{SS} pin of the mask ROM version is connected to Vss through this resistor, the function of mask ROM version works well in the same manner as flash memory version.

CHAPTER 3

APPENDIX

APPENDIX

3.1 Electrical characteristics

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 voltages **(Note 1)**	All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to 7.0	V
V_{CC}	Power source voltages **(Note 2)**		–0.3 to 6.5	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P8_0$–$P8_7$, V_{REF}		–0.3 to V_{CC} +0.3	V
V_I	Input voltage $P7_0$–$P7_7$		–0.3 to 5.8	V
V_I	Input voltage $\overline{RESET}$, X_{IN}		–0.3 to V_{CC} +0.3	V
V_I	Input voltage CNV_{SS} **(Note 3)**		–0.3 to 7	V
V_I	Input voltage CNV_{SS} **(Note 4)**		–0.3 to V_{CC} +0.3	V
V_I	Input voltage CNV_{SS} **(Note 5)**		–0.3 to 13	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P8_0$–$P8_7$, X_{OUT}		–0.3 to V_{CC} +0.3	V
V_O	Output voltage $P7_0$–$P7_7$		–0.3 to 5.8	V
P_d	Power dissipation	T_a = 25 °C	500	mW
T_{opr}	Operating temperature		–20 to 85	°C
T_{stg}	Storage temperature		–40 to 125	°C

Notes 1: M38867M8A, M38867E8A
2: M38869M8A, M38869MCA, M38869MFA, M38869FFA
3: M38867M8A
4: M38869M8A, M38869MCA, M38869MFA
5: M38867E8A, M38869FFA

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions (1)

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage ($f(X_{IN}) \leq 4.1$ MHz)		2.7	5.0	5.5	V
	Power source voltage ($f(X_{IN}) = 10$ MHz)		4.0	5.0	5.5	
V_{CC}	Power source voltage (flash memory version)		4.0	5.0	5.5	V
V_{SS}	Power source voltage			0		V
V_{REF}	Analog reference voltage (when A-D converter is used)		2.0		V_{CC}	V
	Analog reference voltage (when D-A converter is used)		2.7		V_{CC}	
AV_{SS}	Analog power source voltage			0		V
V_{IA}	A-D converter input voltage	AN_0–AN_7	AV_{SS}		V_{CC}	V
V_{IH}	"H" input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$, $P4_1$, $P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P8_0$–$P8_7$	0.8V_{CC}		V_{CC}	V
V_{IH}	"H" input voltage	$P7_6$, $P7_7$	0.8V_{CC}		5.5	V
V_{IH}	"H" input voltage (when I^2C-BUS input level is selected)	S_{DA}, S_{CL}	0.7V_{CC}		5.5	V
V_{IH}	"H" input voltage (when SMBUS input level is selected)	S_{DA}, S_{CL}	1.4		5.5	V
V_{IH}	"H" input voltage (when CMOS input level is selected)	$P4_2$–$P4_6$, DQ_0–DQ_7, $\overline{W}$, $\overline{R}$, $\overline{S_0}$, $\overline{S_1}$, A_0	0.8V_{CC}		V_{CC}	V
V_{IH}	"H" input voltage (when CMOS input level is selected)	$P7_0$–$P7_5$	0.8V_{CC}		5.5	V
V_{IH}	"H" input voltage (when TTL input level is selected)	$P4_2$–$P4_6$, DQ_0–DQ_7, $\overline{W}$, $\overline{R}$, $\overline{S_0}$, $\overline{S_1}$, A_0 **(Note)**	2.0		V_{CC}	V
V_{IH}	"H" input voltage (when TTL input level is selected)	$P7_0$–$P7_5$ **(Note)**	2.0		5.5	V
V_{IH}	"H" input voltage	$\overline{RESET}$, X_{IN}, X_{CIN}, CNV_{SS}	0.8V_{CC}		V_{CC}	V
V_{IL}	"L" input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	0		0.2V_{CC}	V
V_{IL}	"L" input voltage (when I^2C-BUS input level is selected)	S_{DA}, S_{CL}	0		0.3V_{CC}	V
V_{IL}	"L" input voltage (when SMBUS input level is selected)	S_{DA}, S_{CL}	0		0.6	V
V_{IL}	"L" input voltage (when CMOS input level is selected)	$P4_2$–$P4_6$, $P7_0$–$P7_5$, DQ_0–DQ_7, $\overline{W}$, $\overline{R}$, $\overline{S_0}$, $\overline{S_1}$, A_0	0		0.2V_{CC}	V
V_{IL}	"L" input voltage (when TTL input level is selected)	$P4_2$–$P4_6$, $P7_0$–$P7_5$, DQ_0–DQ_7, $\overline{W}$, $\overline{R}$, $\overline{S_0}$, $\overline{S_1}$, A_0 **(Note)**	0		0.8	V
V_{IL}	"L" input voltage	$\overline{RESET}$, CNV_{SS}	0		0.2V_{CC}	V
V_{IL}	"L" input voltage	X_{IN}, X_{CIN}	0		0.16V_{CC}	V

Note : When V_{CC} is 4.0 to 5.5 V.

APPENDIX

3.1 Electrical characteristics

Table 3.1.3 Recommended operating conditions (2)

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$\Sigma I_{OH(peak)}$	"H" total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$ **(Note)**			–80	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current	$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note)**			–80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P3_0$–$P3_7$, $P8_0$–$P8_7$ **(Note)**			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current $P2_4$–$P2_7$ **(Note)**	In single-chip mode			80	mA
		In memory expansion mode In microprocessor mode			40	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$ **(Note)**			80	mA
$\Sigma I_{OH(avg)}$	"H" total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$ **(Note)**			–40	mA
$\Sigma I_{OH(avg)}$	"H" total average output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note)**			–40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P3_0$–$P3_7$, $P8_0$–$P8_7$ **(Note)**			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current $P2_4$–$P2_7$ **(Note)**	In single-chip mode			40	mA
		In memory expansion mode In microprocessor mode			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$ **(Note)**			40	mA

Note : 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.

Table 3.1.4 Recommended operating conditions (3)

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$I_{OH(peak)}$	"H" peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P8_0$–$P8_7$ **(Note 1)**			–10	mA
$I_{OL(peak)}$	"L" peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$ **(Note 1)**			10	mA
$I_{OL(peak)}$	"L" peak output current $P2_4$–$P2_7$ **(Note 1)**	In single-chip mode			20	mA
		In memory expansion mode In microprocessor mode			10	mA
$I_{OH(avg)}$	"H" average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P8_0$–$P8_7$ **(Note 2)**			–5	mA
$I_{OL(avg)}$	"L" average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$ **(Note 2)**			5	mA
$I_{OL(avg)}$	"L" peak output current $P2_4$–$P2_7$ **(Note 2)**	In single-chip mode			15	mA
		In memory expansion mode In microprocessor mode			5	mA
$f(X_{IN})$	Main clock input oscillation frequency **(Note 3)**	High-speed mode 4.0 V≤ V_{CC} ≤ 5.5 V			10	MHz
		High-speed mode 2.7 V≤ V_{CC} ≤ 4.0 V			4.5 V_{CC}–8	MHz
		Middle-speed mode 4.0 V≤ V_{CC} ≤ 5.5 V			10	MHz
		Middle-speed mode 2.7 V≤ V_{CC} ≤ 4.0 V **(Note 5)**			10	MHz
		Middle-speed mode 2.7 V≤ V_{CC} ≤ 4.0 V **(Note 5)**			4.5 V_{CC}–8	MHz
$f(X_{CIN})$	Sub-clock input oscillation frequency **(Notes 3, 4)**			32.768	50	kHz

Notes 1: The peak output current is the peak current flowing in each port.
2: The average output current I_{OL}(avg), I_{OH}(avg) are average value measured over 100 ms.
3: When the oscillation frequency has a duty cycle of 50%.
4: When using the microcomputer in low-speed mode, set the sub-clock input oscillation frequency on condition that $f(X_{CIN}) < f(X_{IN})/3$.
5: When using the timer X/Y, timer 1/2, serial I/O1, serial I/O2, A-D converter, comparator, and PWM, set the main clock input oscillation frequency to the max. 4.5V_{CC}–8 (MHz).

3.1.3 Electrical characteristics

Table 3.1.5 Electrical characteristics (1)

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	"H" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$ $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$ $P6_0$–$P6_7$, $P8_0$–$P8_7$ **(Note)**	I_{OH} = –10 mA V_{CC} = 4.0 to 5.5 V	V_{CC}–2.0			V
		I_{OH} = –1.0 mA V_{CC} = 2.7 to 5.5 V	V_{CC}–1.0			V
V_{OL}	"L" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$ $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$ $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	I_{OL} = 10 mA V_{CC} = 4.0 to 5.5 V			2.0	V
		I_{OL} = 1.6 mA V_{CC} = 2.7 to 5.5 V			0.4	V
V_{T+}–V_{T-}	Hysteresis $CNTR_0$, $CNTR_1$, INT_0, INT_1 INT_{20}–INT_{40}, INT_{21}–INT_{41} $P3_0$–$P3_7$			0.4		V
V_{T+}–V_{T-}	Hysteresis RxD, S_{CLK1}, S_{IN2}, S_{CLK2}			0.5		V
V_{T+}–V_{T-}	Hysteresis $\overline{RESET}$			0.5		V
I_{IH}	"H" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$ $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$ $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	V_I = V_{CC} (Pin floating. Pull-up transistors "off")			5.0	µA
I_{IH}	"H" input current $\overline{RESET}$, CNV_{SS}	V_I = V_{CC}			5.0	µA
I_{IH}	"H" input current X_{IN}	V_I = V_{CC}		4		µA
I_{IL}	"L" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$ $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$ $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	V_I = V_{SS} (Pin floating. Pull-up transistors "off")			–5.0	µA
I_{IL}	"L" input current $\overline{RESET}$,CNV_{SS}	V_I = V_{SS}			–5.0	µA
I_{IL}	"L" input current X_{IN}	V_I = V_{SS}		–4		µA
I_{IL}	"L" input current $P3_0$–$P3_7$ (at Pull-up)	V_I = V_{SS} V_{CC} = 4.0 to 5.5 V	–20	–60	–120	µA
		V_I = V_{SS} V_{CC} = 2.7 to 5.5 V	–10			µA
V_{RAM}	RAM hold voltage	When clock stopped	2.0		5.5	V

Note: $P0_0$–$P0_3$ are measured when the $P0_0$–$P0_3$ output structure selection bit of the port control register 1 (bit 0 of address $002E_{16}$) is "0".
$P0_4$–$P0_7$ are measured when the $P0_4$–$P0_7$ output structure selection bit of the port control register 1 (bit 1 of address $002E_{16}$) is "0".
$P1_0$–$P1_3$ are measured when the $P1_0$–$P1_3$ output structure selection bit of the port control register 1 (bit 2 of address $002E_{16}$) is "0".
$P1_4$–$P1_7$ are measured when the $P1_4$–$P1_7$ output structure selection bit of the port control register 1 (bit 3 of address $002E_{16}$) is "0".
$P4_2$, $P4_3$, $P4_4$, and $P4_6$ are measured when the P4 output structure selection bit of the port control register 2 (bit 2 of address $002F_{16}$) is "0".
$P4_5$ is measured when the $P4_5$/TxD P-channel output disable bit of the UART control register (bit 4 of address $001B_{16}$) is "0".

APPENDIX

3.1 Electrical characteristics

Table 3.1.6 Electrical characteristics (2)

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
I_{CC}	Power source current	High-speed mode $f(X_{IN})$ = 10 MHz $f(X_{CIN})$ = 32.768 kHz Output transistors “off”			8.0	15	mA
		High-speed mode $f(X_{IN})$ = 8 MHz $f(X_{CIN})$ = 32.768 kHz Output transistors “off”			6.8	13	mA
		High-speed mode $f(X_{IN})$ = 10 MHz (in WIT state) $f(X_{CIN})$ = 32.768 kHz Output transistors “off”			1.6		mA
		Low-speed mode $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz Output transistors “off”			60	200	µA
		Low-speed mode $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz (in WIT state) Output transistors “off”			20	40	µA
		Low-speed mode (V_{CC} = 3 V) $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz Output transistors “off”			20	55	µA
		Low-speed mode (V_{CC} = 3 V) $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz (in WIT state) Output transistors “off”			8.0	20.0	µA
		Middle-speed mode $f(X_{IN})$ = 10 MHz $f(X_{CIN})$ = stopped Output transistors “off”			4.0	7.0	mA
		Middle-speed mode $f(X_{IN})$ = 10 MHz (in WIT state) $f(X_{CIN})$ = stopped Output transistors “off”			1.5		mA
		Increment when A-D conversion is executed $f(X_{IN})$ = 10 MHz			800		µA
		All oscillation stopped (in STP state) Output transistors “off”	Ta = 25 °C		0.1	1.0	µA
			Ta = 85 °C			10	µA

3.1.4 A-D converter characteristics

Table 3.1.7 A-D converter characteristics (1)

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, V_{REF} = 2.0 V to V_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

10-bit A-D mode (when conversion mode selection bit (bit 7 of address 0038_{16}) is "0")

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
–	Resolution					10	bit
–	Absolute accuracy (excluding quantization error)		V_{CC} = V_{REF} = 5.0 V			±4	LSB
t_{CONV}	Conversion time					61	2tc(X_{IN})
R_{LADDER}	Ladder resistor			12	35	100	kΩ
I_{VREF}	Reference power source input current	at A-D converter operated	V_{REF} = 5.0 V	50	150	200	µA
		at A-D converter stopped	V_{REF} = 5.0 V			5	µA
$I_{I(AD)}$	A-D port input current					5.0	µA

Table 3.1.8 A-D converter characteristics (2)

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, V_{REF} = 2.0 V to V_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

8-bit A-D mode (when conversion mode selection bit (bit 7 of address 0038_{16}) is "1")

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
–	Resolution					8	bit
–	Absolute accuracy (excluding quantization error)		V_{CC} = V_{REF} = 5.0 V			±2	LSB
t_{CONV}	Conversion time					50	2tc(X_{IN})
R_{LADDER}	Ladder resistor			12	35	100	kΩ
I_{VREF}	Reference power source input current	at A-D converter operated	V_{REF} = 5.0 V	50	150	200	µA
		at A-D converter stopped	V_{REF} = 5.0 V			5	µA
$I_{I(AD)}$	A-D port input current					5.0	µA

3.1.5 D-A converter characteristics

Table 3.1.9 D-A converter characteristics

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, V_{REF} = 2.7 V to V_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
–	Resolution					8	Bits
–	Absolute accuracy	V_{CC} = 4.0 to 5.5 V				1.0	%
		V_{CC} = 2.7 to 4.0 V				2.5	%
tsu	Setting time					3	µs
RO	Output resistor			1	2.5	4	kΩ
I_{VREF}	Reference power source input current (Note 1)					3.2	mA

Note 1: Using one D-A converter, with the value in the D-A conversion register of the other D-A converter being "00_{16}".

3.1.6 Comparator characteristics

Table 3.1.10 Comparator characteristics

(V_{CC} = 2.7 to 5.5 V, V_{CC} = 4.0 to 5.5 V for flash memory version, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
–	Absolute accuracy	1LSB = V_{CC}/16			1/2	LSB
T_{CONV}	Conversion time	at 10 MHz operating			2.8	µs
		at 8 MHz operating			3.5	µs
		at 4 MHz operating			7	µs
V_{IA}	Analog input voltage		0		V_{CC}	V
I_{IA}	Analog input current				5.0	µA
R_{LADDER}	Ladder resistor		20	40	50	kΩ
CMP_{REF}	Internal reference voltage			29V_{CC} /32		V
	External reference input voltage		V_{CC}/32		V_{CC}	V

APPENDIX

3.1 Electrical characteristics

3.1.7 Timing requirements

Table 3.1.11 Timing requirements (1)
(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	16			X_{IN} cycles
$t_{C(XIN)}$	Main clock input cycle time	100			ns
$t_{WH(XIN)}$	Main clock input “H” pulse width	40			ns
$t_{WL(XIN)}$	Main clock input “L” pulse width	40			ns
$t_{C(XCIN)}$	Sub-clock input cycle time	20			µs
$t_{WH(XCIN)}$	Sub-clock input “H” pulse width	5			µs
$t_{WL(XCIN)}$	Sub-clock input “L” pulse width	5			µs
$t_{C(CNTR)}$	$CNTR_0$, $CNTR_1$ input cycle time	200			ns
$t_{WH(CNTR)}$	$CNTR_0$, $CNTR_1$ input “H” pulse width	80			ns
$t_{WL(CNTR)}$	$CNTR_0$, $CNTR_1$ input “L” pulse width	80			ns
$t_{WH(INT)}$	INT_0, INT_1, INT_{20}, INT_{30}, INT_{40}, INT_{21}, INT_{31}, INT_{41} input “H” pulse width	80			ns
$t_{WL(INT)}$	INT_0, INT_1, INT_{20}, INT_{30}, INT_{40}, INT_{21}, INT_{31}, INT_{41} input “L” pulse width	80			ns
$t_{C(SCLK1)}$	Serial I/O1 clock input cycle time **(Note)**	800			ns
$t_{WH(SCLK1)}$	Serial I/O1 clock input “H” pulse width **(Note)**	370			ns
$t_{WL(SCLK1)}$	Serial I/O1 clock input “L” pulse width **(Note)**	370			ns
$t_{su(RxD-SCLK1)}$	Serial I/O1 input setup time	220			ns
$t_{h(SCLK1-RxD)}$	Serial I/O1 input hold time	100			ns
$t_{C(SCLK2)}$	Serial I/O2 clock input cycle time	1000			ns
$t_{WH(SCLK2)}$	Serial I/O2 clock input “H” pulse width	400			ns
$t_{WL(SCLK2)}$	Serial I/O2 clock input “L” pulse width	400			ns
$t_{su(SIN2-SCLK2)}$	Serial I/O2 input setup time	200			ns
$t_{h(SCLK2-SIN2)}$	Serial I/O2 input hold time	200			ns

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

Table 3.1.12 Timing requirements (2)

(V_{CC} = 2.7 to 4.0 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	16			X_{IN} cycles
$t_{C(X_{IN})}$	Main clock input cycle time	1000/(4.5V_{CC}–8)			ns
$t_{WH(X_{IN})}$	Main clock input “H” pulse width	400/(4.5V_{CC}–8)			ns
$t_{WL(X_{IN})}$	Main clock input “L” pulse width	400/(4.5V_{CC}–8)			ns
$t_{C(X_{CIN})}$	Sub-clock input cycle time	20			µs
$t_{WH(X_{CIN})}$	Sub-clock input “H” pulse width	5			µs
$t_{WL(X_{CIN})}$	Sub-clock input “L” pulse width	5			µs
$t_{C(CNTR)}$	$CNTR_0$, $CNTR_1$ input cycle time	500			ns
$t_{WH(CNTR)}$	$CNTR_0$, $CNTR_1$ input “H” pulse width	230			ns
$t_{WL(CNTR)}$	$CNTR_0$, $CNTR_1$ input “L” pulse width	230			ns
$t_{WH(INT)}$	INT_0, INT_1, INT_{20}, INT_{30}, INT_{40}, INT_{21}, INT_{31}, INT_{41} input “H” pulse width	230			ns
$t_{WL(INT)}$	INT_0, INT_1, INT_{20}, INT_{30}, INT_{40}, INT_{21}, INT_{31}, INT_{41} input “L” pulse width	230			ns
$t_{C(SCLK1)}$	Serial I/O1 clock input cycle time **(Note)**	2000			ns
$t_{WH(SCLK1)}$	Serial I/O1 clock input “H” pulse width **(Note)**	950			ns
$t_{WL(SCLK1)}$	Serial I/O1 clock input “L” pulse width **(Note)**	950			ns
$t_{su(RxD\text{-}SCLK1)}$	Serial I/O1 input setup time	400			ns
$t_{h(SCLK1\text{-}RxD)}$	Serial I/O1 input hold time	200			ns
$t_{C(SCLK2)}$	Serial I/O2 clock input cycle time	2000			ns
$t_{WH(SCLK2)}$	Serial I/O2 clock input “H” pulse width	950			ns
$t_{WL(SCLK2)}$	Serial I/O2 clock input “L” pulse width	950			ns
$t_{su(SIN2\text{-}SCLK2)}$	Serial I/O2 input setup time	400			ns
$t_{h(SCLK2\text{-}SIN2)}$	Serial I/O2 input hold time	300			ns

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

APPENDIX

3.1 Electrical characteristics

3.1.8 Timing requirements for system bus interface

Table 3.1.13 Timing requirements for system bus interface (1)
(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{su} (S-R)	$\overline{S0}$, $\overline{S1}$ setup time	0			ns
t_{su} (S-W)	$\overline{S0}$, $\overline{S1}$ setup time	0			ns
t_h (R-S)	$\overline{S0}$, $\overline{S1}$ hold time	0			ns
t_h (W-S)	$\overline{S0}$, $\overline{S1}$ hold time	0			ns
t_{su} (A-R)	A0 setup time	10			ns
t_{su} (A-W)	A0 setup time	10			ns
t_h (R-A)	A0 hold time	0			ns
t_h (W-A)	A0 hold time	0			ns
t_w (R)	Read pulse width	120			ns
t_w (W)	Write pulse width	120			ns
t_{su} (D-W)	Before write data input setup time	50			ns
t_h (W-D)	After write data input hold time	0			ns

Table 3.1.14 Timing requirements for system bus interface (2)
(V_{CC} = 2.7 to 4.0 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{su} (S-R)	$\overline{S0}$, $\overline{S1}$ setup time	0			ns
t_{su} (S-W)	$\overline{S0}$, $\overline{S1}$ setup time	0			ns
t_h (R-S)	$\overline{S0}$, $\overline{S1}$ hold time	0			ns
t_h (W-S)	$\overline{S0}$, $\overline{S1}$ hold time	0			ns
t_{su} (A-R)	A0 setup time	30			ns
t_{su} (A-W)	A0 setup time	30			ns
t_h (R-A)	A0 hold time	0			ns
t_h (W-A)	A0 hold time	0			ns
t_w (R)	Read pulse width	250			ns
t_w (W)	Write pulse width	250			ns
t_{su} (D-W)	Before write data input setup time	130			ns
t_h (W-D)	After write data input hold time	0			ns

3.1.9 Switching characteristics

Table 3.1.15 Switching characteristics (1)
(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t_{WH} (S_{CLK1})	Serial I/O1 clock output “H” pulse width	Fig. 3.1.1	$t_{C(SCLK1)}$/2–30			ns
t_{WL} (S_{CLK1})	Serial I/O1 clock output “L” pulse width		$t_{C(SCLK1)}$/2–30			ns
td (S_{CLK1}-TxD)	Serial I/O1 output delay time **(Note 1)**				140	ns
tv (S_{CLK1}-TxD)	Serial I/O1 output valid time **(Note 1)**		–30			ns
tr (S_{CLK1})	Serial I/O1 clock output rising time				30	ns
tf (S_{CLK1})	Serial I/O1 clock output falling time				30	ns
t_{WH} (S_{CLK2})	Serial I/O2 clock output “H” pulse width	Fig. 3.1.2	$t_{C(SCLK2)}$/2–160			ns
t_{WL} (S_{CLK2})	Serial I/O2 clock output “L” pulse width		$t_{C(SCLK2)}$/2–160			ns
td (S_{CLK2}-S_{OUT2})	Serial I/O2 output delay time				200	ns
tv (S_{CLK2}-S_{OUT2})	Serial I/O2 output valid time		0			ns
tf (S_{CLK2})	Serial I/O2 clock output falling time				30	ns
tr (CMOS)	CMOS output rising time **(Note 2)**	Fig. 3.1.1		10	30	ns
tf (CMOS)	CMOS output falling time **(Note 2)**			10	30	ns

Notes 1: When the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.
2: The X_{OUT} pin is excluded.

Table 3.1.16 Switching characteristics (2)
(V_{CC} = 2.7 to 4.0 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t_{WH} (S_{CLK1})	Serial I/O1 clock output “H” pulse width	Fig. 3.1.1	$t_{C(SCLK1)}$/2–50			ns
t_{WL} (S_{CLK1})	Serial I/O1 clock output “L” pulse width		$t_{C(SCLK1)}$/2–50			ns
td (S_{CLK1}-TxD)	Serial I/O1 output delay time **(Note 1)**				350	ns
tv (S_{CLK1}-TxD)	Serial I/O1 output valid time **(Note 1)**		–30			ns
tr (S_{CLK1})	Serial I/O1 clock output rising time				50	ns
tf (S_{CLK1})	Serial I/O1 clock output falling time				50	ns
t_{WH} (S_{CLK2})	Serial I/O2 clock output “H” pulse width	Fig. 3.1.2	$t_{C(SCLK2)}$/2–240			ns
t_{WL} (S_{CLK2})	Serial I/O2 clock output “L” pulse width		$t_{C(SCLK2)}$/2–240			ns
td (S_{CLK2}-S_{OUT2})	Serial I/O2 output delay time				400	ns
tv (S_{CLK2}-S_{OUT2})	Serial I/O2 output valid time		0			ns
tf (S_{CLK2})	Serial I/O2 clock output falling time				50	ns
tr (CMOS)	CMOS output rising time **(Note 2)**	Fig. 3.1.1		20	50	ns
tf (CMOS)	CMOS output falling time **(Note 2)**			20	50	ns

Notes 1: When the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.
2: The X_{OUT} pin is excluded.

3.1.10 Switching characteristics for system bus interface

Table 3.1.17 Switching characteristics for system bus interface (1)
(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_a(R-D)	After read data output enable time			80	ns
t_v(R-D)	After read data output disable time	0		30	ns
t_{PLH}(R-OBF)	After read OBF_{00}, OBF_{01}, OBF_{10} output propagation time			150	ns

Table 3.1.18 Switching characteristics for system bus interface (2)
(V_{CC} =2.7 to 4.0 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_a(R-D)	After read data output enable time			130	ns
t_v(R-D)	After read data output disable time	0		85	ns
t_{PLH}(R-OBF)	After read OBF_{00}, OBF_{01}, OBF_{10} output propagation time			300	ns

APPENDIX

3.1 Electrical characteristics

3.1.11 Timing requirements in memory expansion mode and microprocessor mode

Table 3.1.19 Timing requirements in memory expansion mode and microprocessor mode
(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, in high-speed mode, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{su} ($\overline{ONW}$-ϕ)	$\overline{ONW}$ input setup time	–20			ns
t_h (ϕ-$\overline{ONW}$)	$\overline{ONW}$ input hold time	–20			ns
t_{su} (DB-ϕ)	Data bus setup time	50			ns
t_h (ϕ-DB)	Data bus hold time	0			ns
t_{su} ($\overline{ONW}$-$\overline{RD}$), t_{su} ($\overline{ONW}$-$\overline{WR}$)	$\overline{ONW}$ input setup time	–20			ns
t_h ($\overline{RD}$-$\overline{ONW}$), t_h ($\overline{WR}$-$\overline{ONW}$)	$\overline{ONW}$ input hold time	–20			ns
t_{su} (DB-$\overline{RD}$)	Data bus setup time	50			ns
t_h ($\overline{RD}$-DB)	Data bus hold time	0			ns

3.1.12 Switching characteristics in memory expansion mode and microprocessor mode

Table 3.1.20 Switching characteristics in memory expansion mode and microprocessor mode
(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, in high-speed mode, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_C(\phi)$	ϕ clock cycle time	Fig. 3.1.1		2$t_C(X_{IN})$		ns
$t_{WH}(\phi)$	ϕ clock “H” pulse width		$t_C(X_{IN})$–10			ns
$t_{WL}(\phi)$	ϕ clock “L” pulse width		$t_C(X_{IN})$–10			ns
$t_d(\phi$-AH)	AD_{15}–AD_8 delay time			16	35	ns
$t_d(\phi$-AL)	AD_7–AD_0 delay time			20	40	ns
$t_v(\phi$-AH)	AD_{15}–AD_8 valid time		2	5		ns
$t_v(\phi$-AL)	AD_7–AD_0 valid time		2	5		ns
$t_d(\phi$-SYNC)	SYNC delay time			16		ns
$t_v(\phi$-SYNC)	SYNC valid time			5		ns
$t_d(\phi$-DB)	Data bus delay time			15	30	ns
$t_v(\phi$-DB)	Data bus valid time		10			ns
$t_{WL}(\overline{RD})$, $t_{WL}(\overline{WR})$	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width		$t_C(X_{IN})$–10			ns
	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width (When one-wait is valid)		3$t_C(X_{IN})$–10			ns
t_d(AH-$\overline{RD}$), t_d(AH-$\overline{WR}$)	AD_{15}–AD_8 delay time		$t_C(X_{IN})$–35	$t_C(X_{IN})$–16		ns
t_d(AL-$\overline{RD}$), t_d(AL-$\overline{WR}$)	AD_7–AD_0 delay time		$t_C(X_{IN})$–40	$t_C(X_{IN})$–20		ns
t_v($\overline{RD}$-AH), t_v($\overline{WR}$-AH)	AD_{15}–AD_8 valid time		2	5		ns
t_v($\overline{RD}$-AL), t_v($\overline{WR}$-AL)	AD_7–AD_0 valid time		2	5		ns
t_d($\overline{WR}$-DB)	Data bus delay time			15	30	ns
t_v($\overline{WR}$-DB)	Data bus valid time		10			ns
t_d($\overline{RESET}$-$\overline{RESET_{OUT}}$)	$\overline{RESET_{OUT}}$ output delay time				200	ns
$t_v(\phi$-$\overline{RESET_{OUT}}$)	$\overline{RESET_{OUT}}$ output valid time **(Note)**		0		100	ns

Note: The $\overline{RESET_{OUT}}$ output goes “H” in synchronized with the rise of the ϕ clock that is anywhere between a few cycles and 10-several cycles after $\overline{RESET}$ input goes “H”.

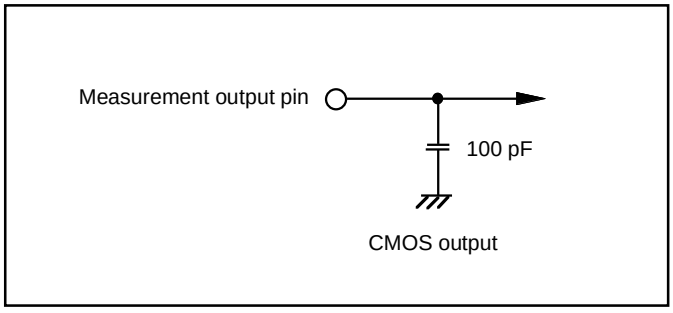


Fig. 3.1.1 Circuit for measuring output switching characteristics (1)

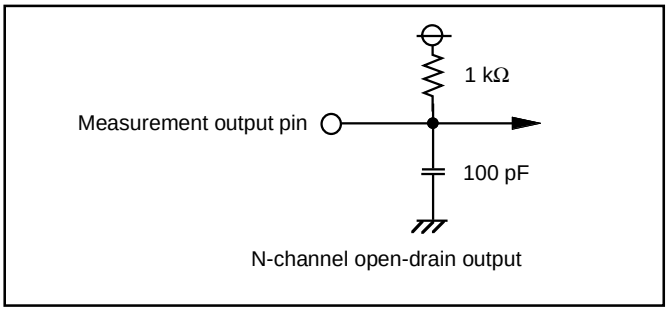


Fig. 3.1.2 Circuit for measuring output switching characteristics (2)

Timing diagram in single-chip mode

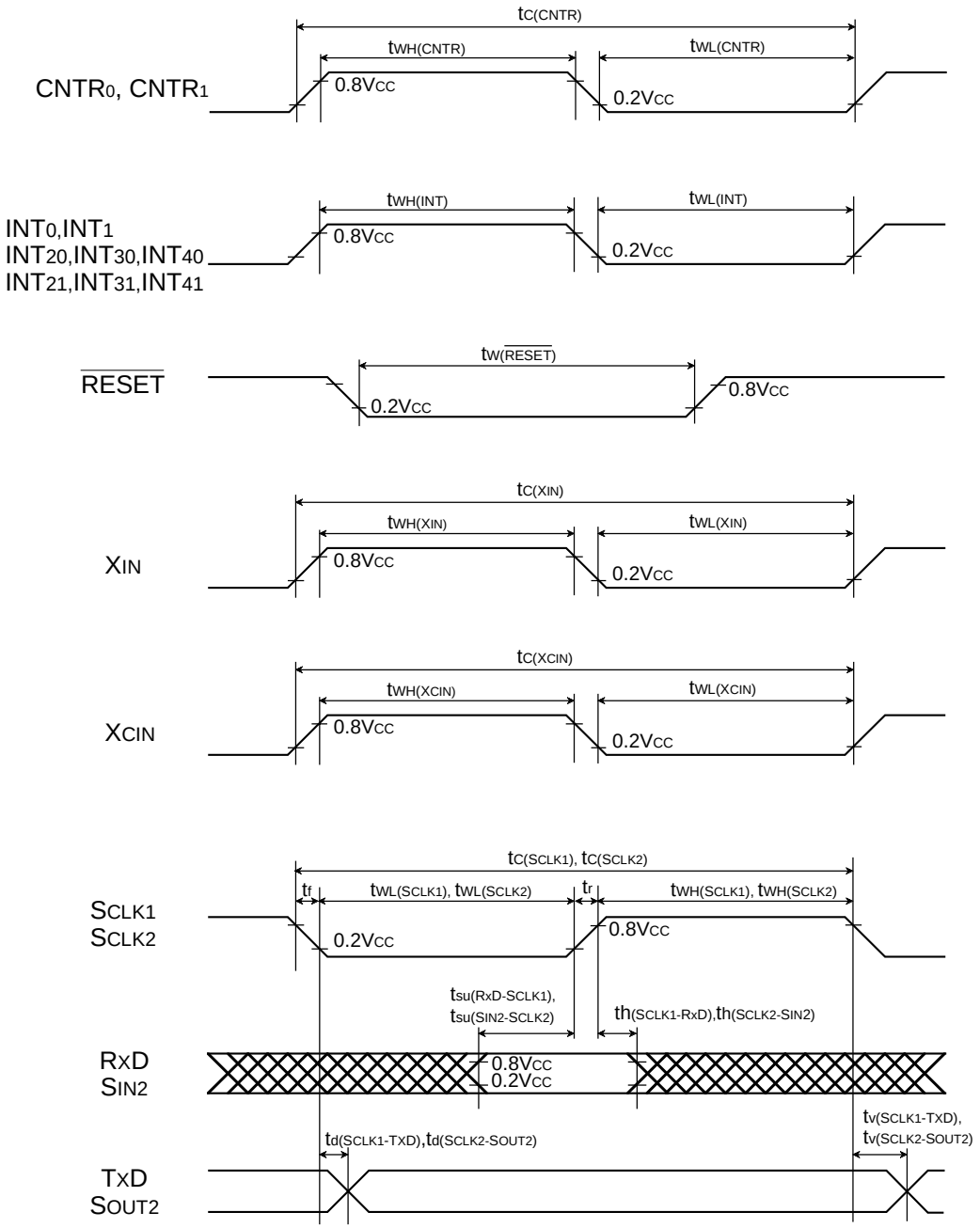
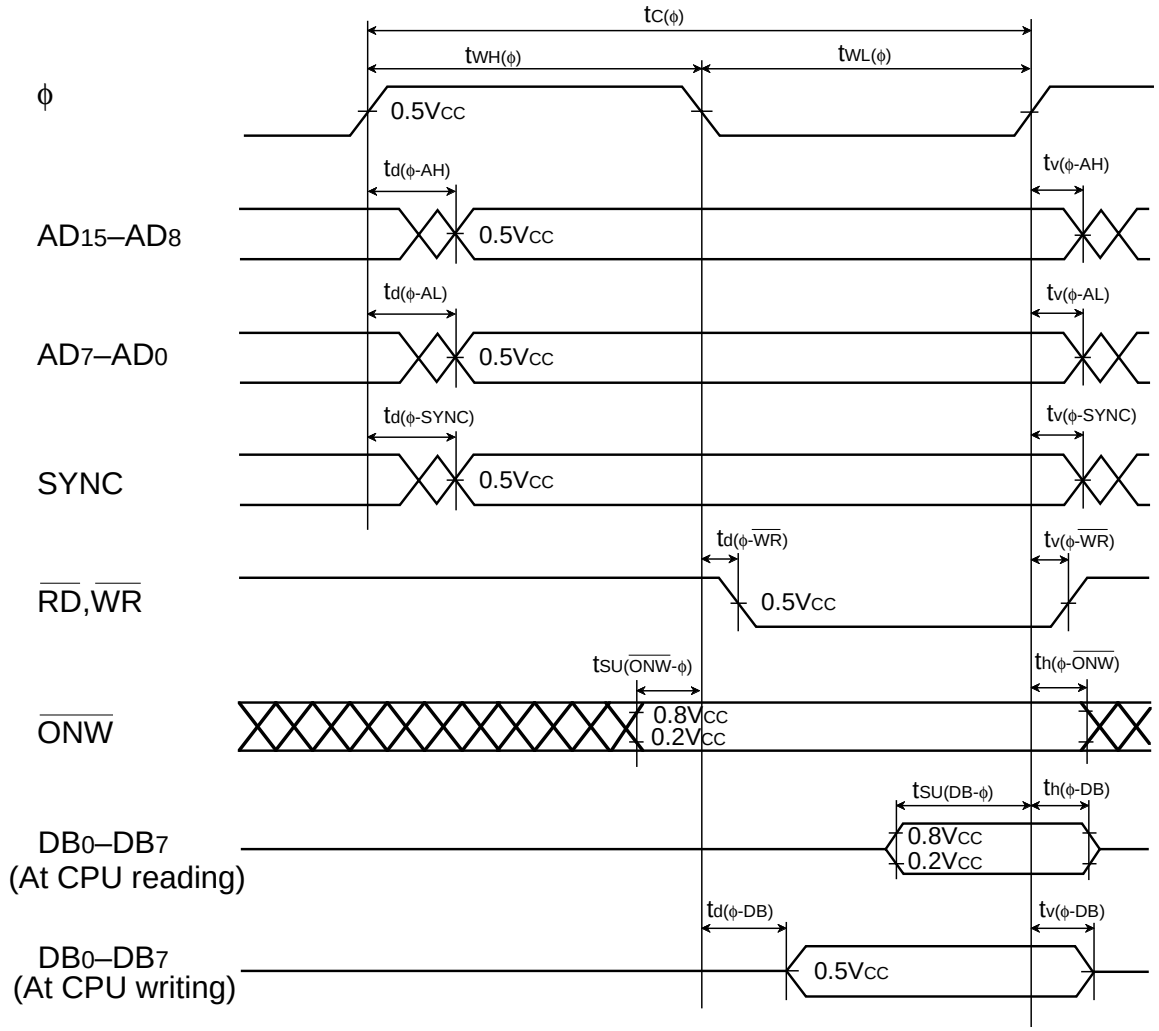


Fig. 3.1.3 Timing diagram (1) (in single-chip mode)

Timing diagram in memory expansion mode and microprocessor mode (1)



Timing diagram in microprocessor mode

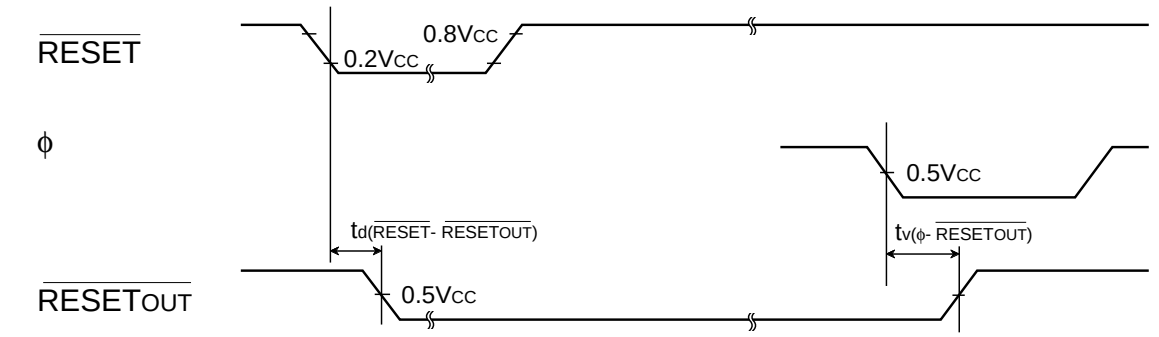


Fig. 3.1.4 Timing diagram (2) (in memory expansion mode and microprocessor mode)

Timing diagram in memory expansion mode and microprocessor mode (2)

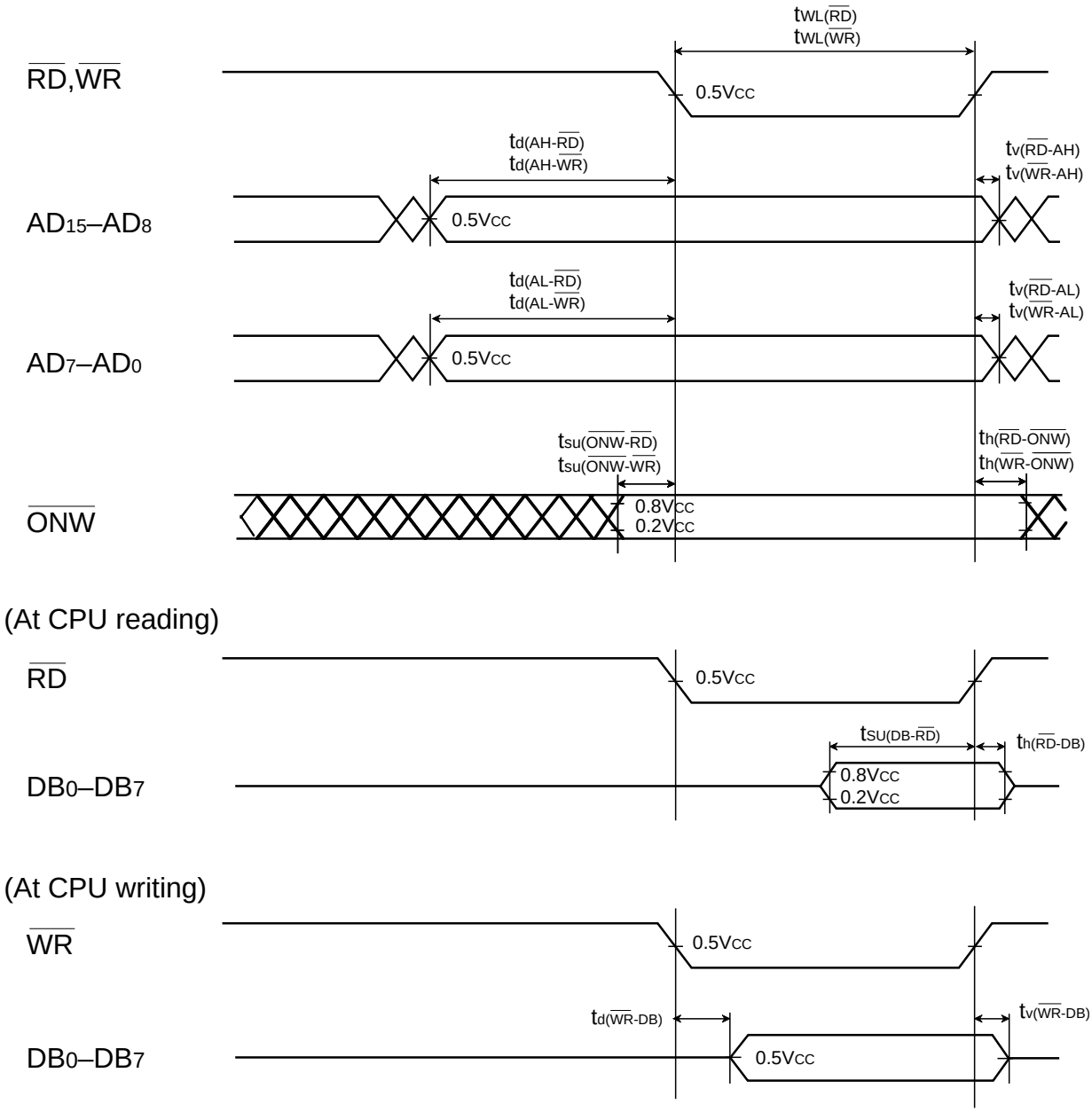
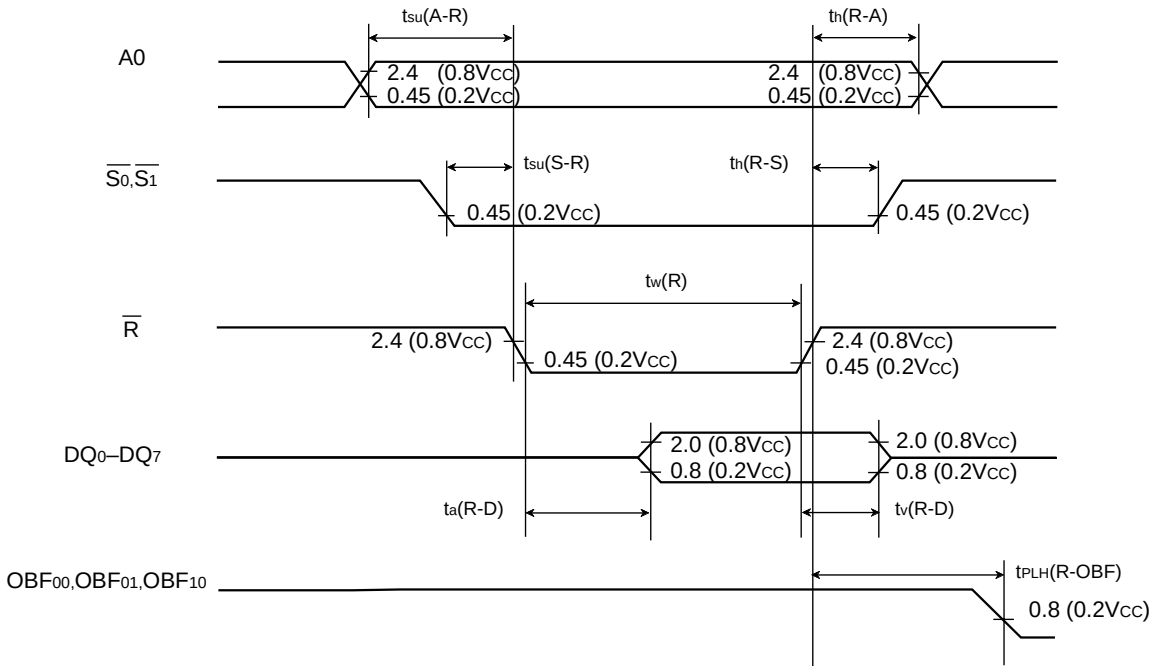


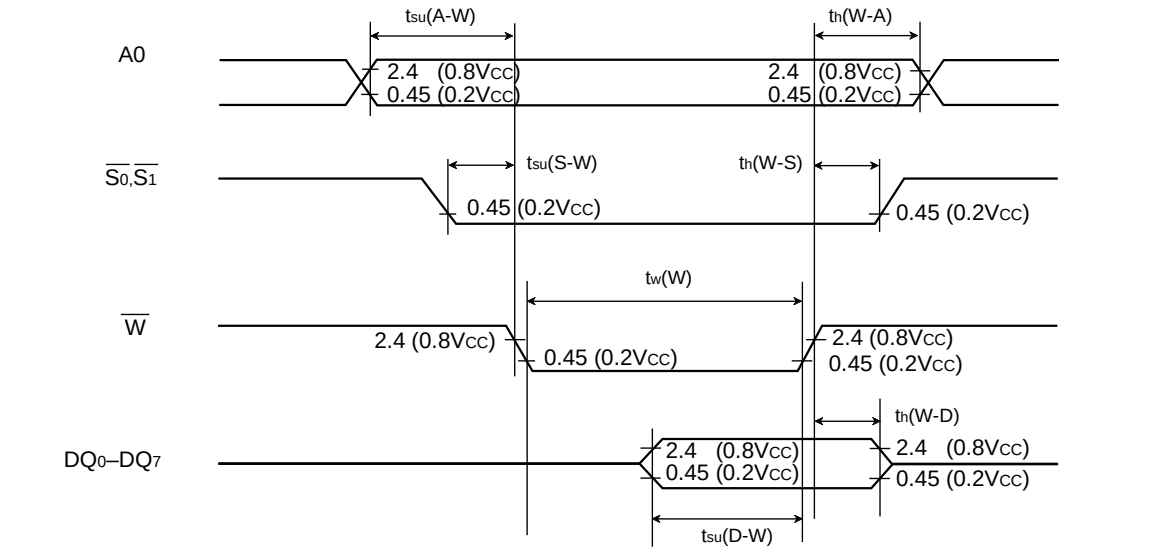
Fig. 3.1.5 Timing diagram (3) (in memory expansion mode and microprocessor mode)

System bus interface timing diagram

Read operation



Write operation



Outside of parenthesis : TTL I/O
Inside of parenthesis : CMOS I/O

Fig. 3.1.6 Timing diagram (4) (system bus interface)

3.1.13 Multi-master I²C-BUS bus line characteristics

Table 3.1.21 Multi-master I²C-BUS bus line characteristics

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

Note: C_b = total capacitance of 1 bus line

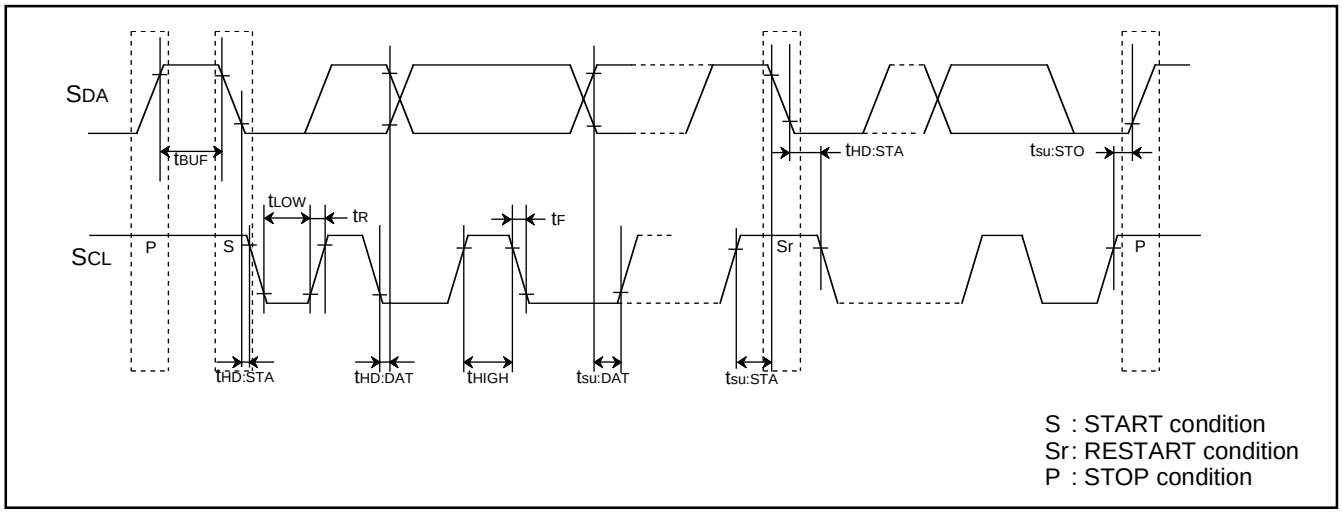


Fig. 3.1.7 Timing diagram of multi-master I²C-BUS

3.2 Standard characteristics

3.2.1 Power source current characteristic examples

Figure 3.2.1, Figure 3.2.2, Figure 3.2.3, Figure 3.2.4, Figure 3.2.5, Figure 3.2.6 and Figure 3.2.7 show power source current characteristic examples.

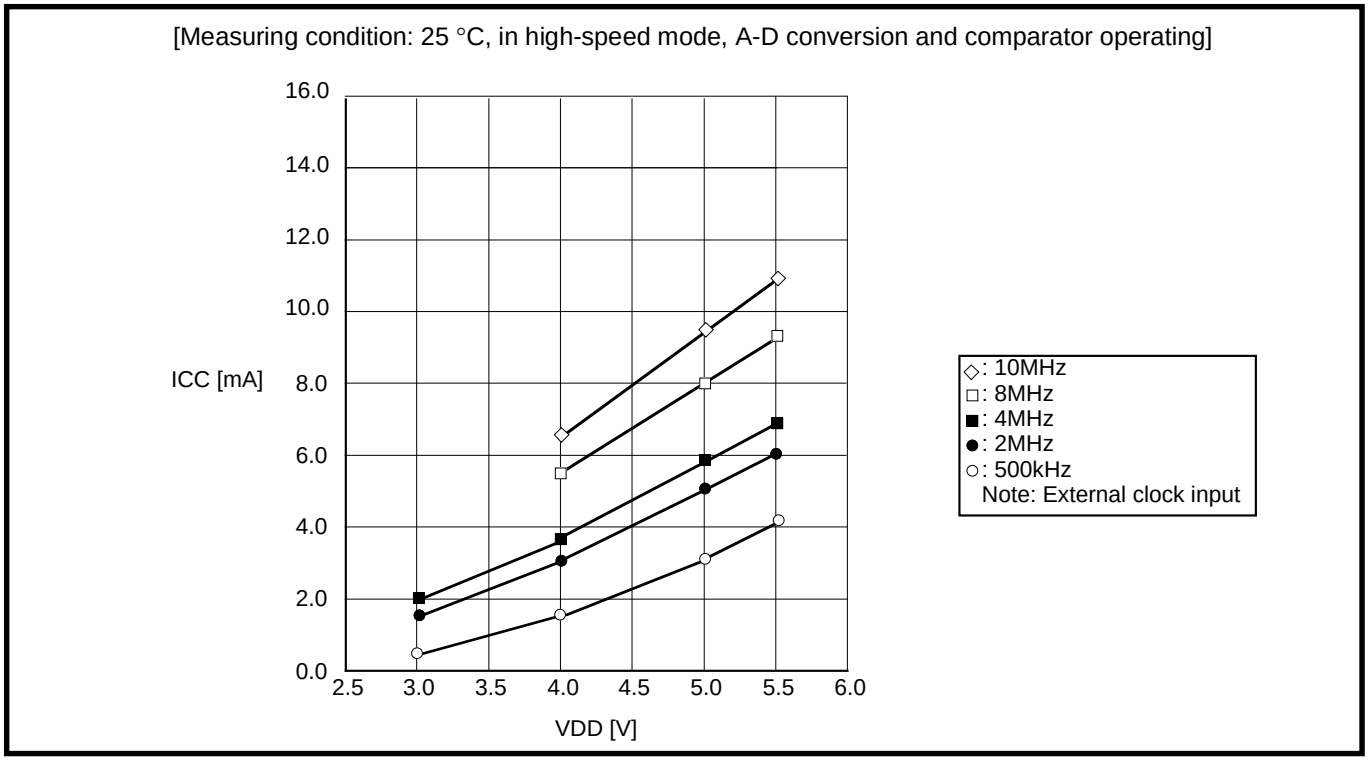


Fig. 3.2.1 Power source current characteristic examples (in high-speed mode, A-D conversion and comparator operating)

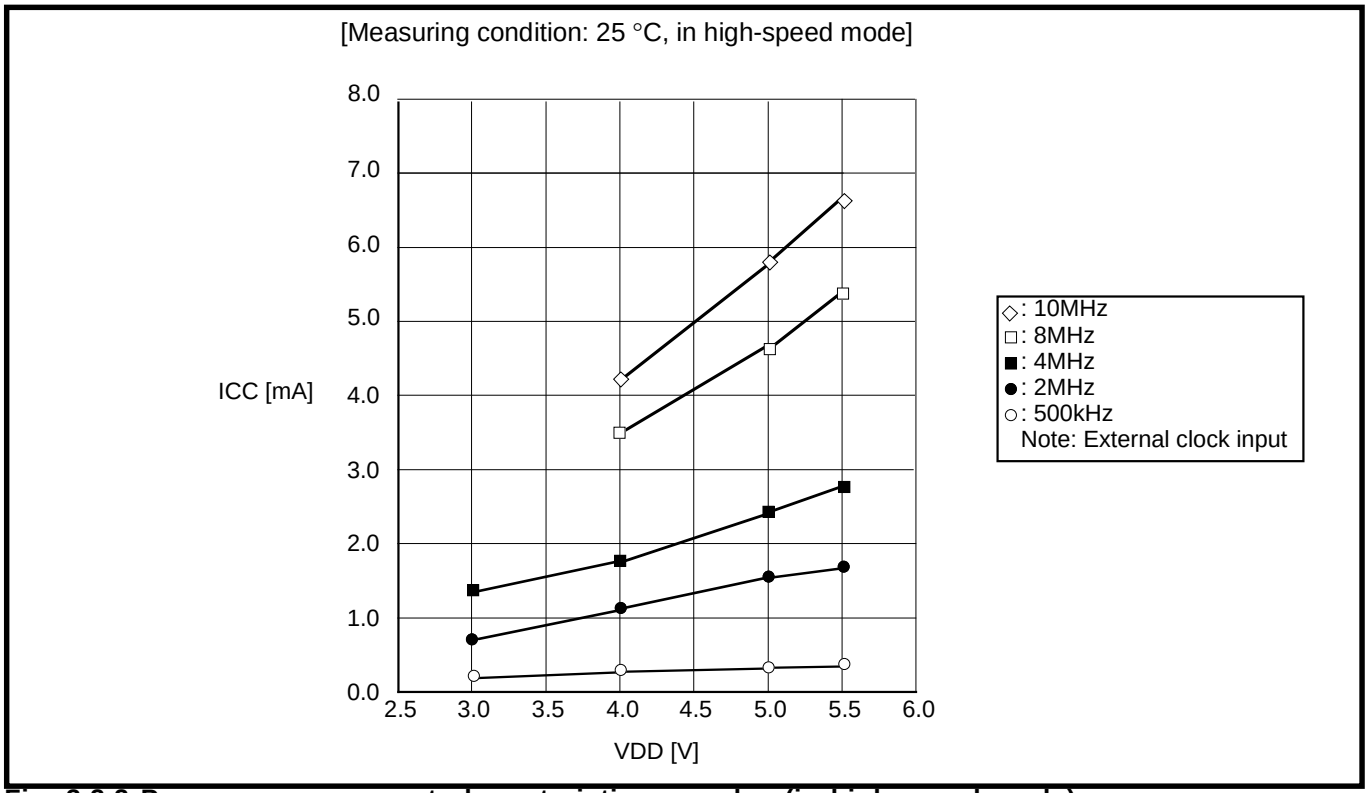


Fig. 3.2.2 Power source current characteristic examples (in high-speed mode)

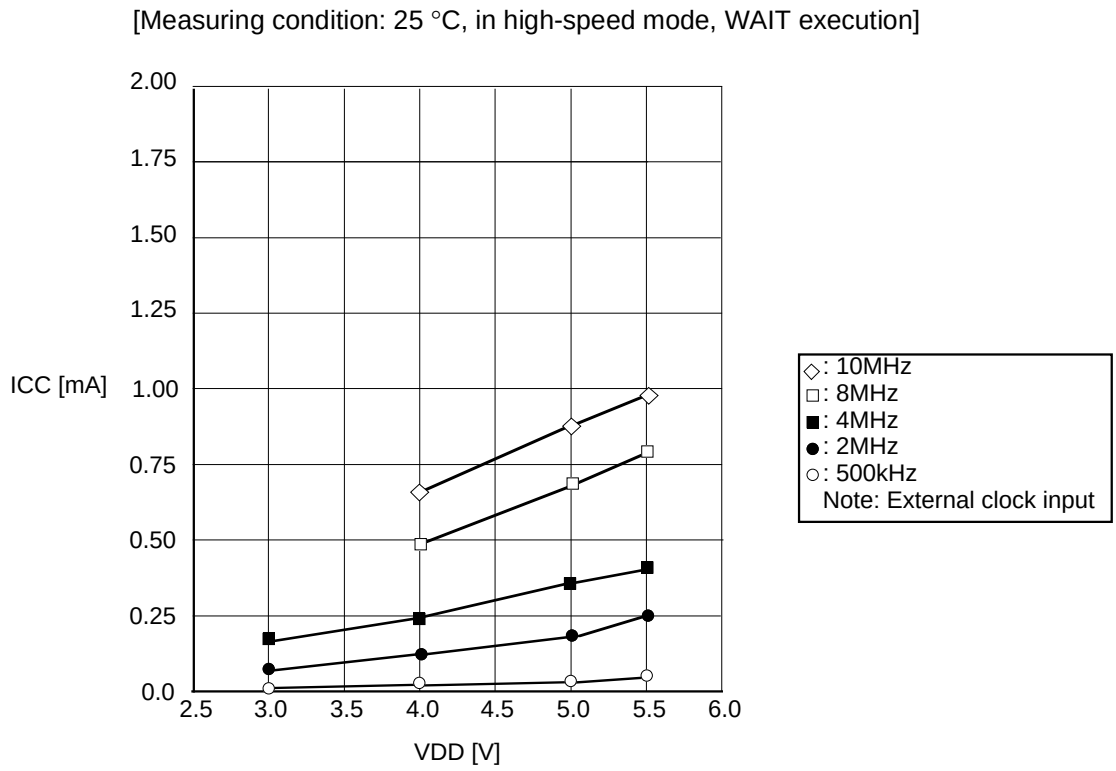


Fig. 3.2.3 Power source current characteristic examples (in high-speed mode, WAIT execution)

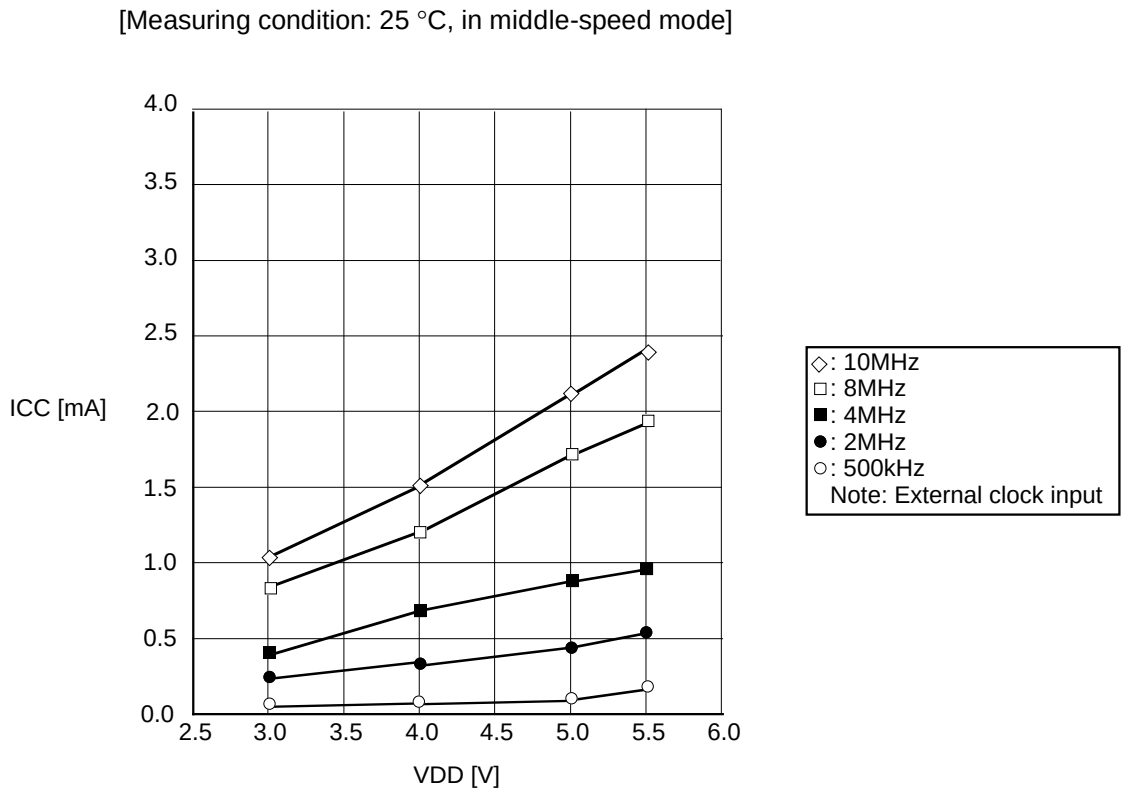


Fig. 3.2.4 Power source current characteristic examples (in middle-speed mode)

3.2 Standard characteristics

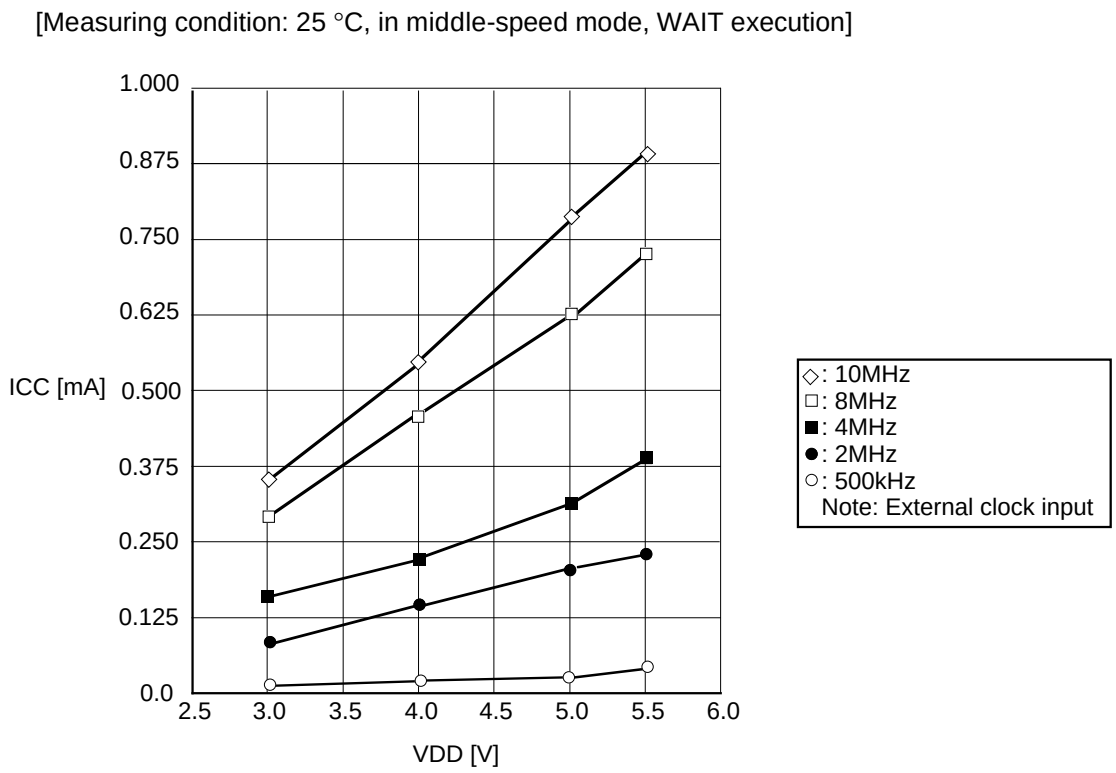


Fig. 3.2.5 Power source current characteristic examples (in middle-speed mode, WAIT execution)

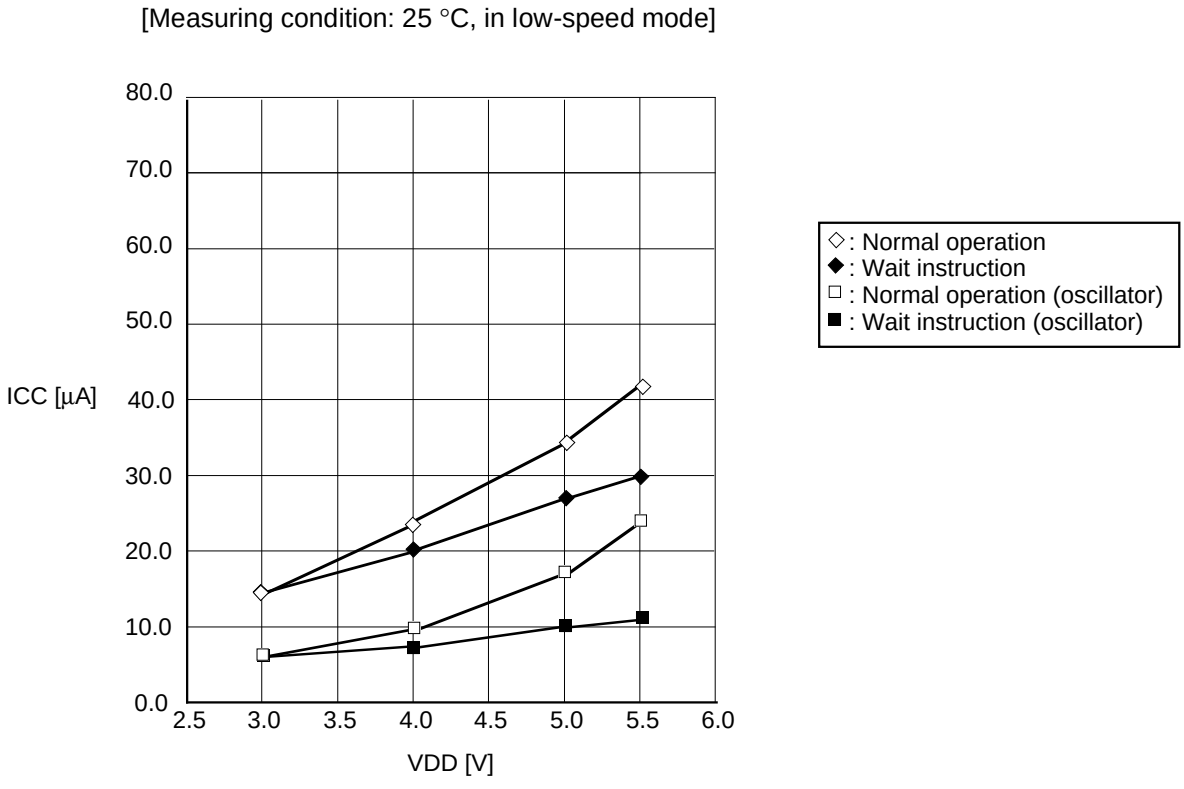


Fig. 3.2.6 Power source current characteristic examples (in low-speed mode)

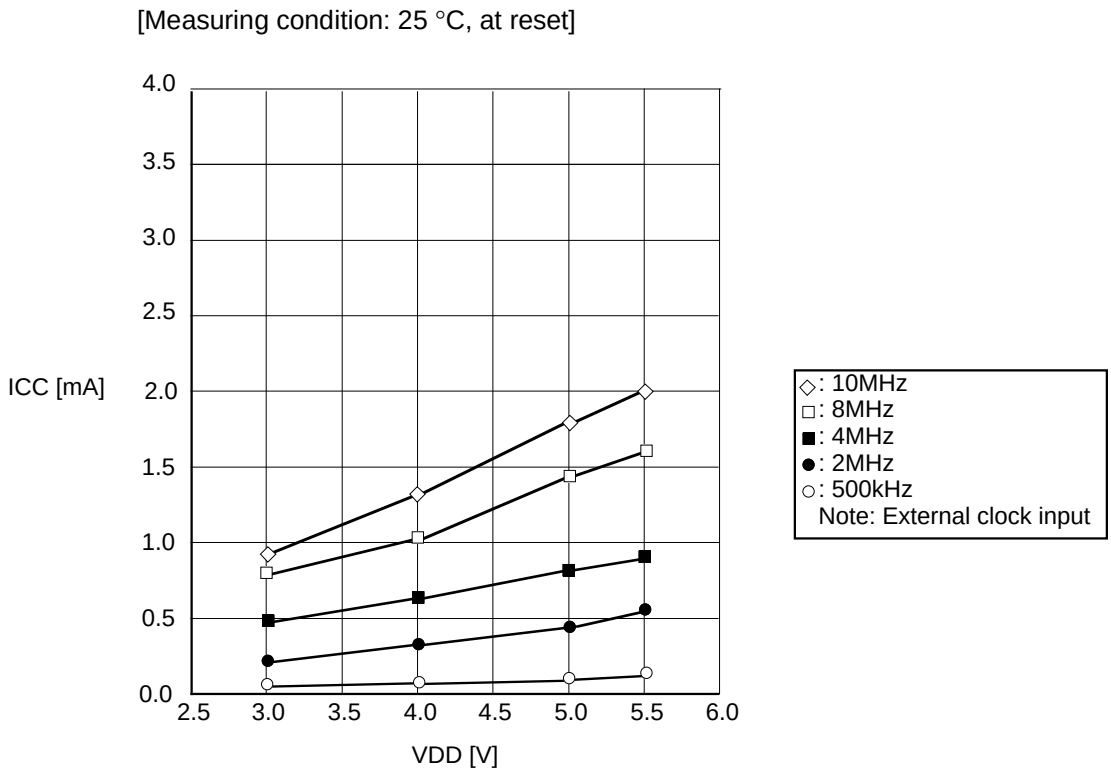


Fig. 3.2.7 Power source current characteristic examples (at reset)

APPENDIX

3.2 Standard characteristics

3.2.2 Port standard characteristic examples

Figures 3.2.8, Figures 3.2.9, Figures 3.2.10, Figures 3.2.11, Figure 3.2.12, and Figure 3.2.13 show port standard characteristic examples.

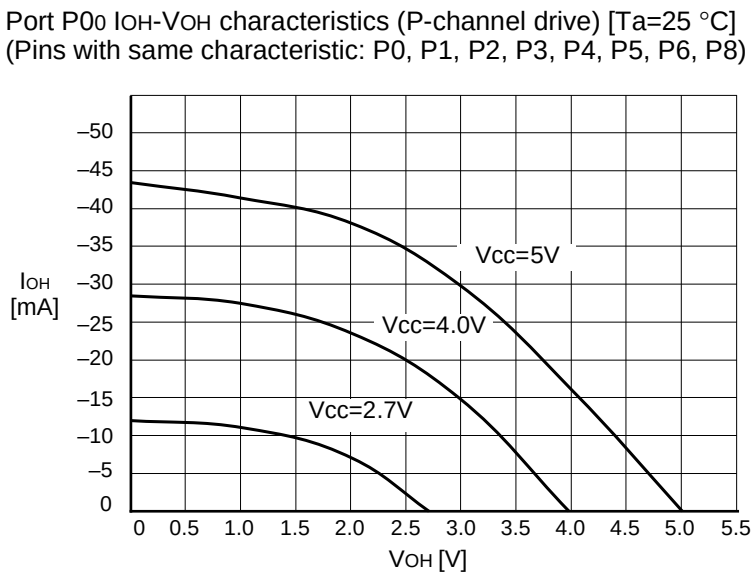


Fig. 3.2.8 Standard characteristic examples of CMOS output port at P-channel drive (Ta=25 °C)

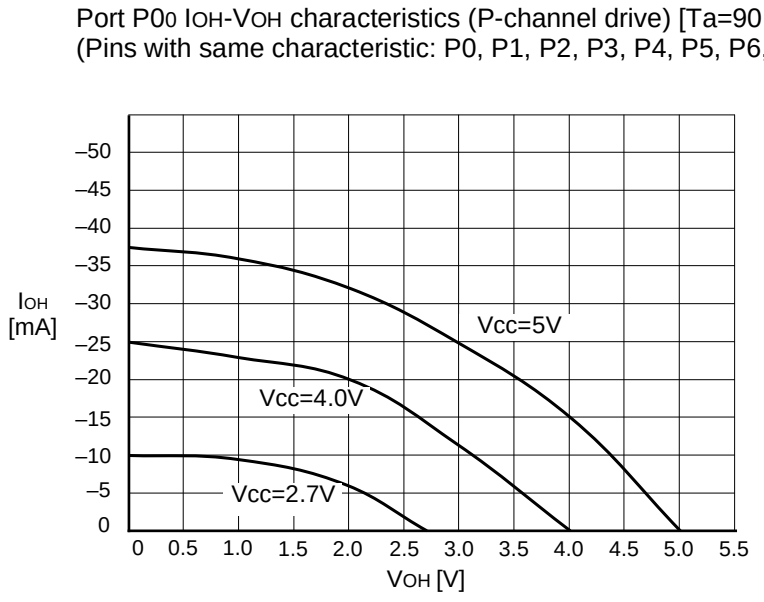


Fig. 3.2.9 Standard characteristic examples of CMOS output port at P-channel drive (Ta=90 °C)

Port $P0_0$ I_{OL}-V_{OL} characteristics (N-channel drive) [Ta=25 °C]
(Pins with same characteristic: P0, P1, $P2_0$–$P2_3$, P3, P4, P5,
P6, P7, P8, and $P2_4$–$P2_7$ except at single-chip mode)

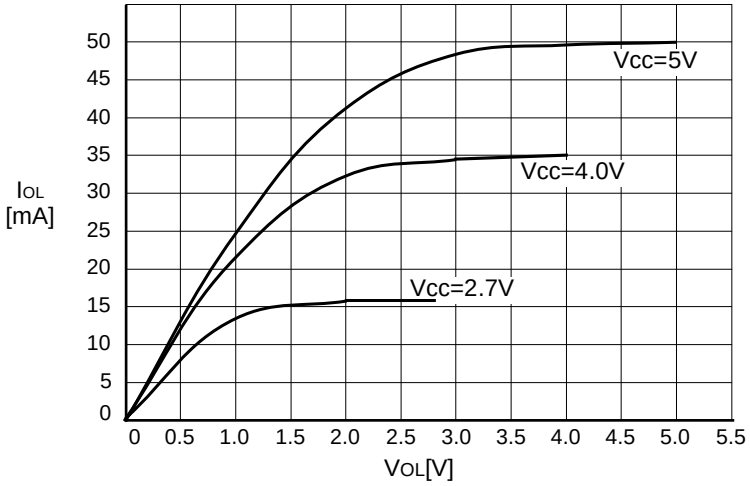


Fig. 3.2.10 Standard characteristic examples of CMOS output port at N-channel drive (Ta=25 °C)

Port $P0_0$ I_{OL}-V_{OL} characteristics (N-channel drive) [Ta=90 °C]
(Pins with same characteristic: P0, P1, $P2_0$–$P2_3$, P3, P4, P5,
P6, P7, P8, and $P2_4$–$P2_7$ except at single-chip mode)

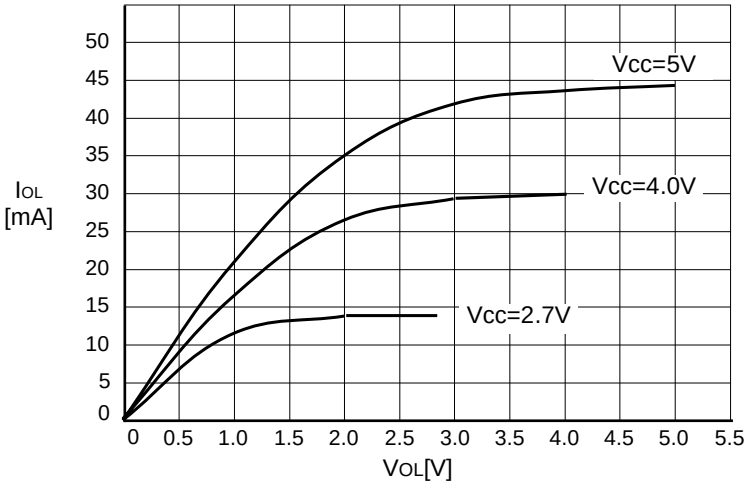


Fig. 3.2.11 Standard characteristic examples of CMOS output port at N-channel drive (Ta=90 °C)

APPENDIX

3.2 Standard characteristics

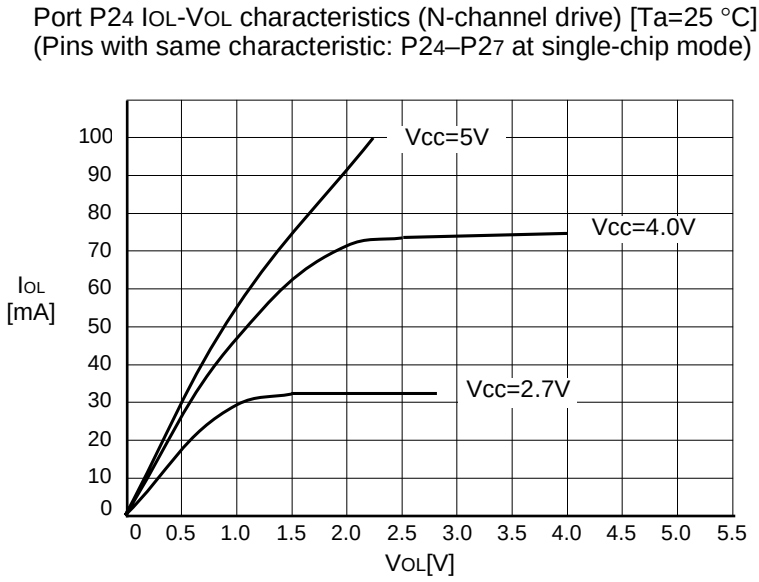


Fig. 3.2.12 Standard characteristic examples of CMOS large current output port at N-channel drive (Ta=25 °C)

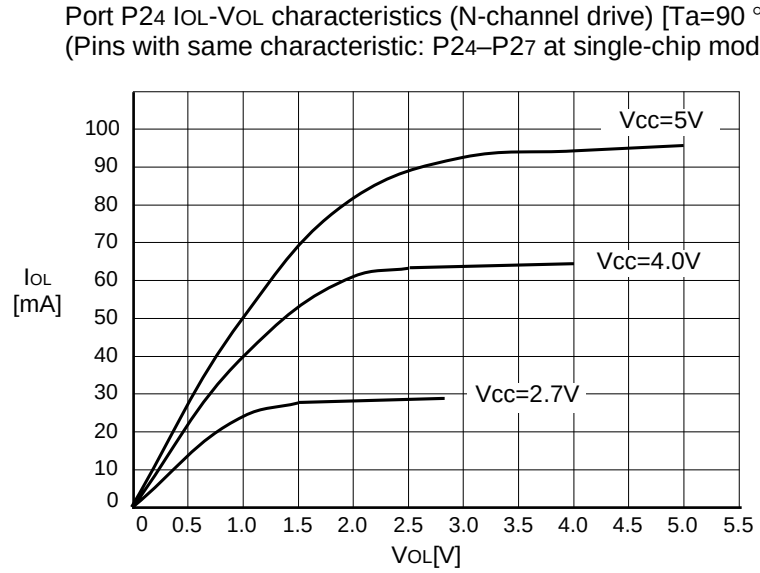


Fig. 3.2.13 Standard characteristic examples of CMOS large current output port at N-channel drive (Ta=90 °C)

3.2.3 Input port standard characteristic examples

Figures 3.2.14 and Figure 3.2.15 show port standard characteristic examples.

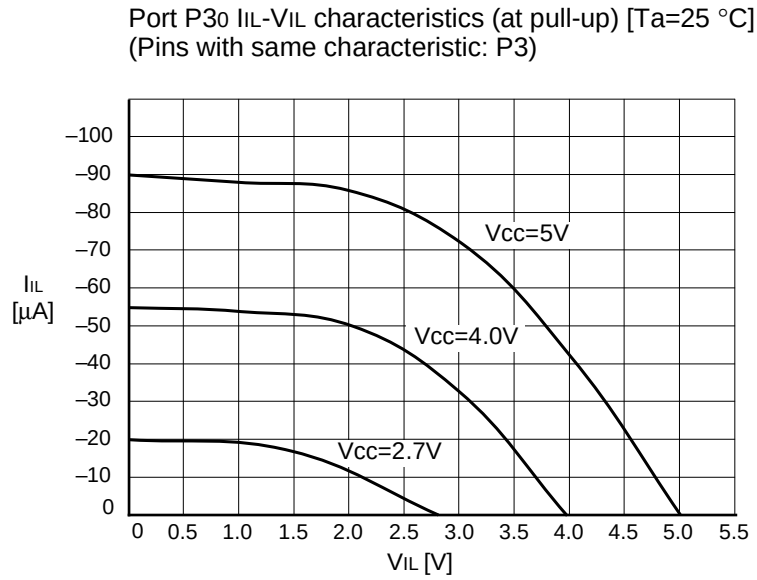


Fig. 3.2.14 Standard characteristic examples of CMOS input port at pull-up (Ta=25 °C)

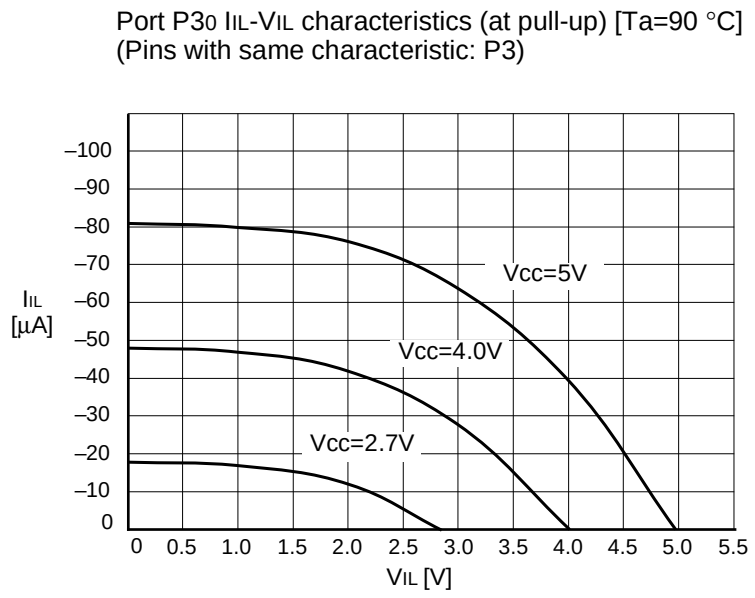


Fig. 3.2.15 Standard characteristic examples of CMOS input port at pull-up (Ta=90 °C)

APPENDIX

3.2 Standard characteristics

3.2.4 A-D conversion standard characteristics

Figure 3.2.16 shows the A-D conversion standard characteristics.

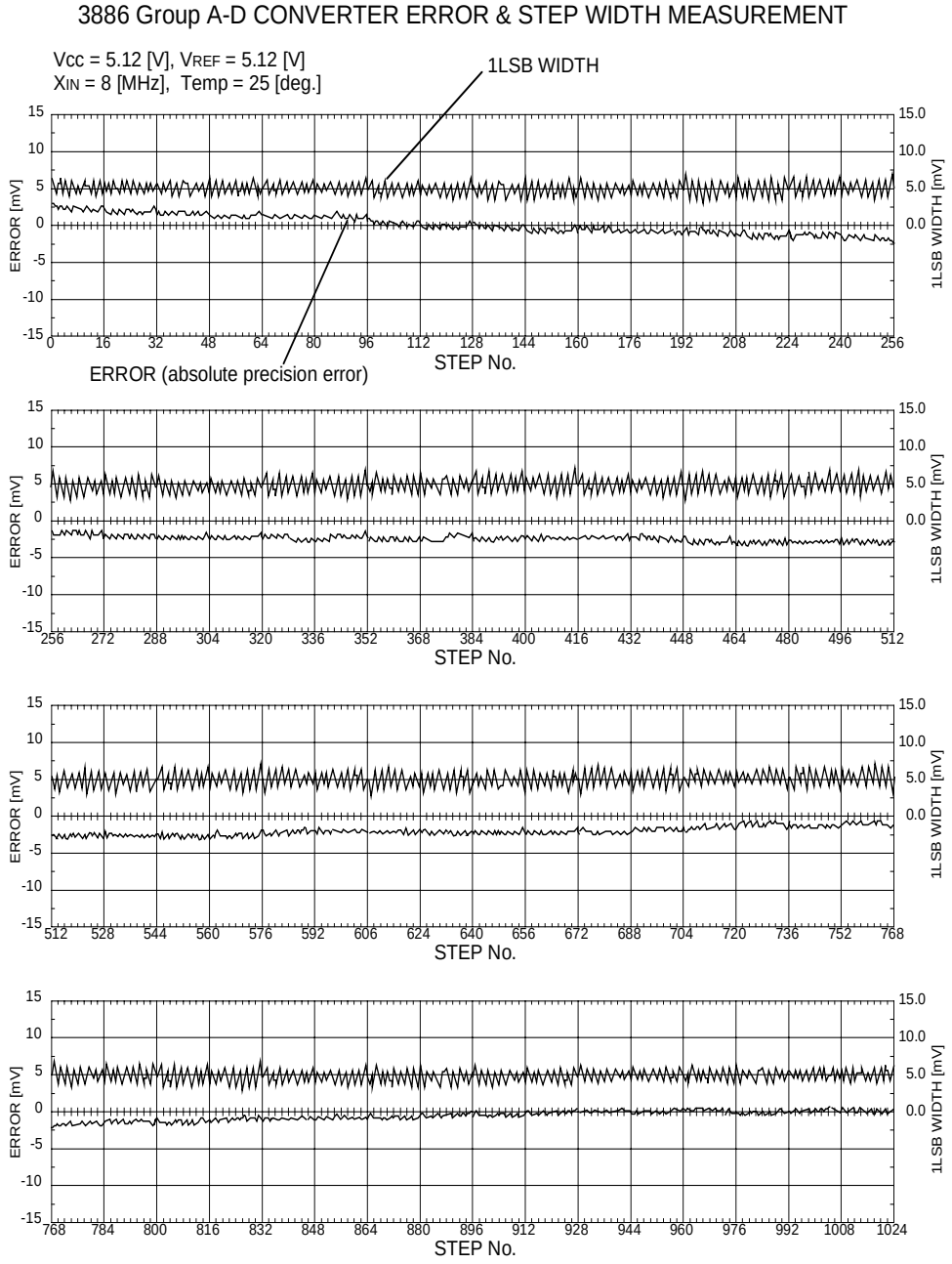


Fig. 3.2.16 A-D conversion standard characteristics

3.2.5 D-A conversion standard characteristics

Figure 3.2.17 shows the D-A conversion standard characteristics.

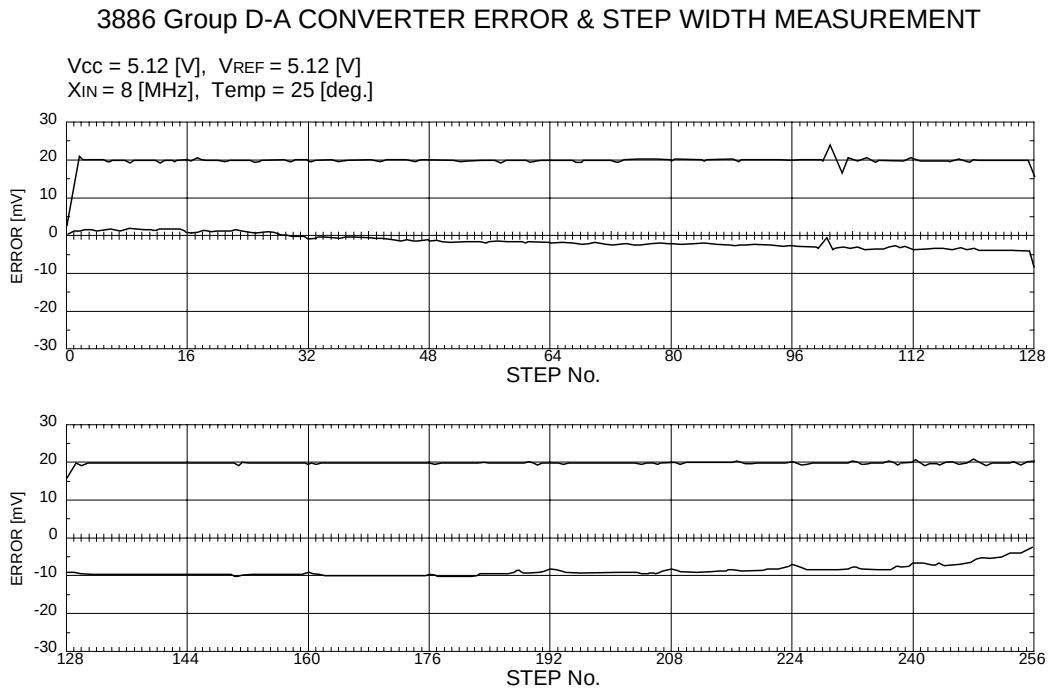


Fig. 3.2.17 D-A conversion standard characteristics

3.3 Notes on use

3.3.1 Notes on input and output pins

(1) Notes in stand-by state

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

Pull-up (connect the port to 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 or pull-down 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**

In I/O ports of the N-channel open-drain, in spite of setting as an output port with its direction register, when the content of the port latch is "1", the transistor becomes the OFF state, which causes the ports to be the high-impedance state. Note that the level becomes "undefined" depending on external circuits.

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

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

(2) Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*2, the value of the unspecified bit may be changed.

■ **Reason**

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

- As for a bit which is set for an input port :
  The pin state is read in the CPU, and is written to this bit after bit managing.
- As for a bit which is set for an output port :
  The bit value of the port latch is read in the CPU, and is written to this bit after bit managing.

Note the following :

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

*2 bit managing instructions : **SEB**, and **CLB** instructions

3.3.2 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. With regard to ports which can select the built-in pull-up resistor, the built-in pull-up resistor can be used.
As for pins whose potential affects to operation modes such as the CNVSS pin 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. With regard to ports which can select the built-in pull-up resistor, the built-in pull-up resistor can be used.
  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.

④ **The AVss pin when not using the A-D/D-A converter :**
- When not using the A-D/D-A converter, handle a power source pin for the A-D/D-A converter, AVss pin as follows:
- AVss: Connect to the Vss pin

(2) Termination remarks

① **Input ports and I/O ports :**
Do not open in the input mode.

■ **Reason**
- The power source current may increase depending on the first-stage circuit.
- An effect due to noise may be easily produced as compared with proper termination ② and ③ shown on the above.

② **I/O ports :**
When setting for the input mode, do not connect to VCC or VSS directly.

■ **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between a port and VCC (or VSS).

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

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

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

3.3.3 Notes on interrupts

(1) Switching external interrupt detection edge
When switching the external interrupt detection edge, switch it in the following sequence.

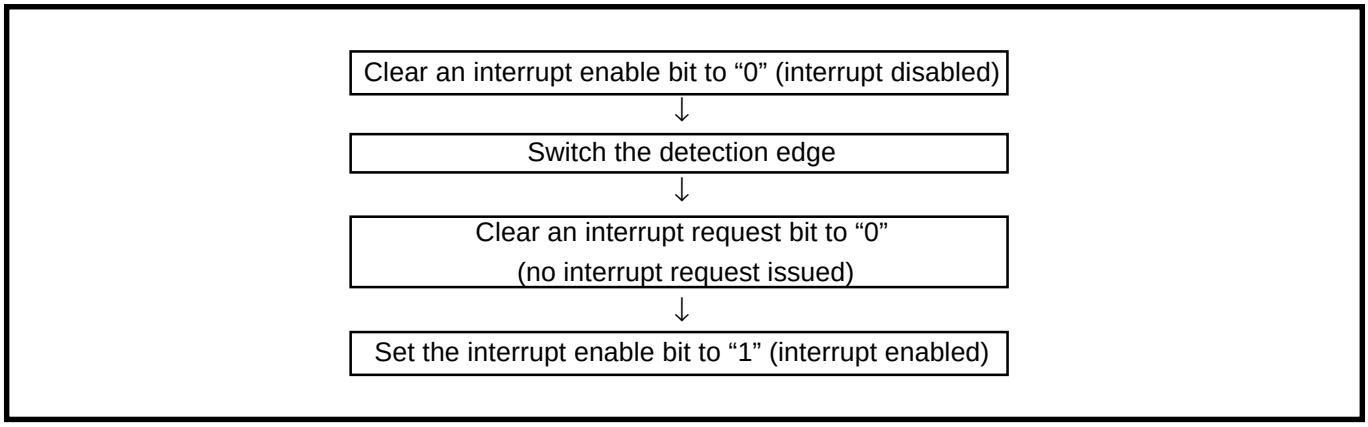


Fig. 3.3.1 Sequence of switching the detection edge

■ **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.

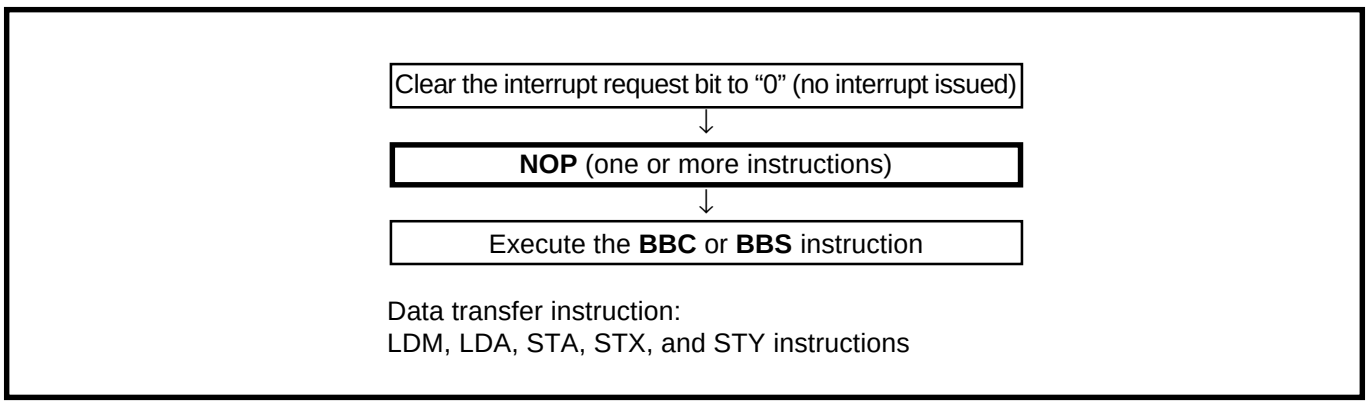


Fig. 3.3.2 Sequence of check of interrupt request bit

■ **Reason**

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

(3) Change of relevant register setting

When the setting of the following register or bit is changed, the interrupt request bit may be set to “1”.

•Interrupt edge selection register (address $3A_{16}$)
•Interrupt source selection register (address 39_{16})
•INT2, INT3, INT4 interrupt switch bit of port control register 2 (bit 4 of address $2F_{16}$)

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

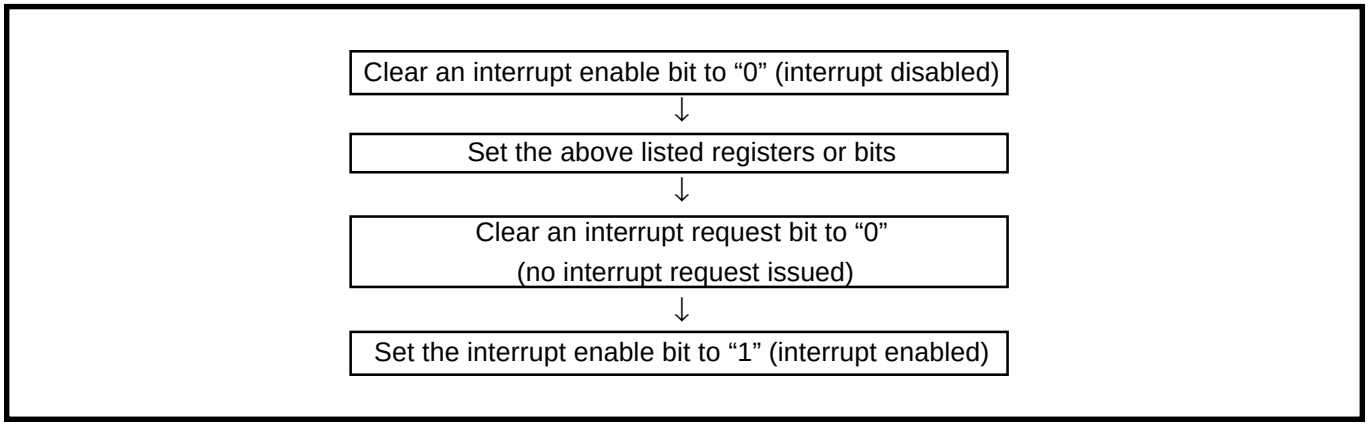


Fig. 3.3.3 Sequence of changing relevant register

3.3.4 Notes on timer

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).
- When switching the count source by the timer Y count source selection bit, the value of timer count is altered in unconsiderable amount owing to generating of a thin pulses in the count input signals. Therefore, select the timer count source before set the value to the timer.

3.3.5 Notes on serial I/O

(1) Notes when selecting clock synchronous serial I/O (Serial I/O1)

① **Stop of transmission operation**
Clear the serial I/O1 enable bit and the transmit enable bit to “0” (Serial I/O1 and transmit disabled).

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

② **Stop of receive operation**
Clear the receive enable bit to “0” (receive disabled), or clear the serial I/O1 enable bit to “0” (Serial I/O1 disabled).

③ **Stop of transmit/receive operation**
Clear both the transmit enable bit and receive enable bit to “0” (transmit and receive disabled). (when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

■ **Reason**
In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to “0” (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O1 enable bit to “0” (Serial I/O1 disabled) (refer to (1) ①).

(2) Notes when selecting clock asynchronous serial I/O (Serial I/O1)

① **Stop of transmission operation**
Clear the transmit enable bit to "0" (transmit disabled).

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

② **Stop of receive operation**
Clear the receive enable bit to "0" (receive disabled).

③ **Stop of transmit/receive operation**
Only transmission operation is stopped.
Clear the transmit enable bit to "0" (transmit disabled).

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

Only receive operation is stopped.
Clear the receive enable bit to "0" (receive disabled).

(3) $\overline{\text{SRDY1}}$ output of reception side (Serial I/O1)
When signals are output from the $\overline{\text{SRDY1}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDY1}}$ output enable bit, and the transmit enable bit to "1" (transmit enabled).

(4) Setting serial I/O1 control register again (Serial I/O1)
Set the serial I/O1 control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to "0."

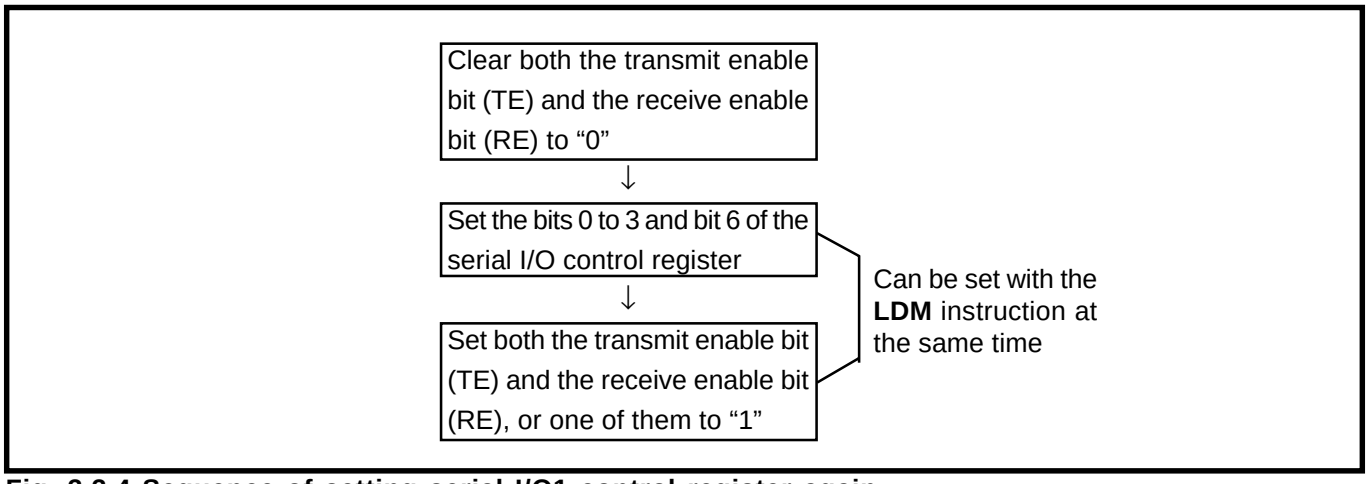


Fig. 3.3.4 Sequence of setting serial I/O1 control register again

3.3 Notes on use

(5) Data transmission control with referring to transmit shift register completion flag (Serial I/O1)
The transmit shift register completion flag changes from “1” to “0” with a delay of 0.5 to 1.5 shift clocks after writing the data to the transmit buffer register. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

(6) Transmission control when external clock is selected (Serial I/O1)
When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLK input level. Also, write the transmit data to the transmit buffer register (serial I/O shift register) at “H” of the SCLK input level.

(7) Transmit interrupt request when transmit enable bit is set (Serial I/O1)
When the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as shown in the following sequence.
① Set the interrupt enable bit to “0” (disabled) with CLB instruction.
② Prepare serial I/O for transmission/reception.
③ Set the interrupt request bit to “0” with CLB instruction after 1 or more instruction has been executed.
④ Set the interrupt enable bit to “1” (enabled).

■ Reason
When the transmission enable bit is set to “1”, the transmit buffer empty flag and transmit shift register completion flag are set to “1”. The interrupt request is generated and the transmission interrupt bit is set regardless of which of the two timings listed below is selected as the timing for the transmission interrupt to be generated.
• Transmit buffer empty flag is set to “1”
• Transmit shift register completion flag is set to “1”

(8) Transmit data writing (Serial I/O2)
In the clock synchronous serial I/O, when selecting an external clock as synchronous clock, write the transmit data to the serial I/O2 register (serial I/O shift register) at “H” of the transfer clock input level.

3.3.6 Notes on multi-master I²C-BUS interface

(1) Read-modify-write instruction

Precautions for read-modify-write instructions, such as **SEB** and **CLB**, when used for any of the registers of the multi-master I²C-BUS interface, are described below.

① **I²C data shift register (S0: address 0012_{16})**
When executing the read-modify-write instruction for this register during transfer, data may become an unexpected value.

② **I²C address register (S0D: address 0013_{16})**
When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become an unexpected value.

■ **Reason**
Because hardware changes the read/$\overline{\text{write}}$ bit (RWB) at detecting the STOP condition.

③ **I²C status register (S1: address 0014_{16})**
Do not execute the read-modify-write instruction for this register because all bits of this register are changed by hardware.

④ **I²C control register (S1D: address 0015_{16})**
When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become an unexpected value.

■ **Reason**
Because hardware changes the bit counter (BC0 to BC2).

⑤ **I²C clock control register (S2: address 0016_{16})**
The read-modify-write instruction can be executed for this register.

⑥ **I²C START/STOP condition control register (S2D: address 0017_{16})**
The read-modify-write instruction can be executed for this register.

(2) Procedure for generating START condition using multi-master

① Procedure example (The necessary conditions for the procedure are described in Items ② to ⑤ below).

```
        LDA #SLADR          (Take out slave address value)
        SEI                 (Disable interrupt)
        BBS 5, S1, BUSBUSY  (BB flag confirmation and branch process)
BUSFREE:
        STA S0              (Write slave address value)
        LDM #$F0, S1        (Trigger START condition generation)
        CLI                 (Enable interrupt)
         :
         :
BUSBUSY:
        CLI                 (Enable interrupt)
         :
         :
```

② Use “Branch on Bit Set” of “BBS 5, S1, –” for the BB flag confirmation and branch process.

③ Use “STA”, “STX” or “STY” of the zero page addressing instruction for writing the slave address value to the I²C data shift register (S0: address 0012_{16}).

④ Execute the branch instruction of above ② and the store instruction of above ③ continuously shown the above procedure example.

⑤ Disable interrupts during the following three process steps:
- BB flag confirmation
- Write slave address value
- Trigger START condition generation

When the BB flag is in bus busy state, enable interrupts immediately.

(3) Procedure for generating RESTART condition

This procedure cannot be applied to M38867M8A and M38867E8A when the external memory is used and the bus cycle is extended by $\overline{\text{ONW}}$ function.

① Procedure example (The necessary conditions for the procedure are described in Items ② to ④ below). Execute the following procedure when the PIN bit is "0".

```
LDM #$00, S1        (Select slave receive mode)
LDA #SLADR          (Take out slave address value)
SEI                 (Disable interrupt)
STA S0              (Write slave address value)
LDM #$F0, S1        (Trigger RESTART condition generation)
CLI                 (Enable interrupt)
 :
 :
```

② Select the slave receive mode when the PIN bit is "0". Do not write "1" to the PIN bit. Neither "0" nor "1" is specified as input to the BB bit. The TRX bit becomes "0" and the SDA pin is released.

③ The SCL pin is released by writing the slave address value to the I^2C data shift register.

④ Disable interrupts during the following two process steps:

- Write slave address value
- Trigger RESTART condition generation

(4) Writing to I^2C status register

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

(5) Process of after STOP condition generating

Do not write data in the I^2C data shift register (S0) and the I^2C status register (S1) until the bus busy flag BB becomes "0" after generating the STOP condition in the master mode. Because the STOP condition waveform might not be normally generated. Reading to the above registers does not have the problem.

(6) STOP condition input at 7th clock pulse

The SDA line may be held at LOW even if flag BB is set to "0" when all the following conditions are satisfied:

•The STOP condition is input at the 7th clock pulse while receiving a slave address or data.
•The clock pulse is continuously input.
•In the slave mode

Countermeasure:

Write dummy data to the I^2C shift register or reset the ES0 bit in the S1D register (ES0 = "L" → ES0 = "H") during a stop condition interrupt routine with flag PIN = "1".

Note: Do not use the read-modify-write instruction at this time. Furthermore, when the ES0 bit is set to "0", the SDA pin becomes a general-purpose port ; so that the port must be set to input mode or output "H".

(7) ES0 bit switch

In standard clock mode when SSC = "00010_2" or in high-speed clock mode, flag BB may switch to "1" if ES0 bit is set to "1" when SDA is "L".

Countermeasure:

Set ES0 to "1" when SDA is "H".

3.3.7 Notes on PWM

- For PWM_0 output, "L" level is output first.
- After data is set to the PWM0L and the PWM0H registers, PWM waveform corresponding to the new data is output from next repetitive period.

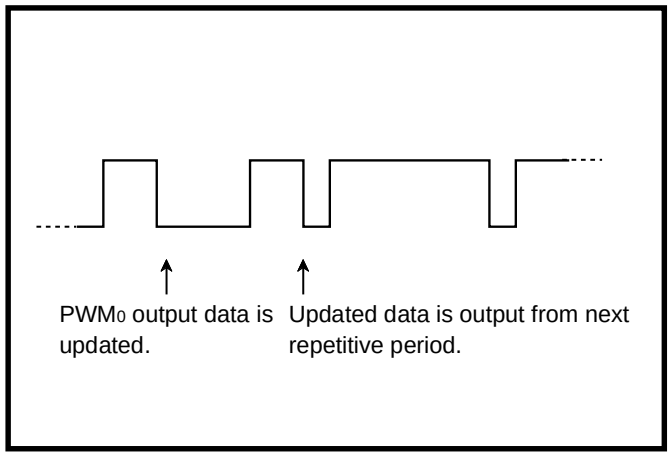


Fig. 3.3.5 PWM_0 output

3.3.8 Notes on A-D converter

(1) Analog input pin

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

■ **Reason**

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

(2) A-D converter power source pin

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

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

■ **Reason**

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

(3) Clock frequency during A-D conversion

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

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

3.3.9 Notes on D-A converter

(1) Vcc when using D-A converter

The D-A converter accuracy when Vcc is 4.0 V or less differs from that of when Vcc is 4.0 V or more. When using the D-A converter, we recommend using a Vcc of 4.0 V or more.

(2) D-Ai conversion register when not using D-A converter

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

3.3 Notes on use

3.3.10 Notes on watchdog timer

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

3.3.11 Notes on $\overline{\text{RESET}}$ pin

(1) Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the V_{SS} pin. Use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

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

■ **Reason**

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

3.3.12 Notes on CPU reprogramming mode

(1) Transfer the CPU reprogramming mode control program to the internal RAM before selecting the CPU reprogramming mode, and then, execute it on the internal RAM. Additionally, when the subroutine or stack operation instruction is used in the control program, make sure the control program is not destroyed by the stack operation.

(2) Make sure each instruction description (specified address etc.) is correct, because the CPU reprogramming mode control program is transferred to the internal RAM and executed on the internal RAM.

(3) In order to avoid generation of a watchdog timer reset, write to the watchdog timer control register periodically during the CPU reprogramming mode control program (refer to "2.7 Watchdog timer").

(4) Notes on flash memory version

The CNV_{SS} pin is connected to the internal memory circuit block by a low-ohmic resistance, since it works as a program power source pin (V_{PP} pin), as well.

To improve the noise margin, connect the CNV_{SS} pin to V_{SS} through 1 to 10 kΩ resistor.

When the CNV_{SS} pin of the mask ROM version is connected to Vss through this resistor, the function of mask ROM version works well in the same manner as flash memory version.

3.3.13 Notes on using stop mode

■**Clock restoration**

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

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

3.3.14 Notes on wait mode

■**Clock restoration**

If the wait mode is released by a reset when X_{CIN} is set as the system clock and X_{IN} oscillation is stopped during execution of the WIT instruction, X_{CIN} oscillation stops, X_{IN} oscillation starts, and X_{IN} is set as the system clock.

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

3.3.15 Notes on low-speed operation mode

(1) Using sub-clock

To use a sub-clock, fix bit 3 of the CPU mode register to "1" and control the Rd (refer to Figure 3.3.6) resistance value to a certain level to stabilize an oscillation. For resistance value of Rd, consult the oscillator manufacturer.

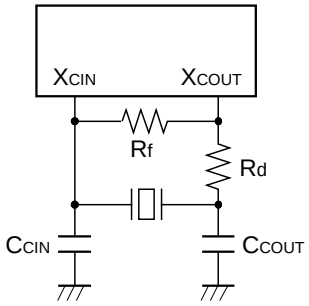


Fig. 3.3.6 Ceramic resonator circuit

■ **Reason**

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

3.3.16 Notes on restarting oscillation

(1) Restarting oscillation

Usually, when the MCU stops the clock oscillation by STP instruction and the STP instruction has been released by an external interrupt source, the fixed values of Timer 1 and Prescaler 12 (Timer 1 = 01_{16}, Prescaler 12 = FF_{16}) are automatically reloaded in order for the oscillation to stabilize. The user can inhibit the automatic setting by writing "1" to bit 6 of the port control register 2 (address $002F_{16}$).

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

■ **Reason**

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

3.3 Notes on use

3.3.17 Notes on programming

(1) Processor status register

① **Initializing of processor status register**

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

■ **Reason**

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

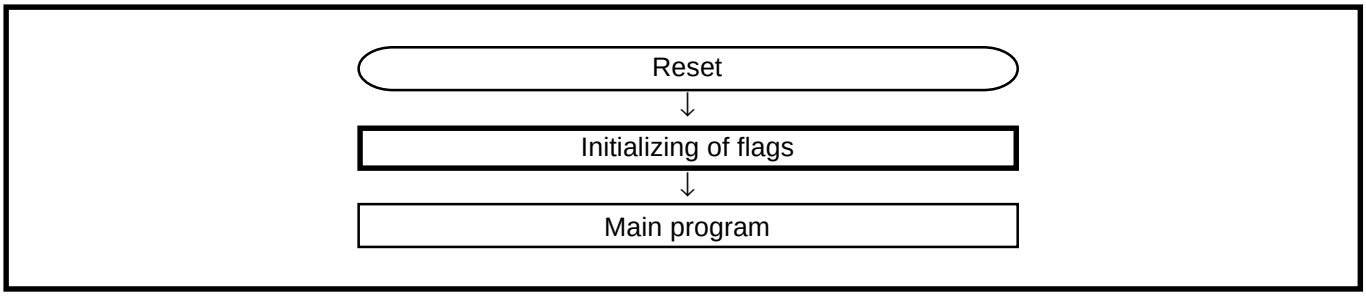


Fig. 3.3.7 Initialization of processor status register

② **How to reference the processor status register**

To reference the contents of the processor status register (PS), execute the **PHP** instruction once then read the contents of (S+1). If necessary, execute the **PLP** instruction to return the PS to its original status.
A **NOP** instruction should be executed after every **PLP** instruction.

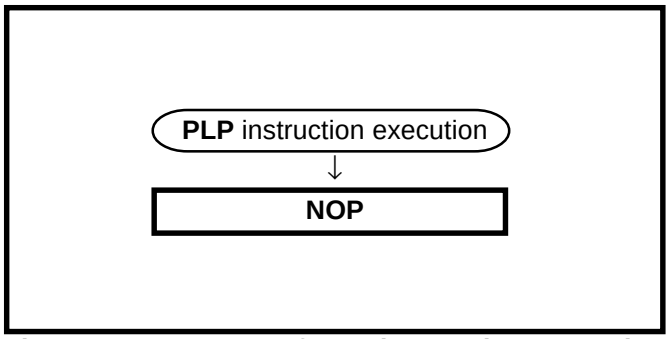


Fig. 3.3.8 Sequence of PLP instruction execution

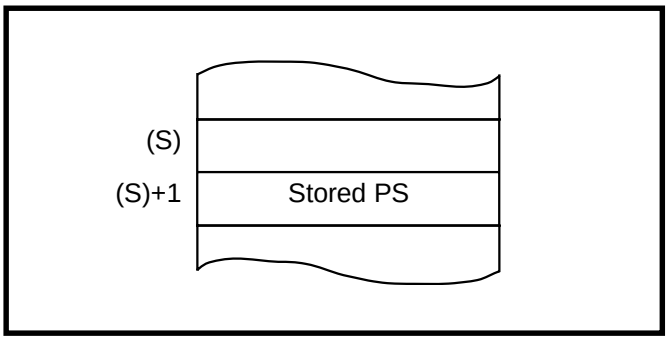


Fig. 3.3.9 Stack memory contents after PHP instruction execution

(2) BRK instruction

① **Detection of interrupt source**
It can be detected that the **BRK** instruction interrupt event or the least priority interrupt event by referring the stored B flag state. Refer to the stored B flag state in the interrupt routine.

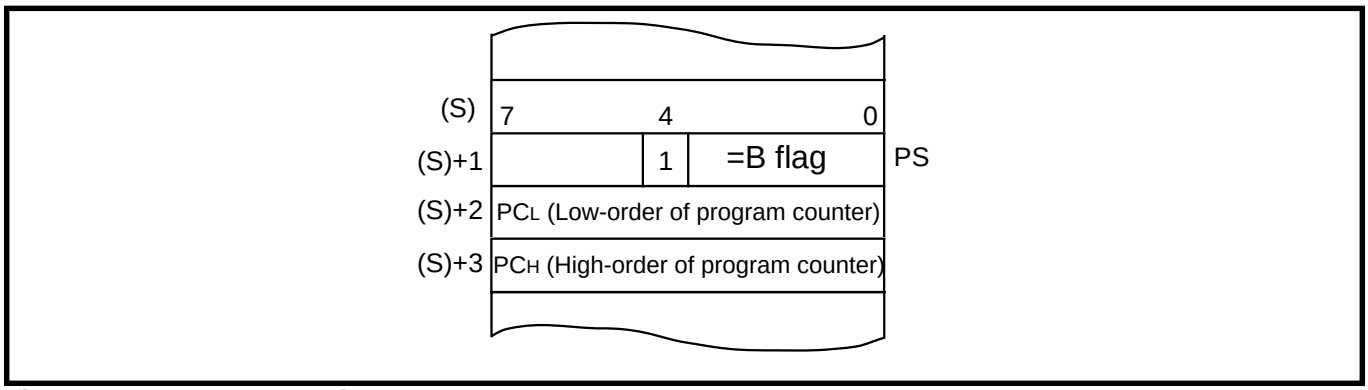


Fig. 3.3.10 Interrupt routine

② **Interrupt priority level**
When the **BRK** instruction is executed with the following conditions satisfied, the interrupt execution is started from the address of interrupt vector which has the highest priority.
- Interrupt request bit and interrupt enable bit are set to "1".
- Interrupt disable flag (I) is set to "1" to disable interrupt.

(3) Decimal calculations

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

② **Notes on status flag in decimal mode**
When decimal mode is selected, the values of three of the flags in the status register (the N, V, and Z flags) are invalid after a **ADC** or **SBC** instruction is executed.
The carry flag (C) is set to "1" if a carry is generated as a result of the calculation, or is cleared to "0" if a borrow is generated. To determine whether a calculation has generated a carry, the C flag must be initialized to "0" before each calculation. To check for a borrow, the C flag must be initialized to "1" before each calculation.

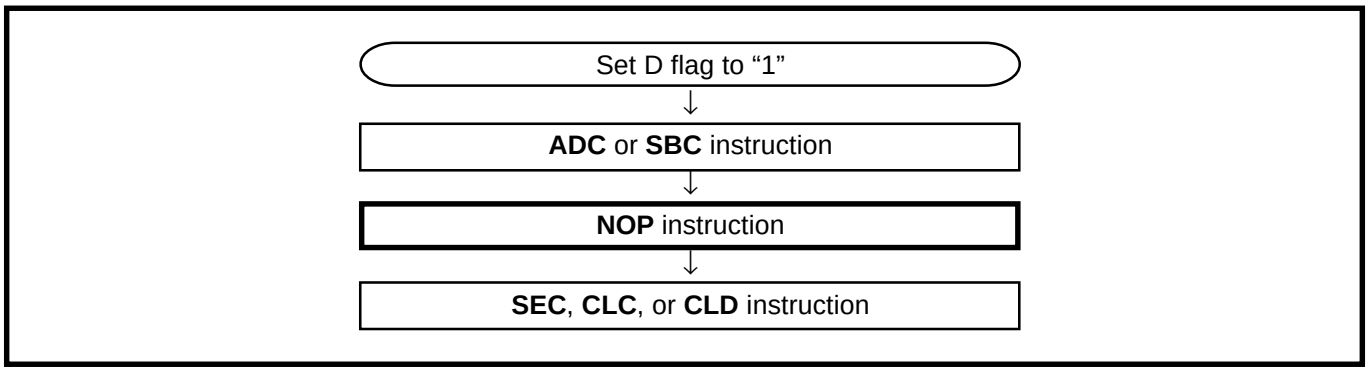


Fig. 3.3.11 Status flag at decimal calculations

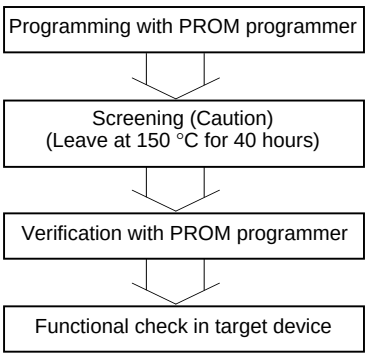
(4) JMP instruction

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

3.3.18 Programming and test of built-in PROM version

As for in the One Time PROM version (shipped in blank) and the built-in EPROM version, their built-in PROM can be read or programmed with a general-purpose PROM programmer using a special programming adapter.

The built-in EPROM version is available only for program development and on-chip program evaluation. The programming test and screening for PROM of the One Time PROM version (shipped in blank) are not performed in the assembly process and the following processes. To ensure reliability after programming, performing programming and test according to the Figure 3.3.12 before actual use are recommended.



Caution: The screening temperature is far higher than the storage temperature. Never expose to 150 °C exceeding 100 hours.

Fig. 3.3.12 Programming and testing of One Time PROM version

3.3.19 Notes on built-in PROM version

(1) Programming adapter

Use a special programming adapter shown in Table 3.3.1 and a general-purpose PROM programmer when reading from or programming to the built-in PROM in the built-in PROM version.

Table 3.3.1 Programming adapters

Microcomputer	Programming adapter
M38867E8AFS	PCA4738L-80A
M38867E8AHP (One Time PROM version shipped in blank)	PCA4738H-80A

(2) Programming/reading

In PROM mode, operation is the same as that of the M5M27C101AK, but programming conditions of PROM programmer are not set automatically because there are no internal device ID codes. Accurately set the following conditions for data programming/reading. Take care not to apply 21 V to V_{PP} pin (is also used as the CNV_{SS} pin), or the product may be permanently damaged.

- Programming voltage: 12.5 V
- Setting of PROM programmer switch: refer to Table 3.3.2.

Table 3.3.2 PROM programmer address setting

Product name format	PROM programmer start address	PROM programmer end address
M38867E8AFS	Address 8080_{16}	Address $0FFFD_{16}$
M38867E8AHP		

Note: Addresses 8080_{16} to $FFFD_{16}$ in the built-in PROM corresponds to addresses 0808_{16} to $0FFFD_{16}$ in the PROM programmer.

(3) Erasing

Contents of the windowed EPROM are erased through an ultraviolet light source of the wavelength 2537 Ångstrom. At least 15 W • sec/cm^2 are required to erase EPROM contents.

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

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

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

● **Reason**

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

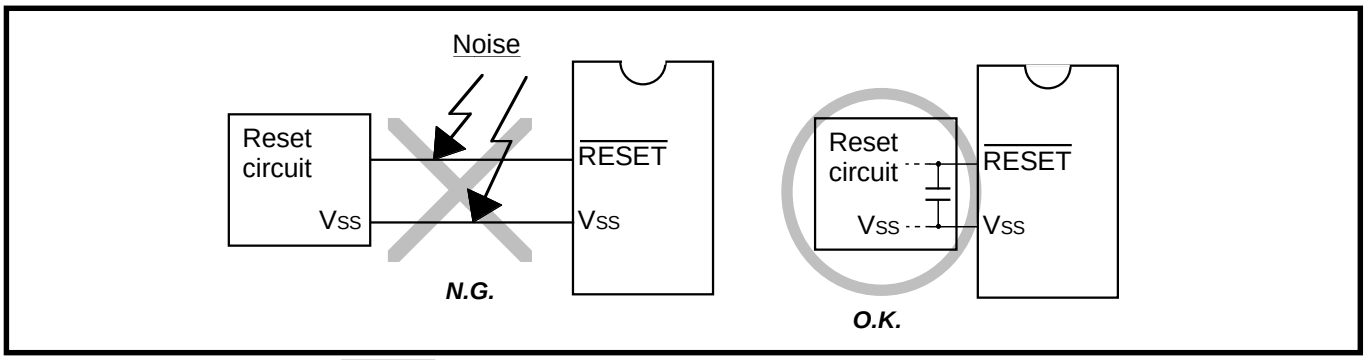


Fig. 3.4.1 Wiring for the $\overline{\text{RESET}}$ pin

(2) Wiring for clock input/output pins

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

● **Reason**

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

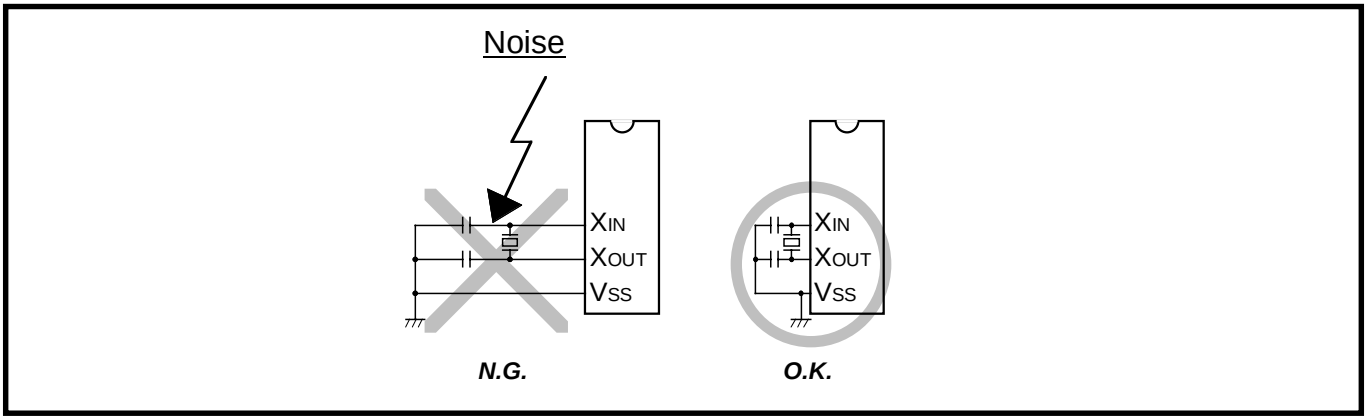


Fig. 3.4.2 Wiring for clock I/O pins

(3) Wiring to CNVSS pin

Connect the CNVSS pin to the VSS pin with the shortest possible wiring.

● **Reason**

The processor mode of a microcomputer is influenced by a potential at the CNVSS pin. If a potential difference is caused by the noise between pins CNVSS and VSS, the processor mode may become unstable. This may cause a microcomputer malfunction or a program runaway.

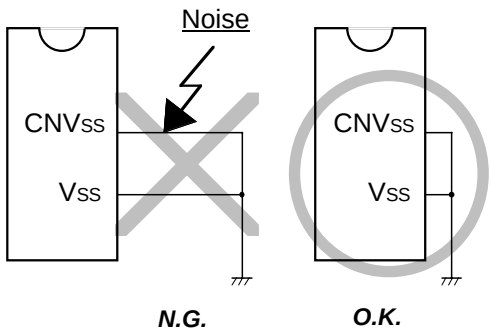


Fig. 3.4.3 Wiring for CNVSS pin

(4) Wiring to VPP pin of One Time PROM version and EPROM version

Connect an approximately 5 kΩ resistor to the VPP pin the shortest possible in series and also to the VSS pin. When not connecting the resistor, make the length of wiring between the VPP pin and the VSS 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 VPP 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 VPP 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 VPP pin, abnormal instruction codes or data are read from the built-in PROM, which may cause a program runaway.

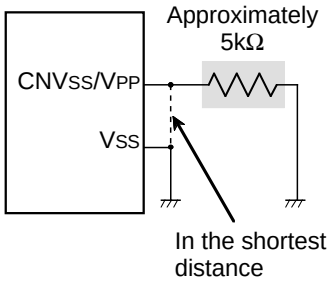


Fig. 3.4.4 Wiring for the VPP pin of the One Time PROM version and the EPROM version

APPENDIX

3.4 Countermeasures against noise

3.4.2 Connection of bypass capacitor across Vss line and Vcc line

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

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

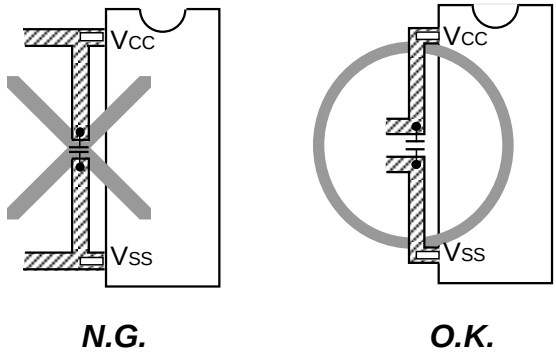


Fig. 3.4.5 Bypass capacitor across the Vss line and the Vcc line

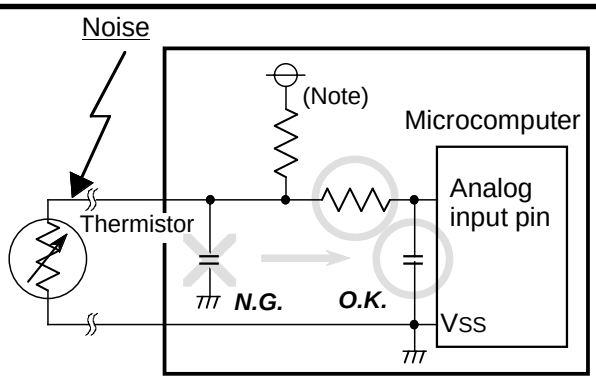
3.4.3 Wiring to analog input pins

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

● **Reason**

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

If a capacitor between an analog input pin and the Vss pin is grounded at a position far away from the Vss pin, noise on the GND line may enter a microcomputer through the capacitor.



Note : The resistor is used for dividing resistance with a thermistor.

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.

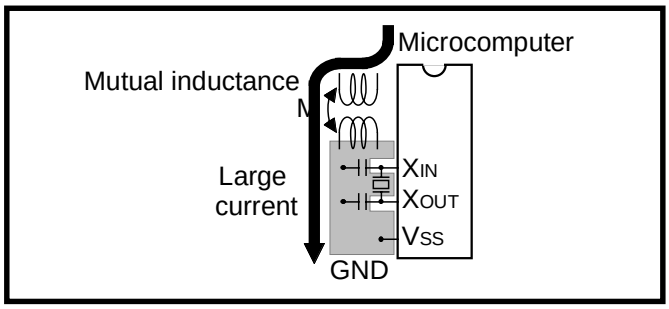


Fig. 3.4.7 Wiring for a large current signal line

(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.

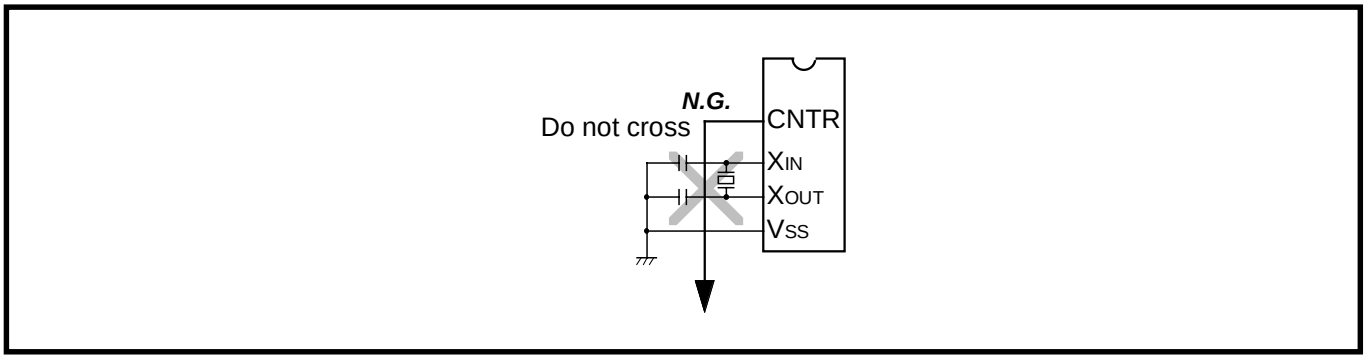


Fig. 3.4.8 Wiring of $\overline{\text{RESET}}$ pin

3.4 Countermeasures against noise

(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.

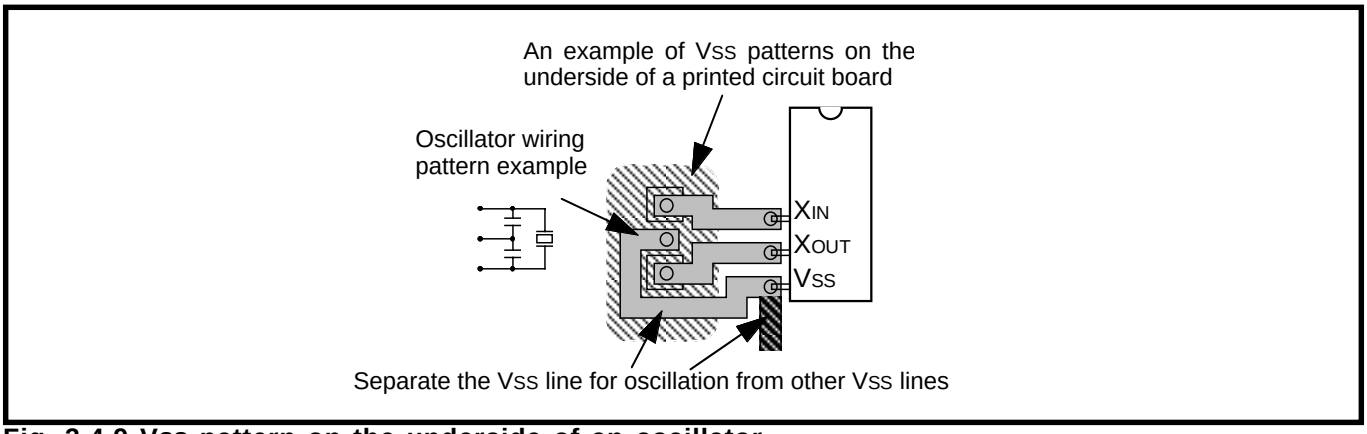


Fig. 3.4.9 Vss pattern on the underside of an oscillator

3.4.5 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 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.

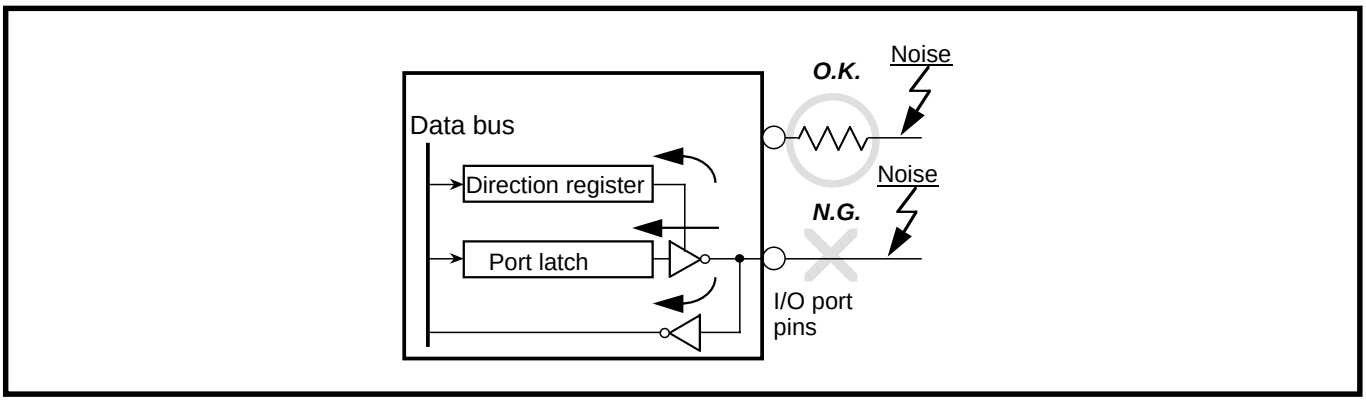


Fig. 3.4.10 Setup for I/O ports

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.

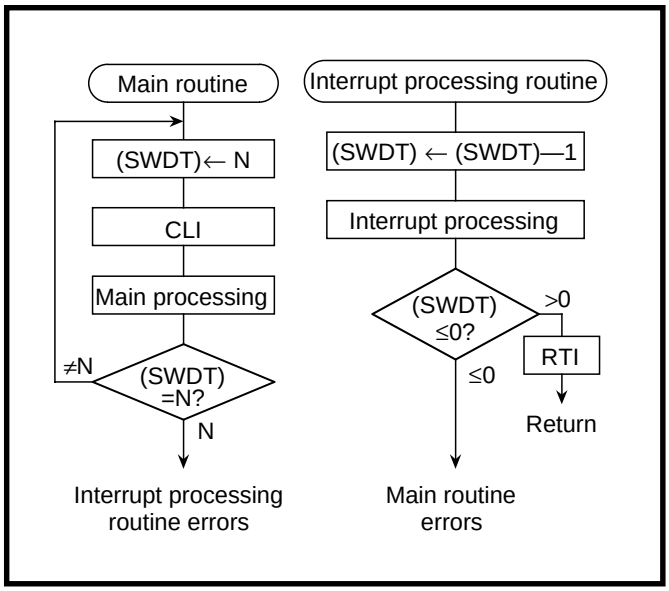


Fig. 3.4.11 Watchdog timer by software

3.5 List of registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (Pi) (i = 0, 1, 2, 3, 4, 5, 6, 7, 8)
[Address : 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}, $0A_{16}$, $0C_{16}$, $0E_{16}$, 10_{16}]

B	Name	Function	At reset	R	W
0	Port Pi_0	• In output mode Write } Port latch Read } • In input mode Write : Port latch Read : Value of pins	Undefined	○	○
1	Port Pi_1		Undefined	○	○
2	Port Pi_2		Undefined	○	○
3	Port Pi_3		Undefined	○	○
4	Port Pi_4		Undefined	○	○
5	Port Pi_5		Undefined	○	○
6	Port Pi_6		Undefined	○	○
7	Port Pi_7		Undefined	○	○

Fig. 3.5.1 Structure of Port Pi

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (PiD) (i = 0, 1, 2, 3, 4, 5, 6, 7, 8)
[Address : 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, $0F_{16}$, 11_{16}]

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

Fig. 3.5.2 Structure of Port Pi direction register

I²C data shift register

b7 b6 b5 b4 b3 b2 b1 b0

I²C data shift register (S0) [Address : 12_{16}]

B	Function	At reset	R	W
0	This register is an 8-bit shift register to store receive data or write transmit data.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Note: Secure 8 machine cycles from clearing MST bit to "0" (slave mode) until writing data to I²C data shift register.
If executing the read-modify-write instruction(SEB, CLB etc.) for this register during transfer, data may become a value not intended.

Fig. 3.5.3 Structure of I²C data shift register

I²C address register

b7 b6 b5 b4 b3 b2 b1 b0

I²C address register (S0D) [Address : 13_{16}]

B	Name	Function	At reset	R	W
0	Read / $\overline{\text{write}}$ bit (RWB)	0 : Write bit 1 : Read bit	0	○	○
1	Slave address (SAD0, SAD1, SAD2, SAD3, SAD4, SAD5, SAD6)	These bits are compared with the address data transmitted from the master.	0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Note: If the read-modify-write instruction(SEB, CLB, etc.) is executed for this register at detecting the stop condition, data may become a value not to intend.

Fig. 3.5.4 Structure of I²C address register

3.5 List of registers

I²C status register

b7 b6 b5 b4 b3 b2 b1 b0

I²C status register (S1) [Address : 14_{16}]

B	Name	Function	At reset	R	W
0	Last receive bit (LRB)	0 : Last bit = "0" 1 : Last bit = "1" (**Note1**)	?	○	×
1	General call detecting flag(AD0)	0 : No general call detected 1 : General call detected(**Note1, 2**)	0	○	×
2	Slave address comparison flag (AAS)	0 : Address disagreement 1 : Address agreement (**Note1, 2**)	0	○	×
3	Arbitration lost detecting flag (AL)	0 : Not detected 1 : Detected (**Note1**)	0	○	×
4	SCL pin low hold bit (PIN)	0 : SCL pin low hold 1 : SCL pin low release	1	○	✻
5	Bus busy flag (BB)	0 : Bus free 1 : Bus busy	0	○	○
6	Communication mode specification bits (TRX, MST)	00 : Slave receive mode 01 : Slave transmit mode 10 : Master receive mode 11 : Master transmit mode	0	○	○
7			0	○	○

Notes 1: These bits and flags can be read out, but cannot be written.
2: These bits can be detected when data format selection bit (ALS) of I²C control register is "0".
3: Do not execute the read-modify-write instruction (SEB, CLB) for this register because all bits of this register are changed by hardware.
✻: "1" can be written to this bit, but "0" cannot be written by program.

Fig. 3.5.5 Structure of I²C status register

I²C control register

b7	b6	b5	b4	b3	b2	b1	b0

I²C control register (S1D) [Address :15₁₆]

B	Name	Function	At reset	R	W
0 1 2	Bit counter (Number of transmit/receive bits) (BC0, BC1, BC2)	b2 b1 b0 0 0 0 : 8 0 0 1 : 7 0 1 0 : 6 0 1 1 : 5 1 0 0 : 4 1 0 1 : 3 1 1 0 : 2 1 1 1 : 1	0	○	○
3	I²C-BUS interface enable bit (ES0)	0 : Disabled 1 : Enabled	0	○	○
4	Data format selection bit (ALS)	0 : Addressing format 1 : Free data format	0	○	○
5	Addressing format selection bit (10 BIT SAD)	0 : 7-bit addressing format 1 : 10-bit addressing format	0	○	○
6	System clock stop selection bit (CLKSTP)	0 : System clock stop when executing WIT or STP instruction 1 : Not system clock stop when executing WIT instruction (Do not use the STP instruction.)	0	○	○
7	I²C-BUS interface pin input level selection bit (TISS)	0 : CMOS input 1 : SMBUS input	0	○	○

Notes : When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended.

Fig. 3.5.6 Structure of I²C control register

APPENDIX

3.5 List of registers

I²C clock control register

b7	b6	b5	b4	b3	b2	b1	b0

I²C clock control register (S2) [Address : 16_{16}]

B	Name	Function	At reset	R	W
0	SCL frequency control bits (CCR0, CCR1, CCR2, CCR3, CCR4)	Refer to Table 3.5.1	0	○	○
1					
2					
3					
4					
5	SCL mode specification bit (FAST MODE)	0 : Standard clock mode 1 : High-speed clock mode	0	○	○
6	ACK bit (ACK BIT)	0 : ACK is returned 1 : ACK is not returned	0	○	○
7	ACK clock bit (ACK)	0 : No ACK clock 1 : ACK clock	0	○	○

Fig. 3.5.7 Structure of I²C clock control register

Table 3.5.1 Set value of I²C clock control register and SCL frequency

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

Notes 1: Each value of SCL frequency exceeds the limit at ϕ = 4 MHz or more. When using these setting value, use ϕ of 4 MHz or less.

2: The data formula of SCL frequency is described below:
ϕ/(8 × CCR value) Standard clock mode
ϕ/(4 × CCR value) High-speed clock mode (CCR value ≠ 5)
ϕ/(2 × CCR value) High-speed clock mode (CCR value = 5)
Do not set 0 to 2 as CCR value regardless of ϕ frequency.
Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

3: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at ϕ = 4 MHz). "H" duration of the clock fluctuates from –4 to +2 machine cycles in the standard clock mode, and fluctuates from –2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because "L" duration is extended instead of "H" duration reduction.
These are value when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

I²C START/STOP condition control register

b7 b6 b5 b4 b3 b2 b1 b0

I²C START/STOP condition control register (S2D) [Address : 17_{16}]

B	Name	Function	At reset	R	W
0	START/STOP condition set bit (SSC0, SSC1, SSC2, SSC3, SSC4) (**Note**)	SCL release time= $\phi(\mu s) \times (SSC+1)$ Set up time= $\phi(\mu s) \times (SSC+1)/2$ Hold time= $\phi(\mu s) \times (SSC+1)/2$	?	○	○
1					
2					
3					
4					
5	SCL/SDA interrupt pin polarity selection bit(SIP)	0 : Falling edge active 1 : Rising edge active	0	○	○
6	SCL/SDA interrupt pin selection bit (SIS)	0 : SDA valid 1 : SCL valid	0	○	○
7	START/STOP condition generating selection bit (STSPSEL)	0 : Setup/Hold time short mode 1 : Setup/Hold time long mode	0	○	○

Note: Do not set "00000_2" or an odd number to the START/STOP condition set bit (SSC4 to SSC0).

Fig. 3.5.8 Structure of I²C START/STOP condition control register

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

Transmit/Receive buffer register (TB/RB) [Address : 18_{16}]

B	Name	Function	At reset	R	W
0	The transmission data is written to or the receive data is read out from this buffer register. • At writing: A data is written to the transmit buffer register. • At reading: The contents of the receive buffer register are read out.		?	○	○
1			?	○	○
2			?	○	○
3			?	○	○
4			?	○	○
5			?	○	○
6			?	○	○
7			?	○	○

Note: The contents of transmit buffer register cannot be read out. The data cannot be written to the receive buffer register.

Fig. 3.5.9 Structure of Transmit/Receive buffer register

3.5 List of registers

Serial I/O1 status register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O1 status register (SIO1STS) [Address : 19_{16}]

B	Name	Function	At reset	R	W
0	Transmit buffer empty flag (TBE)	0 : Buffer full 1 : Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0 : Buffer empty 1 : Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0 : Transmit shift in progress 1 : Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0 : No error 1 : Overrun error	0	○	×
4	Parity error flag (PE)	0 : No error 1 : Parity error	0	○	×
5	Framing error flag (FE)	0 : No error 1 : Framing error	0	○	×
6	Summing error flag (SE)	0 : (OE) U (PE) U (FE) = 0 1 : (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "1".		1	○	×

Fig. 3.5.10 Structure of Serial I/O1 status register

Serial I/O1 control register

b7 b6 b5 b4 b3 b2 b1 b0

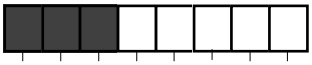
Serial I/O1 control register (SIO1CON) [Address : $1A_{16}$]

B	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : $f(X_{IN})$ ($f(X_{CIN})$ in Low Speed Mode) 1 : $f(X_{IN})/4$ ($f(X_{CIN})/4$ in Low Speed Mode)	0	○	○
1	Serial I/O1 synchronous clock selection bit (SCS)	• In clock synchronous serial I/O 0 : BRG output devided by 4 1 : External clock input • In UART 0 : BRG output devided by 16 1 : External clock input devided by 16	0	○	○
2	$\overline{S_{RDY1}}$ output enable bit (SRDY)	0 : $P4_7$ pin operates as ordinary I/O pin 1 : $P4_7$ pin operates as $\overline{S_{RDY1}}$ output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0 : Interrupt when transmit buffer has emptied 1 : Interrupt when transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0 : Transmit disabled 1 : Transmit enabled	0	○	○
5	Receive enable bit (RE)	0 : Receive disabled 1 : Receive enabled	0	○	○
6	Serial I/O1 mode selection bit (SIOM)	0 : Clock asynchronous(UART) serial I/O 1 : Clock synchronous serial I/O	0	○	○
7	Serial I/O1 enable bit (SIOE)	0 : Serial I/O1 disabled (pins $P4_4$ to $P4_7$ operate as ordinary I/O pins) 1 : Serial I/O1 enabled (pins $P4_4$ to $P4_7$ operate as serial I/O pins)	0	○	○

Fig. 3.5.11 Structure of Serial I/O1 control register

UART control register

b7 b6 b5 b4 b3 b2 b1 b0



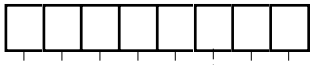
UART control register (UARTCON) [Address : $1B_{16}$]

B	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	○	○
1	Parity enable bit (PARE)	0 : Parity checking disabled 1 : Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	○	○
4	$P4_5$/TxD P-channel output disable bit (POFF)	In output mode 0 : CMOS output 1 : N-channel open-drain output	0	○	○
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "1".		1	○	×
6			1	○	×
7			1	○	×

Fig. 3.5.12 Structure of UART control register

Baud rate generator

b7 b6 b5 b4 b3 b2 b1 b0



Baud rate generator (BRG) [Address : $1C_{16}$]

B	Function	At reset	R	W
0	Set a count value of baud rate generator.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 3.5.13 Structure of Baud rate generator

3.5 List of registers

Serial I/O2 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 control register (SIO2CON) [Address : $1D_{16}$]

B	Name	Function	At reset	R	W
0	Internal synchronous clock selection bits	b2 b1 b0 0 0 0: f(X_{IN})/8 0 0 1: f(X_{IN})/16 0 1 0: f(X_{IN})/32 0 1 1: f(X_{IN})/64 1 1 0: f(X_{IN})/128 1 1 1: f(X_{IN})/256	0	○	○
1			0	○	○
2			0	○	○
3	Serial I/O2 port selection bit	0: I/O port ($P7_1$, $P7_2$) 1: S_{OUT2},S_{CLK2} signal output	0	○	○
4	$\overline{S_{RDY2}}$ output enable bit	0: I/O port ($P7_3$) 1: $\overline{S_{RDY2}}$ signal output	0	○	○
5	Transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
6	Serial I/O2 synchronous clock selection bit	0: External clock 1: Internal clock	0	○	○
7	Comparator reference input selection bit	0: $P0_0$/$P3_{REF}$ input 1: Reference input fixed	0	○	○

Fig. 3.5.14 Structure of Serial I/O2 control register

Watchdog timer control register

b7 b6 b5 b4 b3 b2 b1 b0

Watchdog timer control register (WDTCON) [Address : $1E_{16}$]

B	Name	Function	At reset	R	W
0	Watchdog timer H (for read-out of high-order 6 bits)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0 : STP instruction enabled 1 : STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0 : Watchdog timer L underflow 1 : f(X_{IN})/16 or f(X_{CIN})/16	0	○	○

Fig. 3.5.15 Structure of Watchdog timer control register

Serial I/O2 register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 register (SIO2) [Address : $1F_{16}$]

B	Name	Function	At reset	R	W
0		•Serial I/O2 register is the shift register for serial transfer. •At transmit: transmit data is set. •At receive: receive data is set.	?	○	○
1			?	○	○
2			?	○	○
3			?	○	○
4			?	○	○
5			?	○	○
6			?	○	○
7			?	○	○

Fig. 3.5.16 Structure of Serial I/O2 register

Prescaler 12, Prescaler X, Prescaler Y

b7 b6 b5 b4 b3 b2 b1 b0

Prescaler 12 (PRE12) [Address : 20_{16}]
Prescaler X (PREX) [Address : 24_{16}]
Prescaler Y (PREY) [Address : 26_{16}]

B	Name	Function	At reset	R	W
0		•Set a count value of each prescaler. •The value set in this register is written to both each prescaler and the corresponding prescaler latch at the same time. •When this register is read out, the count value of the corresponding prescaler is read out.	1	○	○
1			1	○	○
2			1	○	○
3			1	○	○
4			1	○	○
5			1	○	○
6			1	○	○
7			1	○	○

Fig. 3.5.17 Structure of Prescaler 12, Prescaler X, Prescaler Y

Timer 1

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1 (T1) [Address : 21_{16}]

B	Name	Function	At reset	R	W
0		•Set a count value of timer 1. •The value set in this register is written to both timer 1 and timer 1 latch at the same time. •When this register is read out, the timer 1's count value is read out.	1	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Fig. 3.5.18 Structure of Timer 1

Timer 2, Timer X, Timer Y

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2 (T2) [Address : 22_{16}]
Timer X (TX) [Address : 25_{16}]
Timer Y (TY) [Address : 27_{16}]

B	Name	Function	At reset	R	W
0		•Set a count value of each timer. •The value set in this register is written to both each timer and each timer latch at the same time. •When this register is read out, each timer's count value is read out.	1	○	○
1			1	○	○
2			1	○	○
3			1	○	○
4			1	○	○
5			1	○	○
6			1	○	○
7			1	○	○

Fig. 3.5.19 Structure of Timer 2, Timer X, Timer Y

Timer XY mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer XY mode register (TM) [Address : 23 16]

B	Name	Function	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	$CNTR_0$ active edge selection bit	The function depends on the operating mode of Timer X. (Refer to Table 3.5.2)	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Timer Y operating mode bits	b5 b4 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
5			0	○	○
6	$CNTR_1$ active edge selection bit	The function depends on the operating mode of Timer Y. (Refer to Table 3.5.2)	0	○	○
7	Timer Y count stop bit	0 : Count start 1 : Count stop	0	○	○

Fig. 3.5.20 Structure of Timer XY mode register

Table 3.5.2 $CNTR_0$ /$CNTR_1$ active edge selection bit function

Timer X /Timer Y operation modes	$CNTR_0$ / $CNTR_1$ active edge selection bit (bits 2, 6 of address 23_{16}) contents	
Timer mode	"0"	$CNTR_0$ / $CNTR_1$ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	$CNTR_0$ / $CNTR_1$ interrupt request occurrence: Rising edge ; No influence to timer count
Pulse output mode	"0"	Pulse output start: Beginning at "H" level $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Rising edge
Event counter mode	"0"	Timer X / Timer Y: Rising edge count $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: Falling edge count $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer X / Timer Y: "H" level width measurement $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: "L" level width measurement $CNTR_0$ / $CNTR_1$ interrupt request occurrence: Rising edge

Data bus buffer register i

b7 b6 b5 b4 b3 b2 b1 b0

Data bus buffer register i (DBBi) (i=0, 1) [Address : $28_{16}/2B_{16}$]

B	Name	Function	At reset	R	W
0	• Buffer register to write output data and read input data. • at write: data is written to output data buffer register. • at read: the contents of input data buffer register are read out.		?	○	○
1			?	○	○
2			?	○	○
3			?	○	○
4			?	○	○
5			?	○	○
6			?	○	○
7			?	○	○

Note: Output data bus buffer and input data bus buffer are ssigned to the same address.
The contents of output data bus buffer register cannot be read out.
Writing to input data bus buffer register is disabled.

Fig. 3.5.21 Structure of Data bus buffer register

Data bus buffer status register i

b7 b6 b5 b4 b3 b2 b1 b0

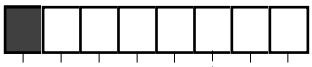
Data bus buffer status register i (DBBSTSi) (i=0, 1) [Address $29_{16}/2C_{16}$]

B	Name	Function	At reset	R	W
0	Output buffer full flag i	0 : Buffer empty 1 : Buffer full	0	○	×
1	Input buffer full flag i	0 : Buffer empty 1 : Buffer full	0	○	×
2	User definable flags (This flag can be defined by user freely.)		0	○	○
3	A_{0i} flag	This flag indicates the condition of A_{0i} status when the IBF_i flag is set.	0	○	×
4	User definable flags (This flag can be defined by user freely.)		0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Fig. 3.5.22 Structure of Data bus buffer status register

Data bus buffer control register

b7 b6 b5 b4 b3 b2 b1 b0



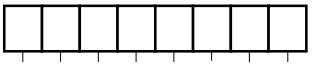
Data bus buffer control register (DBBCON) [Address $2A_{16}$]

B	Name	Function	At reset	R	W
0	Data bus buffer enable bit	0 : $P5_0$–$P5_3$, P8 I/O port 1 : Data bus buffer enabled	0	○	○
1	Data bus buffer function selection bit	0 : Single data bus buffer mode ($P4_7$ functions as I/O port.) 1 : Double data bus buffer mode ($P4_7$ functions as $\overline{S_1}$ input.)	0	○	○
2	OBF_0 output selection bit	0 : OBF_{00} valid 1 : OBF_{01} valid	0	○	○
3	OBF_{00} output enable bit	0 : $P4_2$ functions as port I/O pin. 1 : $P4_2$ functions as OBF_{00} output pin.	0	○	○
4	OBF_{01} output enable bit	0 : $P4_3$ functions as port I/O pin. 1 : $P4_3$ functions as OBF_{01} output pin.	0	○	○
5	OBF_{10} output enable bit	0 : $P4_6$ functions as port I/O pin. 1 : $P4_6$ functions as OBF_{10} output pin.	0	○	○
6	Input level selection bit	0 : CMOS level input 1 : TTL level input	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 3.5.23 Structure of Data bus buffer control register

Comparator data register

b7 b6 b5 b4 b3 b2 b1 b0



Comparator data register (CMPD) [Address : $2D_{16}$]

B	Name	Function	At reset	R	W
0	• At writing: The voltage comparison is immediately performed by writing operation. • At reading: The comparison result is read out.		?	○	○
1			?	○	○
2			?	○	○
3			?	○	○
4			?	○	○
5			?	○	○
6			?	○	○
7			?	○	○

Fig. 3.5.24 Structure of Comparator data register

3.5 List of registers

Port control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Port control register 1 (PCTL1) [Address $2E_{16}$]

B	Name	Function	At reset	R	W
0	$P0_0$–$P0_3$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	○	○
1	$P0_4$–$P0_7$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	○	○
2	$P1_0$–$P1_3$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	○	○
3	$P1_4$–$P1_7$ output structure selection bit	0: CMOS 1: N-channel open-drain	0	○	○
4	$P3_0$–$P3_3$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
5	$P3_4$–$P3_7$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
6	PWM_0 enable bit	0: PWM_0 output disabled 1: PWM_0 output enabled	0	○	○
7	PWM_1 enable bit	0: PWM_1 output disabled 1: PWM_1 output enabled	0	○	○

Fig. 3.5.25 Structure of Port control register 1

Port control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Port control register 2 (PCTL2) [Address $2F_{16}$]

B	Name	Function	At reset	R	W
0	P4 input level selection bit ($P4_2$–$P4_6$)	0: CMOS level input 1: TTL level input	0	○	○
1	P7 input level selection bit ($P7_0$–$P7_5$)	0: CMOS level input 1: TTL level input	0	○	○
2	P4 output structure selection bit ($P4_2$, $P4_3$, $P4_4$, $P4_6$)	0: CMOS 1: N-channel open-drain	0	○	○
3	P8 function selection bit	0: Port P8/Port P8 direction register 1: Port P4 input register/Port P7 input register	0	○	○
4	INT_2, INT_3, INT_4 interrupt switch bit	0: INT_{20}, INT_{30}, INT_{40} interrupt 1: INT_{21}, INT_{31}, INT_{41} interrupt	0	○	○
5	Timer Y count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ in low-speed mode) 1: $f(X_{CIN})$	0	○	○
6	Oscillation stabilizing time set after STP instruction released bit	0: Automatic set "01_{16}" to timer 1 and "FF_{16}" to prescaler 12 1: No automatic set	0	○	○
7	Port output $P4_2$/$P4_3$ clear function selection bit	0: Only software clear 1: Software clear and output data bus buffer 0 reading (system bus side)	0	○	○

Fig. 3.5.26 Structure of Port control register 2

PWM0H register

b7 b6 b5 b4 b3 b2 b1 b0

PWM0H register (PWM0H) [Address : 30_{16}]

B	Function	At reset	R	W
0	• The high-order 8 bits of the PWM0H output data are set. • At writing: A written data is transferred to PWM0 latch at every sub-period (64 μs). ($f(X_{IN})$ = 8 MHz) • At reading: The contents of the PWM0H register are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 3.5.27 Structure of PWM0H register

PWM0L register

b7 b6 b5 b4 b3 b2 b1 b0

PWM0L register (PWM0L) [Address : 31_{16}]

B	Function	At reset	R	W
0	• The low-order 6 bits of the PWM0L output data are set. • At writing: A written data is transferred to PWM0 latch at every repetitive period (4096 μs). ($f(X_{IN})$ = 8 MHz) • At reading: The low-order 6 bits of PWM0 latch are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is “0”.	?	○	×
7	• The completion of transfer to the PWM0 latch is indicated. 0: Transfer completed. 1: Not transferred. • At writing: This bit is set to “1”.	?	○	×

Fig. 3.5.28 Structure of PWM0L register

3.5 List of registers

PWM1H register

b7 b6 b5 b4 b3 b2 b1 b0

PWM1H register (PWM1H) [Address : 32_{16}]

B	Function	At reset	R	W
0	• The high-order 8 bits of the PWM1H output data are set. • At writing: A written data is transferred to PWM1 latch at every sub-period (64 μs). ($f(X_{IN})$ = 8 MHz) • At reading: The contents of the PWM1H register are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 3.5.29 Structure of PWM1H register

PWM1L register

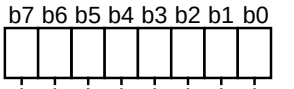
b7 b6 b5 b4 b3 b2 b1 b0

PWM1L register (PWM1L) [Address : 33_{16}]

B	Function	At reset	R	W
0	• The low-order 6 bits of the PWM1L output data are set. • At writing: A written data is transferred to PWM1 latch at every repetitive period (4096 μs). ($f(X_{IN})$ = 8 MHz) • At reading: The low-order 6 bits of PWM1 latch are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is “0”.	?	○	×
7	• The completion of transfer to the PWM1 latch is indicated. 0: Transfer completed. 1: Not transferred. • At writing: This bit is set to “1”.	?	○	×

Fig. 3.5.30 Structure of PWM1L register

AD/DA control register

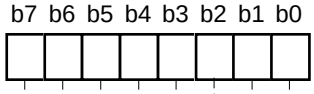


AD/DA control register
(ADCON: address 34_{16})

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	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
4	PWM_0 output pin selection bit	0: $P5_6/PWM_{01}$ 1: $P3_0/PWM_{00}$	0	○	○
5	PWM_1 output pin selection bit	0: $P5_7/PWM_{11}$ 1: $P3_1/PWM_{10}$	0	○	○
6	DA_1 output enable bit	0: DA_1 output disabled 1: DA_1 output enabled	0	○	○
7	DA_2 output enable bit	0: DA_2 output disabled 1: DA_2 output enabled	0	○	○

Fig. 3.5.31 Structure of AD/DA control register

A-D conversion register 1



A-D conversion register 1 (AD1) [Address : 35_{16}]

B	Function	At reset	R	W
0	The read-only register in which the A-D conversion's results are stored. < 8-bit read> b7 b0 b9 b8 b7 b6 b5 b4 b3 b2 < 10-bit read> b7 b0 b7 b6 b5 b4 b3 b2 b1 b0	?	○	×
1		?	○	×
2		?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

Fig. 3.5.32 Structure of AD conversion register 1

D-Ai conversion register

b7 b6 b5 b4 b3 b2 b1 b0

D-Ai conversion register (DAi) (i = 1, 2)
[Address: 36_{16}, 37_{16}]

b	Functions	At reset	R	W
0	This is D-A output value stored bits. This is write exclusive register.	0	×	○
1		0	×	○
2		0	×	○
3		0	×	○
4		0	×	○
5		0	×	○
6		0	×	○
7		0	×	○

Fig. 3.5.33 Structure of D-Ai conversion register

A-D conversion register 2

b7 b6 b5 b4 b3 b2 b1 b0

A-D conversion register 2 (AD2) [Address : 38_{16}]

B	Name	Function	At reset	R	W
0	The read-only register in which the A-D conversion's results are stored. < 10-bit read> b7 b0 b9 b8		?	○	×
1			?	○	×
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		?	○	×
3			?	○	×
4			?	○	×
5			?	○	×
6			?	○	×
7	Conversion mode selection bit	0: 10-bit A-D mode 1: 8-bit A-D mode	?	○	○

Fig. 3.5.34 Structure of A-D convesion register 2

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register [INTSEL: address 0039_{16}]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt source selection bit	0 : INT_0 interrupt 1 : Input buffer full interrupt	0	○	○
1	INT_1/output buffer empty interrupt source selection bit	0 : INT_1 interrupt 1 : Output buffer empty interrupt	0	○	○
2	Serial I/O1 transmit/S_{CL},S_{DA} interrupt source selection bit	0 : Serial I/O1 transmit interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
3	$CNTR_0$/S_{CL},S_{DA} interrupt source selection bit	0 : $CNTR_0$ interrupt ✻1 1 : S_{CL},S_{DA} interrupt	0	○	○
4	Serial I/O2/I^2C interrupt source selection bit	0 : Serial I/O2 interrupt ✻2 1 : I^2C interrupt	0	○	○
5	INT_2/I^2C interrupt source selection bit	0 : INT_2 interrupt ✻2 1 : I^2C interrupt	0	○	○
6	$CNTR_1$/key-on wake-up interrupt source selection bit	0 : $CNTR_1$ interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○
7	AD converter/key-on wake-up interrupt source selection bit	0 : A-D converter interrupt ✻3 1 : Key-on wake-up interrupt	0	○	○

✻1: Do not write "1" to these bits simultaneously.
✻2: Do not write "1" to these bits simultaneously.
✻3: Do not write "1" to these bits simultaneously.

Fig. 3.5.35 Structure of Interrupt source selection register

Interrupt edge selection register

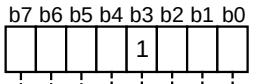
b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register (INTEDGE) [Address : $3A_{16}$]

B	Name	Function	At reset	R	W
0	INT_0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT_1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×
3	INT_2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4	INT_3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
5	INT_4 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
6	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
7			0	○	×

Fig. 3.5.36 Structure of Interrupt edge selection register

CPU mode register



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

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : Memory expansion mode **(Note)** 10 : Microprocessor mode **(Note)** 11 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to "1".		1	○	○
4	Port Xc switch bit	0: I/O port function (oscillation stopped) 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 bits	b7 b6 0 0: ϕ=f(X_{IN})/2 (high-speed mode) 0 1: ϕ=f(X_{IN})/8 (middle-speed mode) 1 0: ϕ=f(X_{CIN})/2 (low-speed mode) 1 1: not available	1	○	○
7			0	○	○

*: The initial value of bit 1 depends on the CNVss level.
Note: This mode is not available for M38869M8A/MCA/MFA or the flash memory version.

Fig. 3.5.37 Structure of CPU mode register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT_1/output buffer empty interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 3.5.38 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2 (IREQ2) [Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	$CNTR_1$/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2/I^2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT_4 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

✻: These bits can be cleared to "0" by program, but cannot be set to "1".

Fig. 3.5.39 Structure of Interrupt request register 2

3.5 List of registers

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1 (ICON1) [Address : $3E_{16}$]

B	Name	Function	At reset	R	W
0	INT_0/input buffer full interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1/output buffer empty interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 3.5.40 Structure of Interrupt control register 1

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0

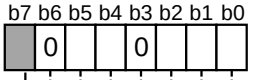
0

Interrupt control register 2 (ICON2) [Address : $3F_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$/S_{CL}, S_{DA} interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	$CNTR_1$/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2/I^2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	INT_4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 3.5.41 Structure of Interrupt control register 2

Flash memory control register



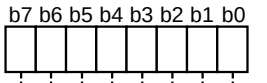
Flash memory control register
(FCON : address FFE16)

b	Name	Functions	At reset	R	W
0	CPU reprogramming mode select bit **(Note)**	0 : CPU reprogramming mode is invalid. (Normal operation mode) 1 : When applying 0 V to CNVSS/VPP pin, CPU reprogramming mode is invalid. When applying VPPH to CNVSS/VPP pin, CPU reprogramming mode is valid.	0	○	○
1	Erase/Program busy flag	0 : Erase and program are completed or have not been executed. 1 : Erase/program is being executed.	0	○	○
2	CPU reprogramming mode monitor flag	0 : CPU reprogramming mode is invalid 1 : CPU reprogramming mode is valid	0	○	○
3	Fix this bit to “0”.		0	○	○
4	Erase/Program area select bits	b5 b4 0 0: Address 1000$_{16}$ to FFFF$_{16}$ (total 60Kbytes) 0 1: Address 1000$_{16}$ to 7FFF$_{16}$ (total 28Kbytes) 1 0: Address 8000$_{16}$ to FFFF$_{16}$ (total 32Kbytes) 1 1: not available	0	○	○
5			0	○	○
6	Fix this bit to “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: Bit 0 can be reprogrammed only when 0 V is applied to the CNVSS/VPP pin.

Fig. 3.5.42 Structure of Flash memory control register

Flash command register



Flash command register
(FCMD: address FFF_{16})

b	Functions		At reset	R	W
0	Writing of software command		0	×	○
1	<Software command name>	<Command code>	0	×	○
2	• Read command	• "00_{16}"	0	×	○
3	• Program command	• "40_{16}"	0	×	○
4	• Program verify command	• "$C0_{16}$"	0	×	○
5	• Erase command	• "20_{16}"+ "20_{16}"	0	×	○
6	• Erase verify command	• "$A0_{16}$"	0	×	○
7	• Reset command	• "FF_{16}" + "FF_{16}"	0	×	○

Note: The flash command register is a write-only register.

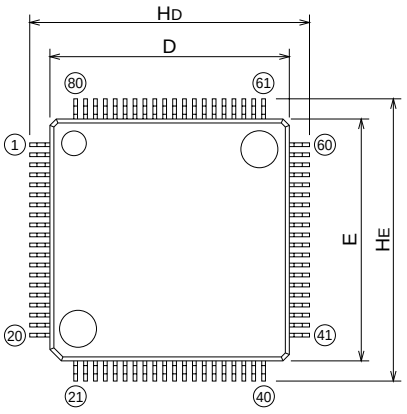
Fig. 3.5.43 Structure of Flash command register

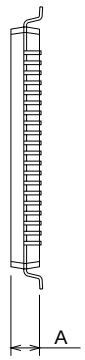
3.6 Package outline

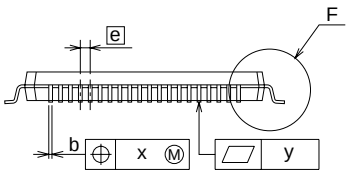
80P6Q-A (MMP)

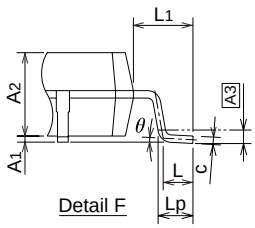
Plastic 80pin 12×12mm body LQFP

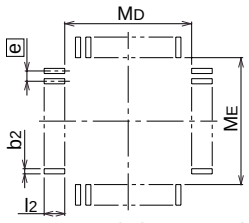
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP80-P-1212-0.5	–	0.47	Cu Alloy











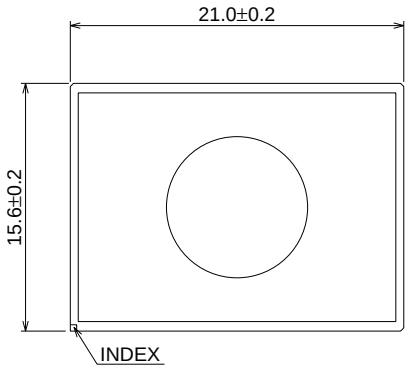
Recommended Mount Pad

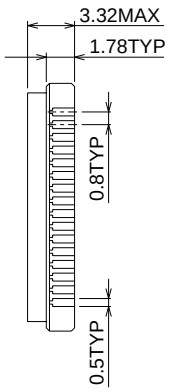
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A1	0	0.1	0.2
A2	–	1.4	–
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	11.9	12.0	12.1
E	11.9	12.0	12.1
e	–	0.5	–
HD	13.8	14.0	14.2
HE	13.8	14.0	14.2
L	0.3	0.5	0.7
L1	–	1.0	–
Lp	0.45	0.6	0.75
A3	–	0.25	–
x	–	–	0.08
y	–	–	0.1
θ	0°	–	10°
b2	–	0.225	–
l2	0.9	–	–
MD	–	12.4	–
ME	–	12.4	–

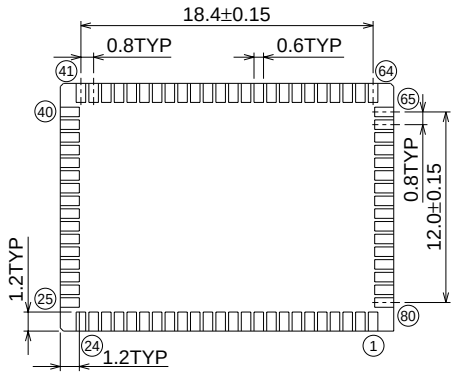
80D0

Glass seal 80pin QFN

EIAJ Package Code	JEDEC Code	Weight(g)
–	–	







3.7 Machine instructions

Symbol	Function	Details	Addressing mode																		Processor status register							
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R			7	6	5	4	3	2	1	0
			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				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				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				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.																			•	•	•	•	•	•	•	•
BCS (Note 4)	C = 1?	This instruction takes a branch to the appointed address if C is 1. The branch address is specified by a relative address. If C is 0, the next instruction is executed.																			•	•	•	•	•	•	•	•
BEQ (Note 4)	Z = 1?	This instruction takes a branch to the appointed address when Z is 1. The branch address is specified by a relative address. If Z is 0, the next instruction is executed.																			•	•	•	•	•	•	•	•
BIT	A ∧ M	This instruction takes a bit-wise logical AND of A and M contents; however, the contents of A and M are not modified. The contents of N, V, Z are changed, but the contents of A, M remain unchanged.													24	3	2				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.																			•	•	•	•	•	•	•	•
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.																			•	•	•	•	•	•	•	•

Addressing mode																																			Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2							N	V	•	•	•	•	Z	C
35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2							N	•	•	•	•	•	Z	•
16	6	2				0E	6	3	1E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																											90	2	2				•	•	•	•	•	•	•	•
																											B0	2	2				•	•	•	•	•	•	•	•
																											F0	2	2				•	•	•	•	•	•	•	•
						2C	4	3																									M_7	M_6	•	•	•	•	Z	•
																											30	2	2				•	•	•	•	•	•	•	•
																											D0	2	2				•	•	•	•	•	•	•	•

APPENDIX
3.7 Machine instructions

Symbol	Function	Details	IMP OP	IMP n	IMP #	IMM OP	IMM n	IMM #	A OP	A n	A #	BIT, A OP	BIT, A n	BIT, A #	ZP OP	ZP n	ZP #	BIT, ZP OP	BIT, ZP n	BIT, ZP #
BPL (Note 4)	N = 0?	This instruction takes a branch to the appointed address if N is 0. The branch address is specified by a relative address. If N is 1, the next instruction is executed.																		
BRA	PC ← PC ± offset	This instruction branches to the appointed address. The branch address is specified by a relative address.																		
BRK	B ← 1 (PC) ← (PC) + 2 M(S) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 M(S) ← PS S ← S – 1 I ← 1 PC_L ← AD_L PC_H ← AD_H	When the BRK instruction is executed, the CPU pushes the current PC contents onto the stack. The BADRS designated in the interrupt vector table is stored into the PC.	00	7	1															
BVC (Note 4)	V = 0?	This instruction takes a branch to the appointed address if V is 0. The branch address is specified by a relative address. If V is 1, the next instruction is executed.																		
BVS (Note 4)	V = 1?	This instruction takes a branch to the appointed address when V is 1. The branch address is specified by a relative address. When V is 0, the next instruction is executed.																		
CLB	Ai or Mi ← 0	This instruction clears the designated bit i of A or M.										1B + 20i	2	1				1F + 20i	5	2
CLC	C ← 0	This instruction clears C.	18	2	1															
CLD	D ← 0	This instruction clears D.	D8	2	1															
CLI	I ← 0	This instruction clears I.	58	2	1															
CLT	T ← 0	This instruction clears T.	12	2	1															
CLV	V ← 0	This instruction clears V.	B8	2	1															
CMP (Note 3)	When T = 0 A – M When T = 1 M(X) – M	When T = 0, this instruction subtracts the contents of M from the contents of A. The result is not stored and the contents of A or M are not modified. When T = 1, the CMP subtracts the contents of M from the contents of M(X). The result is not stored and the contents of X, M, and A are not modified. M(X) represents the contents of memory where is indicated by X.				C9	2	2							C5	3	2			
COM	M ← $\overline{M}$	This instruction takes the one's complement of the contents of M and stores the result in M.													44	5	2			
CPX	X – M	This instruction subtracts the contents of M from the contents of X. The result is not stored and the contents of X and M are not modified.				E0	2	2							E4	3	2			
CPY	Y – M	This instruction subtracts the contents of M from the contents of Y. The result is not stored and the contents of Y and M are not modified.				C0	2	2							C4	3	2			
DEC	A ← A – 1 or M ← M – 1	This instruction subtracts 1 from the contents of A or M.							1A	2	1				C6	5	2			

Addressing mode

Symbol	ZP, X OP	ZP, X n	ZP, X #	ZP, Y OP	ZP, Y n	ZP, Y #	ABS OP	ABS n	ABS #	ABS, X OP	ABS, X n	ABS, X #	ABS, Y OP	ABS, Y n	ABS, Y #	IND OP	IND n	IND #	ZP, IND OP	ZP, IND n	ZP, IND #	IND, X OP	IND, X n	IND, X #	IND, Y OP	IND, Y n	IND, Y #	REL OP	REL n	REL #	SP OP	SP n	SP #
BPL																												10	2	2			
BRA																												80	4	2			
BRK																																	
BVC																												50	2	2			
BVS																												70	2	2			
CLB																																	
CLC																																	
CLD																																	
CLI																																	
CLT																																	
CLV																																	
CMP	D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2						
COM																																	
CPX							EC	4	3																								
CPY							CC	4	3																								
DEC	D6	6	2				CE	6	3	DE	7	3																					

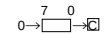
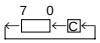
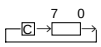
Processor status register

Symbol	7 N	6 V	5 T	4 B	3 D	2 I	1 Z	0 C
BPL	•	•	•	•	•	•	•	•
BRA	•	•	•	•	•	•	•	•
BRK	•	•	•	1	•	1	•	•
BVC	•	•	•	•	•	•	•	•
BVS	•	•	•	•	•	•	•	•
CLB	•	•	•	•	•	•	•	•
CLC	•	•	•	•	•	•	•	0
CLD	•	•	•	•	0	•	•	•
CLI	•	•	•	•	•	0	•	•
CLT	•	•	0	•	•	•	•	•
CLV	•	0	•	•	•	•	•	•
CMP	N	•	•	•	•	•	Z	C
COM	N	•	•	•	•	•	Z	•
CPX	N	•	•	•	•	•	Z	C
CPY	N	•	•	•	•	•	Z	C
DEC	N	•	•	•	•	•	Z	•

Symbol	Function	Details	IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
DEX	X ← X – 1	This instruction subtracts one from the current contents of X.	CA	2	1															
DEY	Y ← Y – 1	This instruction subtracts one from the current contents of Y.	88	2	1															
DIV	A ← (M(zz + X + 1), M(zz + X)) / A M(S) ← one's complement of Remainder S ← S – 1	Divides the 16-bit data in M(zz+(X)) (low-order byte) and M(zz+(X)+1) (high-order byte) by the contents of A. The quotient is stored in A and the one's complement of the remainder is pushed onto the stack.																		
EOR (Note 1)	When T = 0 A ← A ∀ M When T = 1 M(X) ← M(X) ∀ M	When T = 0, this instruction transfers the contents of the M and A to the ALU which performs a bit-wise Exclusive OR, and stores the result in A. When T = 1, the contents of M(X) and M are transferred to the ALU, which performs a bit-wise Exclusive OR and stores the results in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				49	2	2							45	3	2			
INC	A ← A + 1 or M ← M + 1	This instruction adds one to the contents of A or M.							3A	2	1				E6	5	2			
INX	X ← X + 1	This instruction adds one to the contents of X.	E8	2	1															
INY	Y ← Y + 1	This instruction adds one to the contents of Y.	C8	2	1															
JMP	If addressing mode is ABS PC_L ← 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.				A9	2	2							A5	3	2			
LDM	M ← nn	This instruction loads the immediate value in M.													3C	4	3			
LDX	X ← M	This instruction loads the contents of M in X.				A2	2	2							A6	3	2			
LDY	Y ← M	This instruction loads the contents of M in Y.				A0	2	2							A4	3	2			

Symbol	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
DEX																																		N	•	•	•	•	•	Z	•
DEY																																		N	•	•	•	•	•	Z	•
DIV	E2	16	2																															•	•	•	•	•	•	•	•
EOR	55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
INC	F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
INX																																		N	•	•	•	•	•	Z	•
INY																																		N	•	•	•	•	•	Z	•
JMP							4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
JSR							20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
LDA	B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
LDM																																		•	•	•	•	•	•	•	•
LDX				B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
LDY	B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•

(Columns ZP, X through SP are Addressing mode; columns 7–0 are Processor status register.)

Symbol	Function	Details	Addressing mode																		Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			7	6	5	4	3	2	1	0
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
LSR		This instruction shifts either A or M one bit to the right such that bit 7 of the result always is set to 0, and the bit 0 is stored in C.							4A	2	1				46	5	2				0	•	•	•	•	•	Z	C
MUL	M(S) • A ← A ✻ M(zz + X) S ← S – 1	Multiplies Accumulator with the memory specified by the Zero Page X address mode and stores the high-order byte of the result on the Stack and the low-order byte in A.																			•	•	•	•	•	•	•	•
NOP	PC ← PC + 1	This instruction adds one to the PC but does no otheroperation.	EA	2	1																•	•	•	•	•	•	•	•
ORA (Note 1)	When T = 0 A ← A V M When T = 1 M(X) ← M(X) V M	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise “OR”, and stores the result in A. When T = 1, this instruction transfers the contents of M(X) and the M to the ALU which performs a bit-wise OR, and stores the result in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				09	2	2							05	3	2				N	•	•	•	•	•	Z	•
PHA	S ← S – 1	This instruction pushes the contents of A to the memory location designated by S, and decrements the contents of S by one.	48	3	1																•	•	•	•	•	•	•	•
PHP	M(S) ← PS S ← S – 1	This instruction pushes the contents of PS to the memory location designated by S and decrements the contents of S by one.	08	3	1																•	•	•	•	•	•	•	•
PLA	S ← S + 1 A ← M(S)	This instruction increments S by one and stores the contents of the memory designated by S in A.	68	4	1																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		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				N	•	•	•	•	•	Z	C
ROR		This instruction shifts either A or M one bit right through C. C is stored in bit 7 and bit 0 is stored in C.							6A	2	1				66	5	2				N	•	•	•	•	•	Z	C
RRF		This instruction rotates 4 bits of the M content to the right.													82	8	2				•	•	•	•	•	•	•	•
RTI	S ← S + 1 PS ← M(S) S ← S + 1 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																•	•	•	•	•	•	•	•

Addressing mode																																				Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
56	6	2				4E	6	3	5E	7	3																						0	•	•	•	•	•	Z	C
62	15	2																															•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	(Value saved in stack)							
36	6	2				2E	6	3	3E	7	3																						N	•	•	•	•	•	Z	C
76	6	2				6E	6	3	7E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	(Value saved in stack)							
																																	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
SBC (Note 1) (Note 5)	When T = 0 A ← A – M – C When T = 1 M(X) ← M(X) – M – C	When T = 0, this instruction subtracts the value of M and the complement of C from A, and stores the results in A and C. When T = 1, the instruction subtracts the contents of M and the complement of C from the contents of M(X), and stores the results in M(X) and C. A remain unchanged, but status flag are changed. M(X) represents the contents of memory where is indicated by X.				E9	2	2							E5	3	2			
SEB	Ai or Mi ← 1	This instruction sets the designated bit i of A or M.										0B + 20i	2	1				0F + 20i	5	2
SEC	C ← 1	This instruction sets C.	38	2	1															
SED	D ← 1	This instruction set D.	F8	2	1															
SEI	I ← 1	This instruction set I.	78	2	1															
SET	T ← 1	This instruction set T.	32	2	1															
STA	M ← A	This instruction stores the contents of A in M. The contents of A does not change.													85	4	2			
STP		This instruction resets the oscillation control F/F and the oscillation stops. Reset or interrupt input is needed to wake up from this mode.	42	2	1															
STX	M ← X	This instruction stores the contents of X in M. The contents of X does not change.													86	4	2			
STY	M ← Y	This instruction stores the contents of Y in M. The contents of Y does not change.													84	4	2			
TAX	X ← A	This instruction stores the contents of A in X. The contents of A does not change.	AA	2	1															
TAY	Y ← A	This instruction stores the contents of A in Y. The contents of A does not change.	A8	2	1															
TST	M = 0?	This instruction tests whether the contents of M are “0” or not and modifies the N and Z.													64	3	2			
TSX	X ← S	This instruction transfers the contents of S in X.	BA	2	1															
TXA	A ← X	This instruction stores the contents of X in A.	8A	2	1															
TXS	S ← X	This instruction stores the contents of X in S.	9A	2	1															
TYA	A ← Y	This instruction stores the contents of Y in A.	98	2	1															
WIT		The WIT instruction stops the internal clock but not the oscillation of the oscillation circuit is not stopped. CPU starts its function after the Timer X over flows (comes to the terminal count). All registers or internal memory contents except Timer X will not change during this mode. (Of course needs VDD).	C2	2	1															

Notes 1 : The number of cycles “n” is increased by 3 when T is 1.
2 : The number of cycles “n” is increased by 2 when T is 1.
3 : The number of cycles “n” is increased by 1 when T is 1.
4 : The number of cycles “n” is increased by 2 when branching has occurred.
5 : N, V, and Z flags are invalid in decimal operation mode.

Symbol	Addressing mode																																	Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
SBC	F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
SEB																																		•	•	•	•	•	•	•	•
SEC																																		•	•	•	•	•	•	•	1
SED																																		•	•	•	•	1	•	•	•
SEI																																		•	•	•	•	•	1	•	•
SET																																		•	•	1	•	•	•	•	•
STA	95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
STP																																		•	•	•	•	•	•	•	•
STX				96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
STY	94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
TAX																																		N	•	•	•	•	•	Z	•
TAY																																		N	•	•	•	•	•	Z	•
TST																																		N	•	•	•	•	•	Z	•
TSX																																		N	•	•	•	•	•	Z	•
TXA																																		N	•	•	•	•	•	Z	•
TXS																																		•	•	•	•	•	•	•	•
TYA																																		N	•	•	•	•	•	Z	•
WIT																																		•	•	•	•	•	•	•	•

APPENDIX

3.7 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

3.8 List of instruction code

D3 – D0 / D7 – D4 (Hexadecimal notation)		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

3.9 SFR memory map

0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	Port P5 (P5)
$000B_{16}$	Port P5 direction register (P5D)
$000C_{16}$	Port P6 (P6)
$000D_{16}$	Port P6 direction register (P6D)
$000E_{16}$	Port P7 (P7)
$000F_{16}$	Port P7 direction register (P7D)
0010_{16}	Port P8 (P8)/Port P4 input register (P4I)
0011_{16}	Port P8 direction register (P8D)/Port P7 input register (P7I)
0012_{16}	I^2C data shift register (S0)
0013_{16}	I^2C address register (S0D)
0014_{16}	I^2C status register (S1)
0015_{16}	I^2C control register (S1D)
0016_{16}	I^2C clock control register (S2)
0017_{16}	I^2C start/stop condition control register (S2D)
0018_{16}	Transmit/Receive buffer register (TB/RB)
0019_{16}	Serial I/O1 status register (SIO1STS)
$001A_{16}$	Serial I/O1 control register (SIO1CON)
$001B_{16}$	UART control register (UARTCON)
$001C_{16}$	Baud rate generator (BRG)
$001D_{16}$	Serial I/O2 control register (SIO2CON)
$001E_{16}$	Watchdog timer control register (WDTCON)
$001F_{16}$	Serial I/O2 register (SIO2)

0020_{16}	Prescaler 12 (PRE12)
0021_{16}	Timer 1 (T1)
0022_{16}	Timer 2 (T2)
0023_{16}	Timer XY mode register (TM)
0024_{16}	Prescaler X (PREX)
0025_{16}	Timer X (TX)
0026_{16}	Prescaler Y (PREY)
0027_{16}	Timer Y (TY)
0028_{16}	Data bas buffer register 0 (DBB0)
0029_{16}	Data bas buffer status register 0 (DBBSTS0)
$002A_{16}$	Data bas buffer control register (DBBCON)
$002B_{16}$	Data bas buffer register 1 (DBB1)
$002C_{16}$	Data bas buffer status register 1 (DBBSTS1)
$002D_{16}$	Comparator data register (CMPD)
$002E_{16}$	Port control register 1 (PCTL1)
$002F_{16}$	Port control register 2 (PCTL2)
0030_{16}	PWM0H register (PWM0H)
0031_{16}	PWM0L register (PWM0L)
0032_{16}	PWM1H register (PWM1H)
0033_{16}	PWM1L register (PWM1L)
0034_{16}	AD/DA control register (ADCON)
0035_{16}	A-D conversion register 1 (AD1)
0036_{16}	D-A1 conversion register (DA1)
0037_{16}	D-A2 conversion register (DA2)
0038_{16}	A-D conversion register 2 (AD2)
0039_{16}	Interrupt source selection register (INTSEL)
$003A_{16}$	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)

$0FFE_{16}$	Flash memory control register (FCON)	**(Note)**
$0FFF_{16}$	Flash command register (FCMD)	**(Note)**

Note: Flash memory version only

3.10 Pin configurations

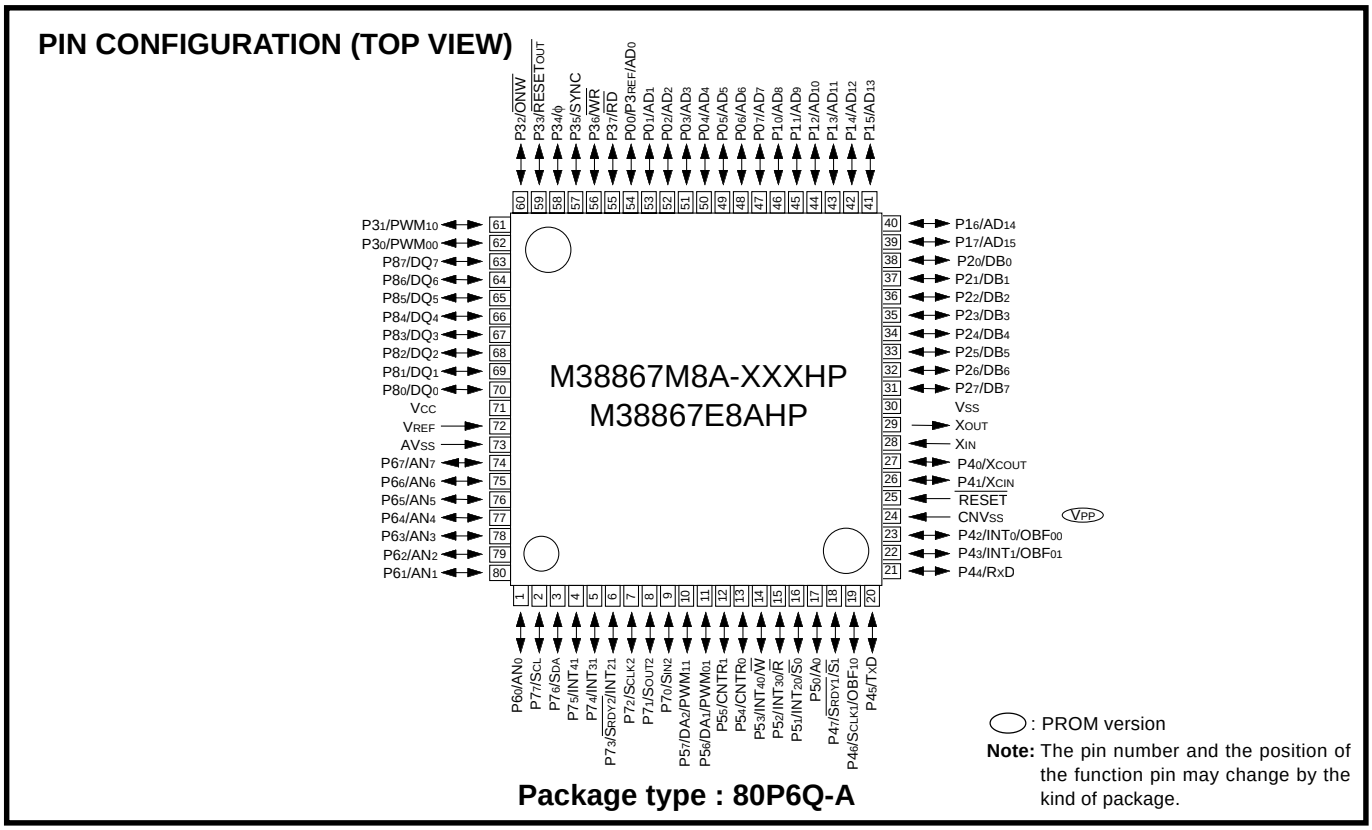


Fig. 3.10.1 M38867M8A-XXXHP, M38867E8AHP pin configuration

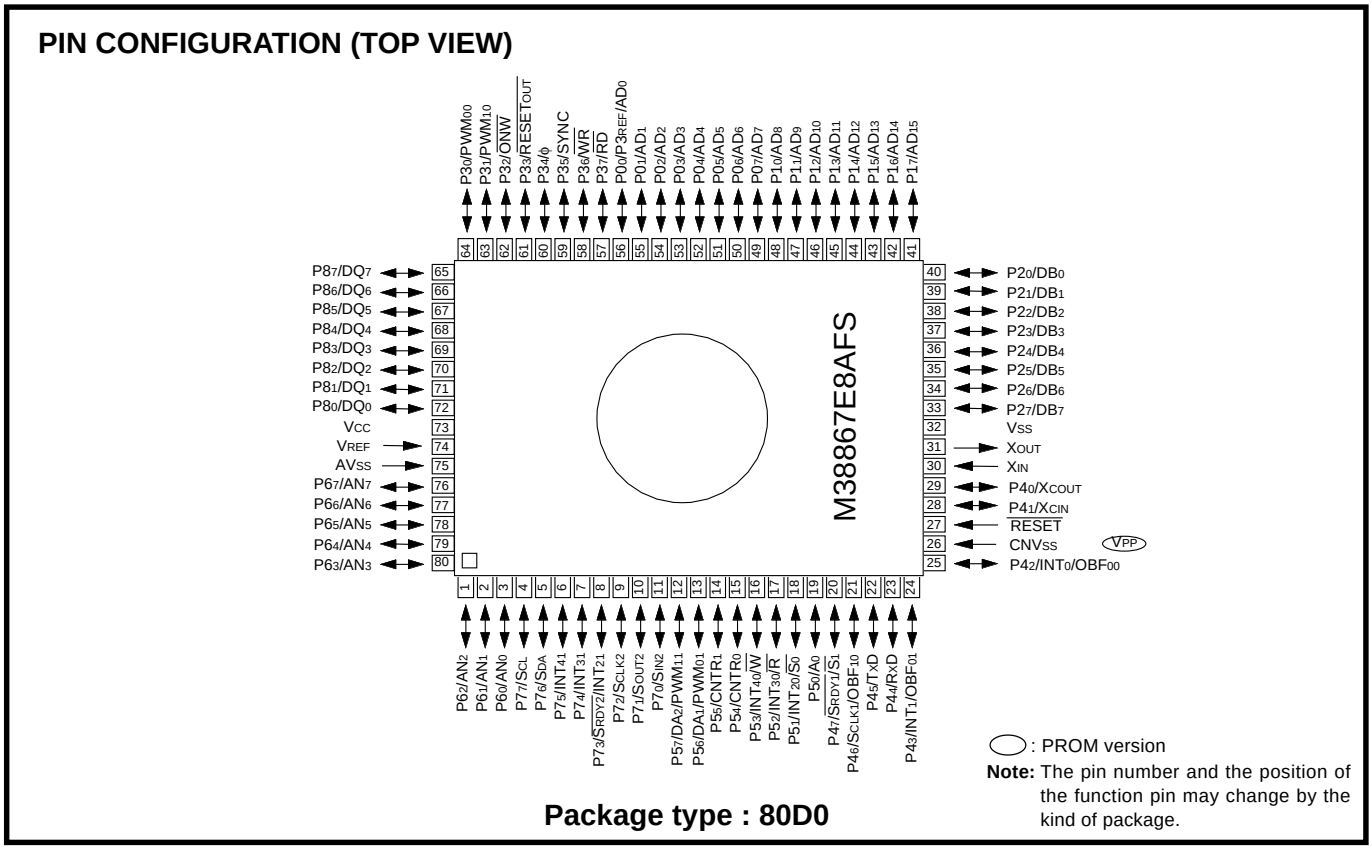


Fig. 3.10.2 M38867E8AFS pin configuration

APPENDIX

3.10 Pin configurations

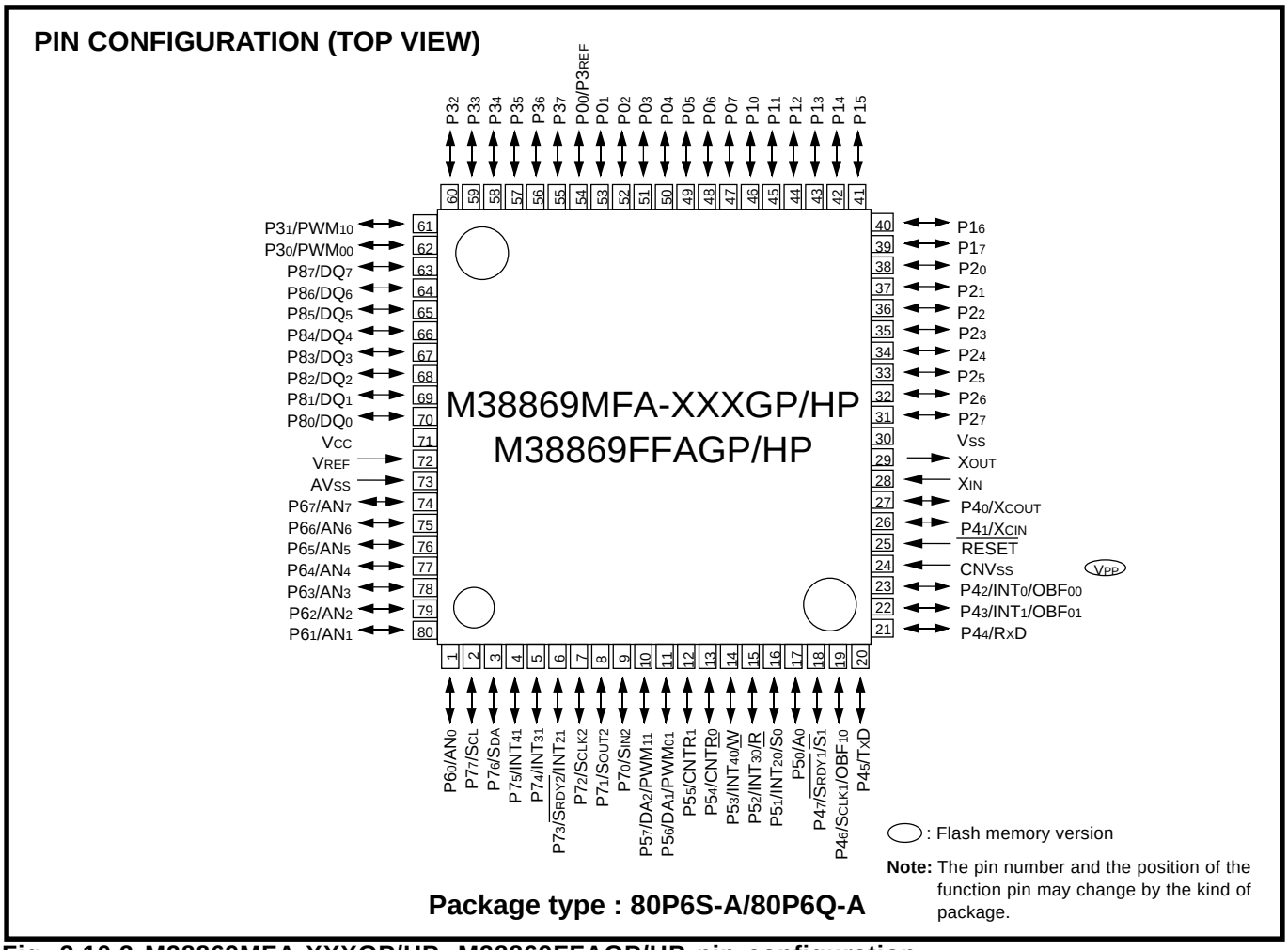


Fig. 3.10.3 M38869MFA-XXXGP/HP, M38869FFAGP/HP pin configuration

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
3886 Group

Sep. Second Edition 2000

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

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

User's Manual
3886 Group

Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

New publication, effective Sep. 2000.
Specifications subject to change without notice.