To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

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

User's Manual

8

3822 Group

User's Manual

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

Renesas Electronics
www.renesas.com

keep safety first in your circuit designs !

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

Notes regarding these materials

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

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

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

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

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

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

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

Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 3822 Group.
After reading this manual, the user should have a through knowledge of the functions and features of the 3822 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 **"SERIES 740 <SOFTWARE> USER'S MANUAL."**

BEFORE USING THIS USER'S MANUAL

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

1. Organization

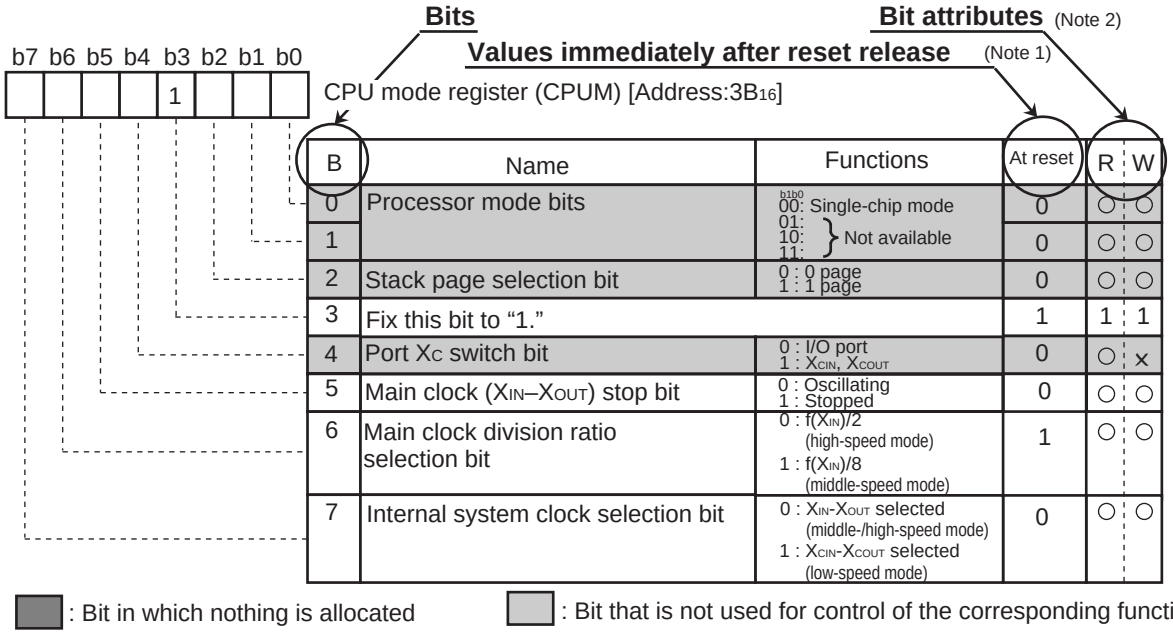
● CHAPTER 1 HARDWARE
This chapter describes features of the microcomputer, operation of each peripheral function and electric characteristics.

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

● CHAPTER 3 APPENDIX
This chapter includes precautions for systems development using the microcomputer, a list of control registers, the masking confirmation forms (mask ROM version), ROM programming confirmation forms (One Time PROM version) and mark specification forms which are to be submitted when ordering.

2. Structure of register

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



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

Notes 1: Values immediately after reset release
0••••••"0" at reset release
1••••••"1" at reset release
?••••••Undefined or reset release

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

R••••••Read
○ ••••••Read enabled
× ••••••Read disabled
0 ••••••Fixed to "0"
1 ••••••Fixed to "1"

W••••••Write
○ ••••••Write enabled
× ••••••Write disabled
✻ ••••••Only "0" write enabled
0 ••••••Fix to "0"
1 ••••••Fix to "1"

Table of contents

CHAPTER 1. HARDWARE

CHAPTER 2. APPLICATION

Table of contents

CHAPTER 3. APPENDIX

List of figures

CHAPTER 1. HARDWARE

CHAPTER 2. APPLICATION

List of figures

CHAPTER 3. APPENDIX

List of tables

CHAPTER 1. HARDWARE

CHAPTER 2. APPLICATION

List of tables

CHAPTER 3. APPENDIX

CHAPTER 1

HARDWARE

3822 Group

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

DESCRIPTION

The 3822 group is the 8-bit microcomputer based on the 740 family core technology.

The 3822 group has the LCD drive control circuit an 8-channel A-D converter, and a Serial I/O as additional functions.

The various microcomputers in the 3822 group include variations of internal memory size and packaging. For details, refer to the section on part numbering.

For details on availability of microcomputers in the 3822 group, refer to the section on group expansion.

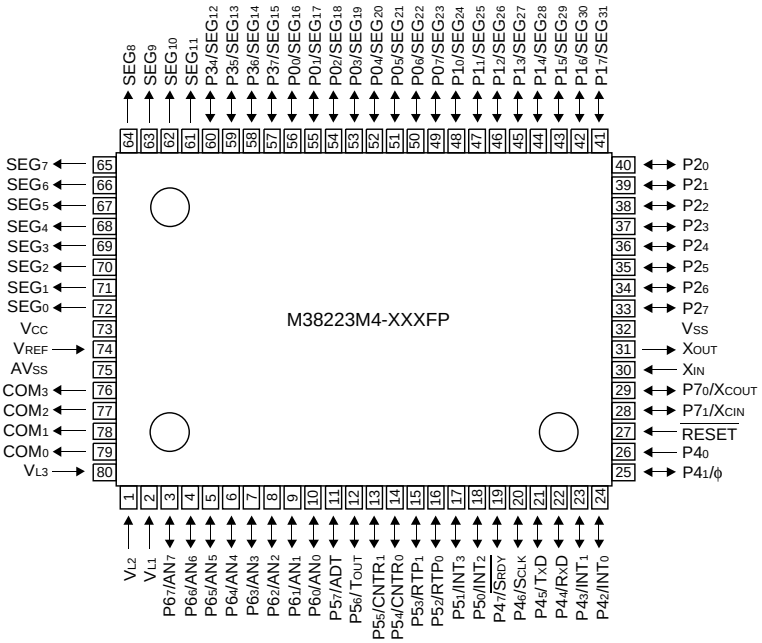
FEATURES

- Basic machine-language instructions 71
- The minimum instruction execution time ..... 0.5 µs
  (at 8MHz oscillation frequency)
- Memory size
  ROM ..... 4 K to 32 K bytes
  RAM ..... 192 to 1024 bytes
- Programmable input/output ports 49
- Software pull-up/pull-down resistors (Ports P0-P7 except Port P4$_0$)
- Interrupts ..... 17 sources, 16 vectors
  (includes key input interrupt)
- Timers ..... 8-bit × 3, 16-bit × 2
- Serial I/O1 ..... 8-bit × 1 (UART or Clock-synchronized)
- Serial I/O2 ..... 8-bit × 8 channels
- LCD drive control circuit
  Bias 1/2, 1/3
  Duty 1/2, 1/3, 1/4
  Common output 4
  Segment output 32
- 2 Clock generating circuit
  Clock (XIN-XOUT) ..... Internal feedback resistor
  Sub-clock (XCIN-XCOUT) ..... Without internal feedback resistor
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage
  In high-speed mode ..... 4.0 to 5.5 V
  (at 8MHz oscillation frequency and high-speed selected)
  In middle-speed mode ..... 2.5 to 5.5 V
  (at 8MHz oscillation frequency and middle-speed selected)
  In low-speed mode ..... 2.5 to 5.5 V
  (Extended operating temperature version: 3.0 V to 5.5 V)
- Power dissipation
  In high-speed mode ..... 32 mW
  (at 8 MHz oscillation frequency)
  In low-speed mode ..... 45 µW
  (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Operating temperature range ..... – 20 to 85°C
  (Extended operating temperature version: –40 to 85°C)

APPLICATIONS

Camera, household appliances, consumer electronics, etc.

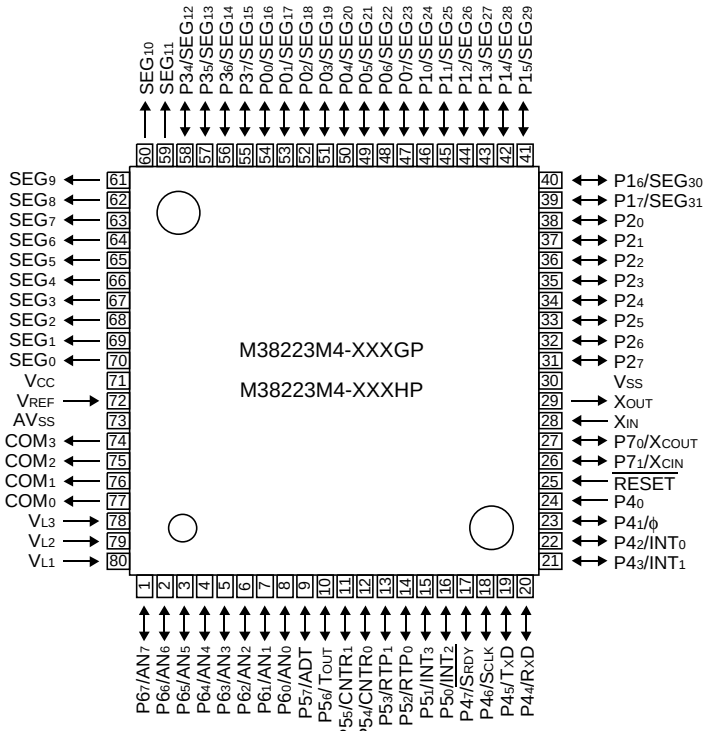
PIN CONFIGURATION (TOP VIEW)



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

Fig. 1 Pin configuration of M38223M4-XXXFP

PIN CONFIGURATION (TOP VIEW)



Package type : 80P6S-A/80P6D-A
80-pin plastic-molded QFP

Fig. 2 Pin configuration of M38223M4-XXXGP/HP

FUNCTIONAL BLOCK DIAGRAM (Package : 80P6S-A)

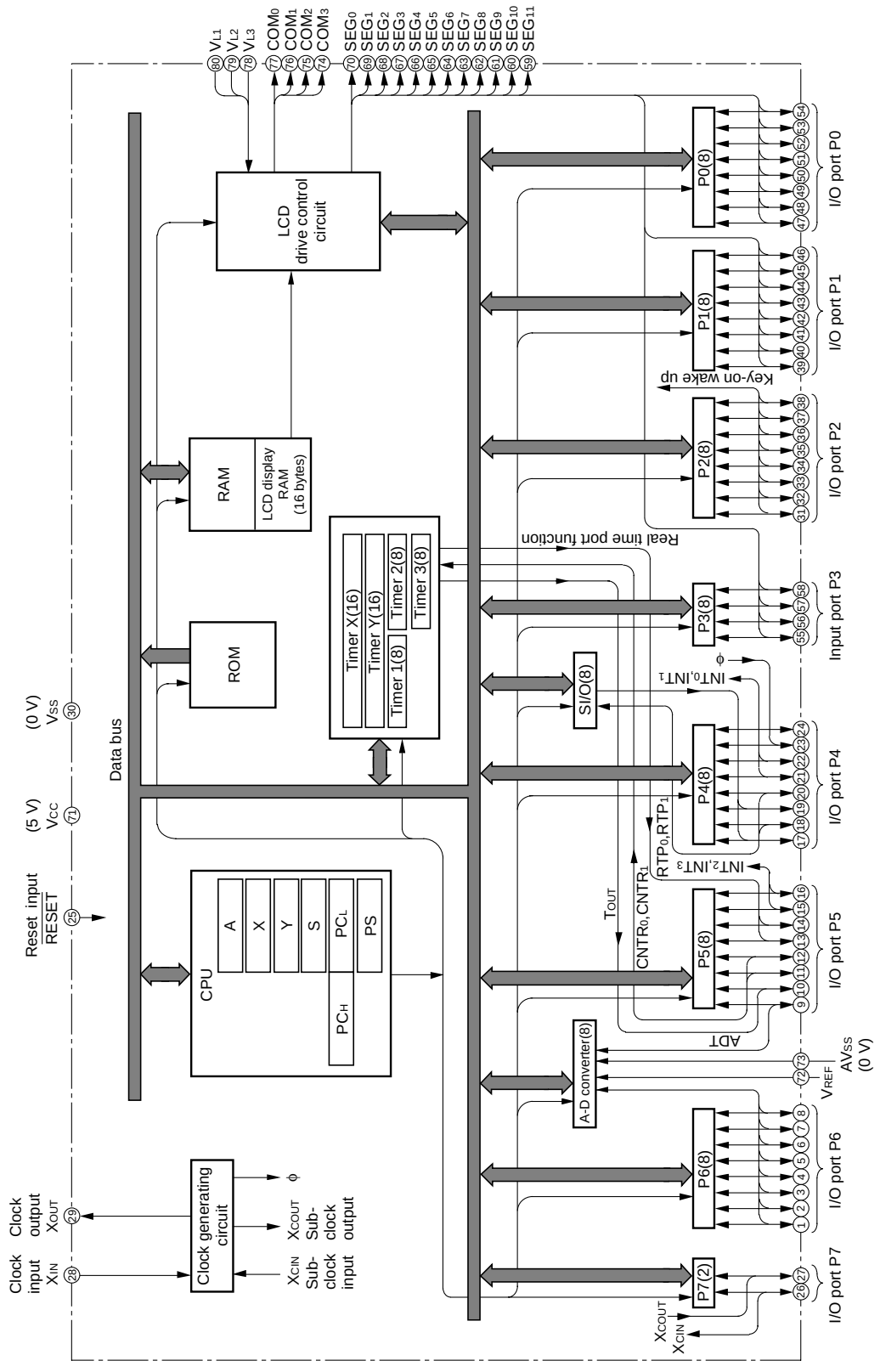


Fig. 3 Functional block diagram

PIN DESCRIPTION

Pin	Name	Function	Function except a port function
V_{CC}, V_{SS}	Power source	• Apply voltage of 2.5 V to 5.5 V to V_{CC}, and 0 V to V_{SS}.	
V_{REF}	Analog reference voltage	• Reference voltage input pin for A-D converter.	
AV_{SS}	Analog power source	• GND input pin for A-D converter. • Connect to V_{SS}.	
$\overline{RESET}$	Reset input	•Reset input pin for active “L”	
X_{IN}	Clock input	• Input and output pins for the main clock generating circuit. • Feedback resistor is built in between X_{IN} pin and X_{OUT} pin. • Connect a ceramic resonator or a quartz-crystal oscillator between the X_{IN} and X_{OUT} pins to set the oscillation frequency. • If an external clock is used, connect the clock source to the X_{IN} pin and leave the X_{OUT} pin open. • This clock is used as the oscillating source of system clock.	
X_{OUT}	Clock output		
V_{L1} – V_{L3}	LCD power source	• Input 0 ≤ V_{L1} ≤ V_{L2} ≤ V_{L3} ≤ V_{CC} voltage • Input 0 – V_{L3} voltage to LCD	
COM_0 – COM_3	Common output	• LCD common output pins • COM_2 and COM_3 are not used at 1/2 duty ratio. • COM_3 is not used at 1/3 duty ratio.	
SEG_0 – SEG_{11}	Segment output	• LCD segment output pins	
$P0_0/SEG_{16}$ – $P0_7/SEG_{23}$	I/O port P0	• 8-bit I/O port • CMOS compatible input level • CMOS 3-state output structure • I/O direction register allows each port to be individually programmed as either input or output. • Pull-down control is enabled.	• LCD segment pins
$P1_0/SEG_{24}$ – $P1_7/SEG_{31}$	I/O port P1		
$P2_0$ – $P2_7$	I/O port P2	• 8-bit I/O port • CMOS compatible input level • CMOS 3-state output structure • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• Key input (key-on wake up) interrupt input pins
$P3_0/SEG_{12}$ – $P3_7/SEG_{15}$	Input port P3	• 4-bit Input port • CMOS compatible input level • Pull-down control is enabled.	• LCD segment pins
$P4_0$	Input port P4	• 1-bit input pin • CMOS compatible input level	
$P4_1/\phi$	I/O port P4	• 7-bit I/O port • CMOS compatible input level • CMOS 3-state output structure • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• ϕ clock output pin
$P4_2/INT_0$, $P4_3/INT_1$			• Interrupt input pins
$P4_4/R_XD$, $P4_5/T_XD$, $P4_6/\overline{S_{CLK}}$, $P4_7/\overline{S_{RDY}}$			• Serial I/O1 function pins

PIN DESCRIPTION

Pin	Name	Function	Function except a port function
P5$_0$/INT$_2$, P5$_1$/INT$_3$	I/O port P5	• 8-bit I/O port • CMOS compatible input level • CMOS 3-state output structure • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• Interrupt input pins
P5$_2$/RTP$_0$, P5$_3$/RTP$_1$			• Real time port function pins
P5$_4$/CNTR$_0$, P5$_5$/CNTR$_1$			• Timer function pins
P5$_6$/TOUT			• Timer output pin
P5$_7$/ADT			• A-D trigger input pin
P6$_0$/AN$_0$-P6$_7$/AN$_7$	I/O port P6	• 8-bit I/O port • CMOS compatible input level • CMOS 3-state output structure • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• A-D conversion input pins
P7$_0$/XCOUT, P7$_1$/XCIN	I/O port P7	• 2-bit I/O port • CMOS compatible input level • CMOS 3-state output structure • I/O direction register allows each pin to be individually programmed as either input or output. • Pull-up control is enabled.	• Sub-clock generating circuit I/O pins (Connect a resonator. External clock cannot be used.)

PART NUMBERING

Product M3822 **3 M 4 - XXX FP**

Package type
FP : 80P6N-A package
GP : 80P6S-A package
HP : 80P6D-A package
FS : 80D0 package

ROM number
Omitted in some types.

Normally, using hyphen
When electrical characteristic, or division of quality identification code using alphanumeric character
– : Standard
D : Extended operating temperature version

ROM/PROM size
1 : 4096 bytes
2 : 8192 bytes
3 : 12288 bytes
4 : 16384 bytes
5 : 20480 bytes
6 : 24576 bytes
7 : 28672 bytes
8 : 32768 bytes

The first 128 bytes and the last 2 bytes of ROM are reserved areas ; they cannot be used.

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

RAM size
0 : 192 bytes
1 : 256 bytes
2 : 384 bytes
3 : 512 bytes
4 : 640 bytes
5 : 768 bytes
6 : 896 bytes
7 : 1024 bytes

Fig. 4 Part numbering

GROUP EXPANSION

Mitsubishi plans to expand the 3822 group as follows:

(1) Support for mask ROM, One Time PROM, and EPROM versions

(2) ROM/PROM size ..... 8 K to 16 K bytes
RAM size ..... 384 to 512 bytes

(3) Packages
80P6N-A ..... 0.8 mm-pitch plastic molded QFP
80P6S-A ..... 0.65 mm-pitch plastic molded QFP
80P6D-A ..... 0.5 mm-pitch plastic molded QFP
80D0 ..... 0.8 mm-pitch ceramic LCC (EPROM version)

Memory Expansion Plan

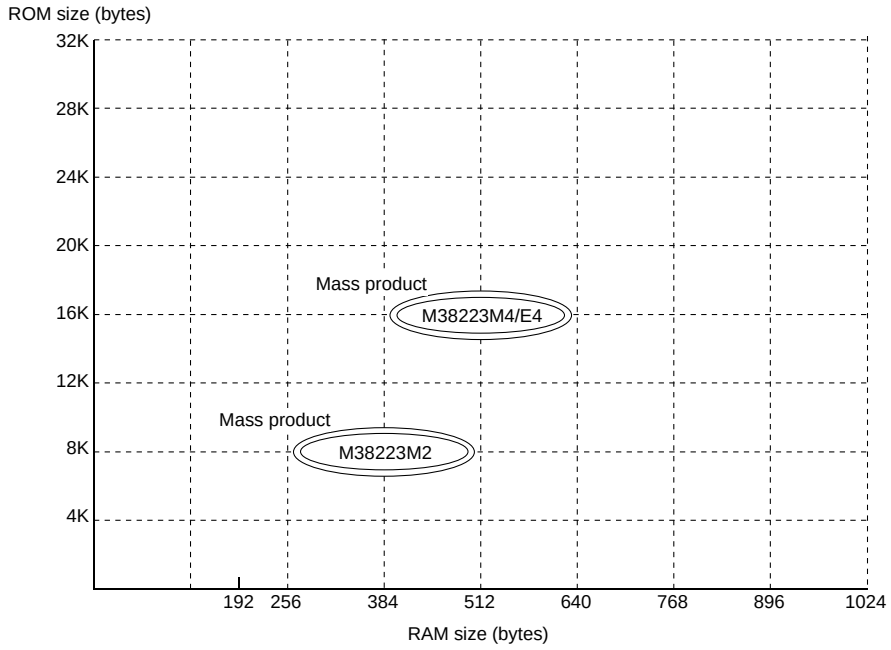


Fig. 5 Memory expansion plan (1)

Currently supported products are listed below. **As of May 1996**

Product	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38223M4-XXXFP	16384 (16254)	512	80P6N-A	Mask ROM version
M38223E4-XXXFP				One Time PROM version
M38223E4FP				One Time PROM version (blank)
M38223M4-XXXGP			80P6S-A	Mask ROM version
M38223E4-XXXGP				One Time PROM version
M38223E4GP				One Time PROM version (blank)
M38223M4-XXXHP			80P6D-A	Mask ROM version
M38223E4-XXXHP				One Time PROM version
M38223E4HP				One Time PROM version (blank)
M38223E4FS			80D0	EPROM version
M38222M2-XXXFP	8192 (8062)	384	80P6N-A	Mask ROM version
M38222M2-XXXGP			80P6S-A	
M38222M2-XXXHP			80P6D-A	

GROUP EXPANSION (EXTENDED OPERATING TEMPERATURE VERSION)

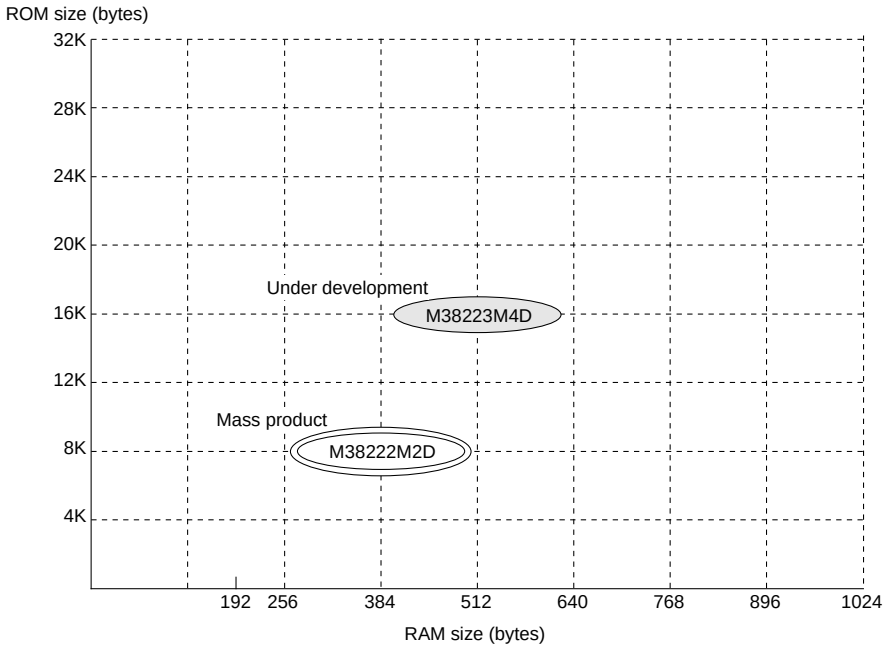
Mitsubishi plans to expand the 3822 group (extended operating temperature version) as follows:

(1) Support for mask ROM, One Time PROM, and EPROM versions

(2) ROM size ..... 16 K bytes
RAM size ..... 512 bytes

(3) Packages
80P6N-A ..... 0.8 mm-pitch plastic molded QFP

Memory Expansion Plan



Products under development: the development schedule and specification may be revised without notice.

Fig. 6 Memory expansion plan (2)

Currently supported products are listed below. **As of May 1996**

Product	ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38223M4DXXXFP	16384(16254)	512	80P6N-A	Mask ROM version
M38222M2DXXXGP	8192(8062)	384	80P6S-A	Mask ROM version

FUNCTIONAL DESCRIPTION

Central Processing Unit (CPU)

The 3822 group uses the standard 740 family instruction set. Refer to the table of 740 family addressing modes and machine instructions or the SERIES 740 <Software> User's Manual for details on the instruction set.

Machine-resident 740 family instructions are as follows:

The FST and SLW instruction cannot be used.

The STP, WIT, MUL, and DIV instruction can be used.

CPU Mode Register

The CPU mode register is allocated at address 003B16.

The CPU mode register contains the stack page selection bit and the internal system clock selection bit.

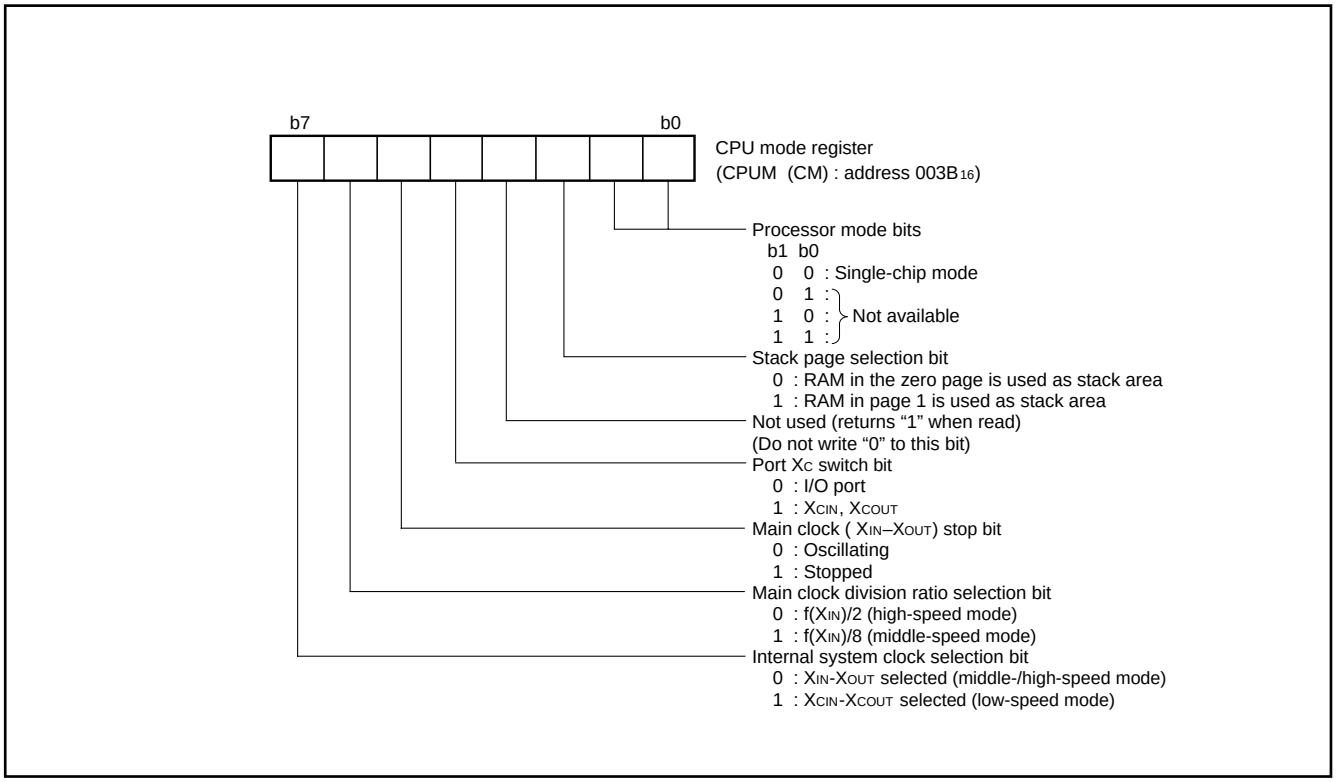


Fig. 7 Structure of CPU mode register

MEMORY

Special Function Register (SFR) Area

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

RAM

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

ROM

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

Interrupt Vector Area

The interrupt vector area contains reset and interrupt vectors.

Zero Page

The 256 bytes from addresses 0000_{16} to $00FF_{16}$ are called the zero page area. The internal RAM and the special function registers (SFR) are allocated to this area.

The zero page addressing mode can be used to specify memory and register addresses in the zero page area. Access to this area with only 2 bytes is possible in the zero page addressing mode.

Special Page

The 256 bytes from addresses $FF00_{16}$ to $FFFF_{16}$ are called the special page area. The special page addressing mode can be used to specify memory addresses in the special page area. Access to this area with only 2 bytes is possible in the special page addressing mode.

RAM area

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

ROM area

ROM size (bytes)	Address $YYYY_{16}$	Address $ZZZZ_{16}$
4096	$F000_{16}$	$F080_{16}$
8192	$E000_{16}$	$E080_{16}$
12288	$D000_{16}$	$D080_{16}$
16384	$C000_{16}$	$C080_{16}$
20480	$B000_{16}$	$B080_{16}$
24576	$A000_{16}$	$A080_{16}$
28672	9000_{16}	9080_{16}
32768	8000_{16}	8080_{16}

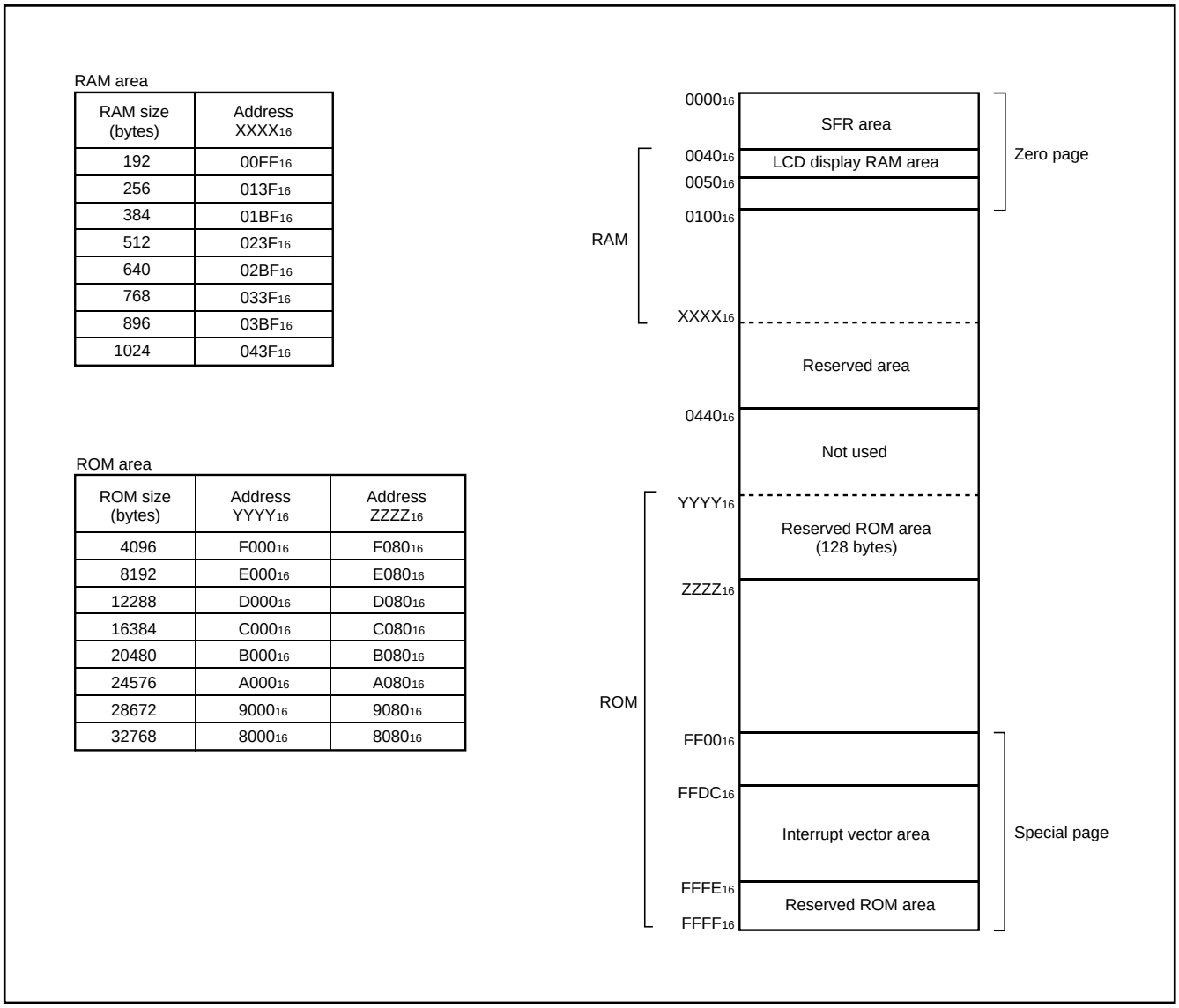


Fig. 8 Memory map diagram

0000_{16}	Port P0 (P0)	0020_{16}	Timer X (low) (TXL)
0001_{16}	Port P0 direction register (P0D)	0021_{16}	Timer X (high) (TXH)
0002_{16}	Port P1 (P1)	0022_{16}	Timer Y (low) (TYL)
0003_{16}	Port P1 direction register (P1D)	0023_{16}	Timer Y (high) (TYH)
0004_{16}	Port P2 (P2)	0024_{16}	Timer 1 (T1)
0005_{16}	Port P2 direction register (P2D)	0025_{16}	Timer 2 (T2)
0006_{16}	Port P3 (P3)	0026_{16}	Timer 3 (T3)
0007_{16}		0027_{16}	Timer X mode register (TXM)
0008_{16}	Port P4 (P4)	0028_{16}	Timer Y mode register (TYM)
0009_{16}	Port P4 direction register (P4D)	0029_{16}	Timer 123 mode register (T123M)
$000A_{16}$	Port P5 (P5)	$002A_{16}$	ϕ output control register (CKOUT)
$000B_{16}$	Port P5 direction register (P5D)	$002B_{16}$	
$000C_{16}$	Port P6 (P6)	$002C_{16}$	
$000D_{16}$	Port P6 direction register (P6D)	$002D_{16}$	
$000E_{16}$	Port P7 (P7)	$002E_{16}$	
$000F_{16}$	Port P7 direction register (P7D)	$002F_{16}$	
0010_{16}		0030_{16}	
0011_{16}		0031_{16}	
0012_{16}		0032_{16}	
0013_{16}		0033_{16}	
0014_{16}		0034_{16}	A-D control register (ADCON)
0015_{16}		0035_{16}	A-D conversion register (AD)
0016_{16}	PULL register A (PULLA)	0036_{16}	
0017_{16}	PULL register B (PULLB)	0037_{16}	
0018_{16}	Transmit/Receive buffer register (TB/RB)	0038_{16}	Segment output enable register (SEG)
0019_{16}	Serial I/O1 status register (SIO1STS)	0039_{16}	LCD mode register (LM)
$001A_{16}$	Serial I/O1 control register (SIO1CON)	$003A_{16}$	Interrupt edge selection register (INTEDGE)
$001B_{16}$	UART control register (UARTCON)	$003B_{16}$	CPU mode register (CPUM)
$001C_{16}$	Baud rate generator (BRG)	$003C_{16}$	Interrupt request register 1(IREQ1)
$001D_{16}$		$003D_{16}$	Interrupt request register 2(IREQ2)
$001E_{16}$		$003E_{16}$	Interrupt control register 1(ICON1)
$001F_{16}$		$003F_{16}$	Interrupt control register 2(ICON2)

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

I/O PORTS

Direction Registers (ports P2, P41–P47, and P5–P7)

The 3822 group has 49 programmable I/O pins arranged in seven I/O ports (ports P0–P2 and P41–P47 and P5–P7). The I/O ports P2, P41–P47, and P5–P7 have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, each pin can be set to be input port or output port.

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

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

Direction Registers (ports P0 and P1)

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

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

When "0" is written to the bit 0 of a direction register, that port becomes an input port. When "1" is written to that port, that port becomes an output port.

Bits 1 to 7 of ports P0 and P1 direction registers are not used.

Ports P3 and P40

These ports are only for input.

Pull-up/Pull-down Control

By setting the PULL register A (address 0016_{16}) or the PULL register B (address 0017_{16}), ports except for port $P4_0$ can control either pull-down or pull-up (pins that are shared with the segment output pins for LCD are pull-down; all other pins are pull-up) with a program.

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

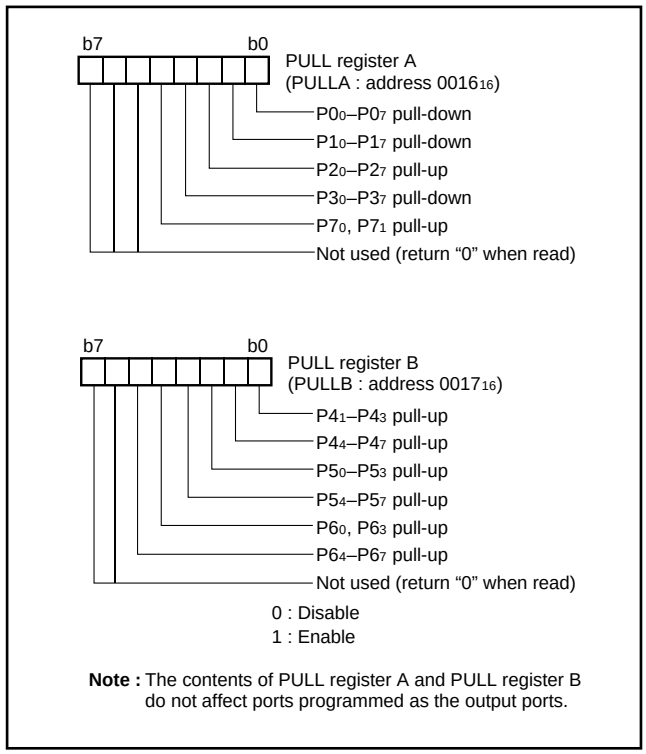


Fig. 10 Structure of PULL register A and PULL register B

Pin	Name	Input/Output	I/O Format	Non-Port Function	Related SFRs	Diagram No.
P00/SEG16–P07/SEG23	Port P0	Input/output, individual ports	CMOS compatible input level CMOS 3-state output	LCD segment output	PULL register A Segment output enable register	(1)
P10/SEG24–P17/SEG31	Port P1	Input/output, individual ports	CMOS compatible input level CMOS 3-state output	LCD segment output	PULL register A Segment output enable register	
P20 – P27	Port P2	Input/output, individual bits	CMOS compatible input level CMOS 3-state output	Key input(Key-on wake up) interrupt input	PULL register A Interrupt control register 2	(2)
P34/SEG12–P37/SEG15	Port P3	Input	CMOS compatible input level	LCD segment output	PULL register A Segment output enable register	(3)
P40	Port P4					(4)
P41/ϕ		Input/output, individual bits	CMOS compatible input level CMOS 3-state output	ϕ clock output	PULL register B ϕ output control register	(5)
P42/INT0, P43/INT1				External interrupt input	PULL register B Interrupt edge selection register	(2)
P44/RxD				Serial I/O function I/O	PULL register B	(6)
P45/TxD					Serial I/O control register	(7)
P46/SCLK1					Serial I/O status register	(8)
P47/$\overline{SRDY}$					UART control register	(9)
P50/INT2, P51/INT3	Port P5			External interrupt input Real time port function oputput	PULL register B Interrupt edge selection register	(2)
P52/RTP0, P53/RTP1						(10)
P54/CNTR0				Timer I/O	PULL register B Timer X mode register	(11)
P55/CNTR1				Timer I/O	PULL register B Timer X mode register	(12)
P56/TOUT				Timer output	PULL register B Timer Y mode register	(13)
P57/ADT				A-D trigger input	PULL register B Timer 123 mode register	(12)
P60/AN0–P67/AN7	Port P6			A-D conversion input	PULL register B A-D control register	(14)
P70/XCOUT	Port P7			Sub-clock generating circuit I/O	PULL register A	(15)
P71/XCIN					CPU mode register	(16)
COM0-COM3	Common	output	LCD common output		LCD mode register	(17)
SEG0-SEG11	Segment		LCD segment output		Segment output enable register	(18)

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

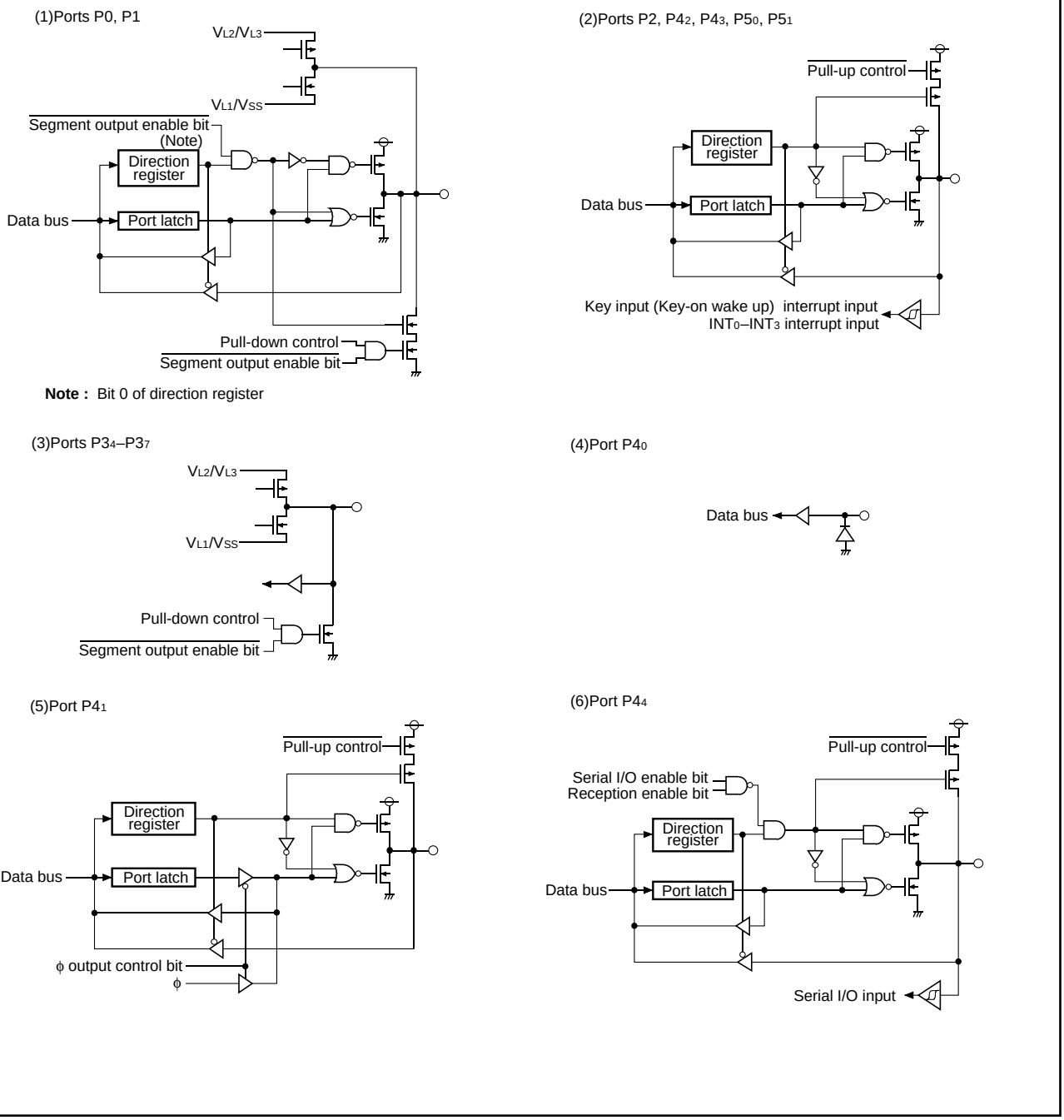


Fig. 11 Port block diagram (1)

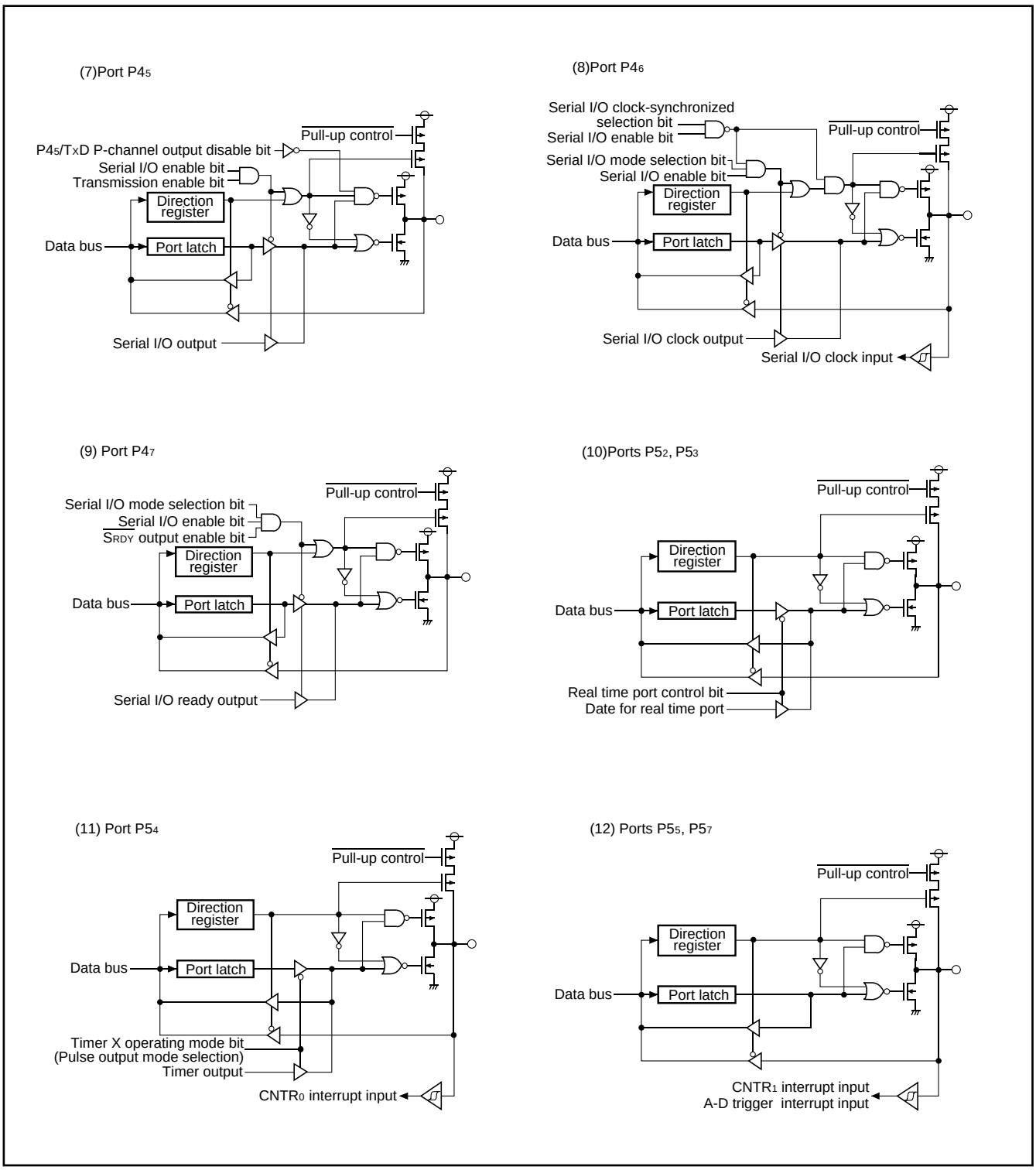


Fig. 12 Port block diagram (2)

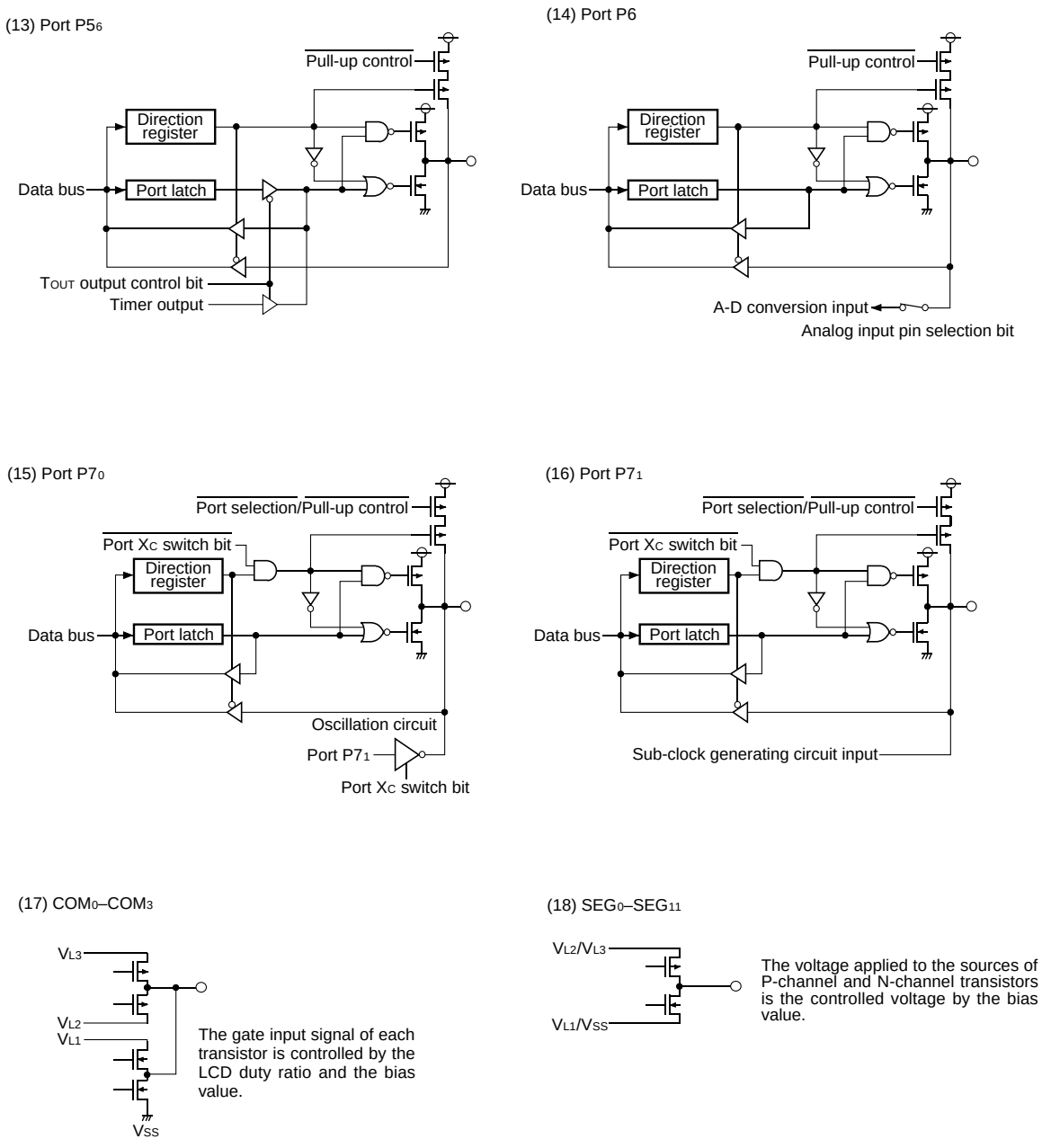


Fig. 13 Port block diagram (3)

INTERRUPTS

Interrupts occur by seventeen sources: eight external, eight 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 flag disables all interrupts except the BRK instruction interrupt.

Interrupt Operation

When an interrupt is received, the contents of the program counter and processor status register are automatically stored into the stack. The interrupt disable flag is set to inhibit other interrupts from interfering.The corresponding interrupt request bit is cleared and the interrupt jump destination address is read from the vector table into the program counter.

Notes on Use

When the active edge of an external interrupt (INT_0–INT_3, $CNTR_0$, or $CNTR_1$) is changed, the corresponding interrupt request bit may also be set. Therefore, please take following sequence;

(1) Disable the external interrupt which is selected.
(2) Change the active edge selection.
(3) Clear the interrupt request bit which is selected to “0”.
(4) Enable the external interrupt which is selected.

Table 1. Interrupt vector addresses and priority

Interrupt Source	Priority	Vector Addresses (Note 1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset (Note 2)	1	$FFFD_{16}$	$FFFC_{16}$	At reset	Non-maskable
INT_0	2	$FFFB_{16}$	$FFFA_{16}$	At detection of either rising or falling edge of INT_0 input	External interrupt (active edge selectable)
INT_1	3	$FFF9_{16}$	$FFF8_{16}$	At detection of either rising or falling edge of INT_1 input	External interrupt (active edge selectable)
Serial I/O reception	4	$FFF7_{16}$	$FFF6_{16}$	At completion of serial I/O data reception	Valid when serial I/O1 is selected
Serial I/O transmission	5	$FFF5_{16}$	$FFF4_{16}$	At completion of serial I/O transmit shift or when transmission buffer is empty	Valid when serial I/O1 is selected
Timer X	6	$FFF3_{16}$	$FFF2_{16}$	At timer X underflow	
Timer Y	7	$FFF1_{16}$	$FFF0_{16}$	At timer Y underflow	
Timer 2	8	$FFEF_{16}$	$FFEE_{16}$	At timer 2 underflow	
Timer 3	9	$FFED_{16}$	$FFEC_{16}$	At timer 3 underflow	
$CNTR_0$	10	$FFEB_{16}$	$FFEA_{16}$	At detection of either rising or falling edge of $CNTR_0$ input	External interrupt (active edge selectable)
$CNTR_1$	11	$FFE9_{16}$	$FFE8_{16}$	At detection of either rising or falling edge of $CNTR_1$ input	External interrupt (active edge selectable)
Timer 1	12	$FFE7_{16}$	$FFE6_{16}$	At timer 1 underflow	
INT_2	13	$FFE5_{16}$	$FFE4_{16}$	At detection of either rising or falling edge of INT_2 input	External interrupt (active edge selectable)
INT_3	14	$FFE3_{16}$	$FFE2_{16}$	At detection of either rising or falling edge of INT_3 input	External interrupt (active edge selectable)
Key input (Key-on wake up)	15	$FFE1_{16}$	$FFE0_{16}$	At falling of conjunction of input level for port P2 (at input mode)	External interrupt (valid when an “L” level is applied)
ADT	16	$FFDF_{16}$	$FFDE_{16}$	At falling of ADT input	Valid when ADT interrupt is selected External interrupt (valid at falling)
A-D conversion				At completion of A-D conversion	Valid when A-D interrupt is selected
BRK instruction	17	$FFDD_{16}$	$FFDC_{16}$	At BRK instruction execution	Non-maskable software interrupt

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

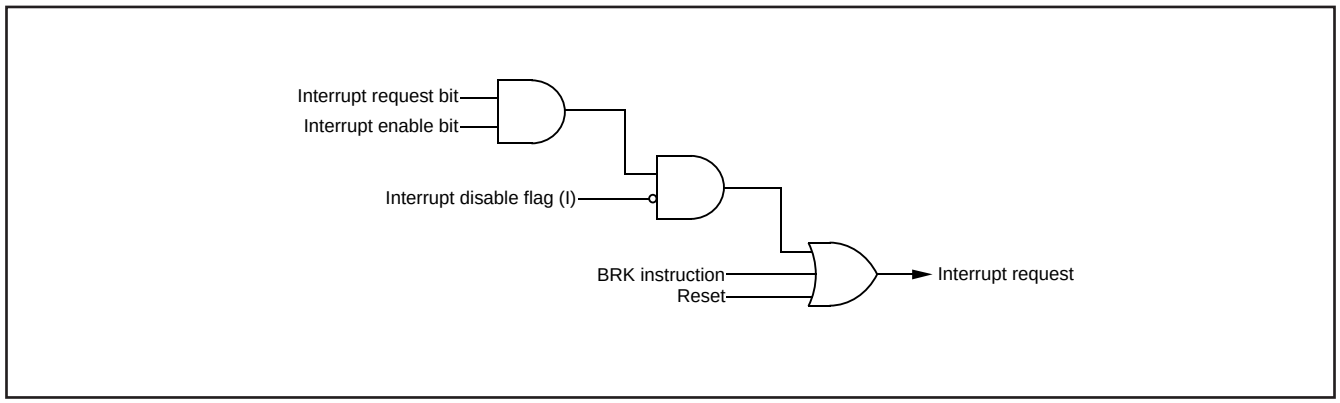


Fig. 14 Interrupt control

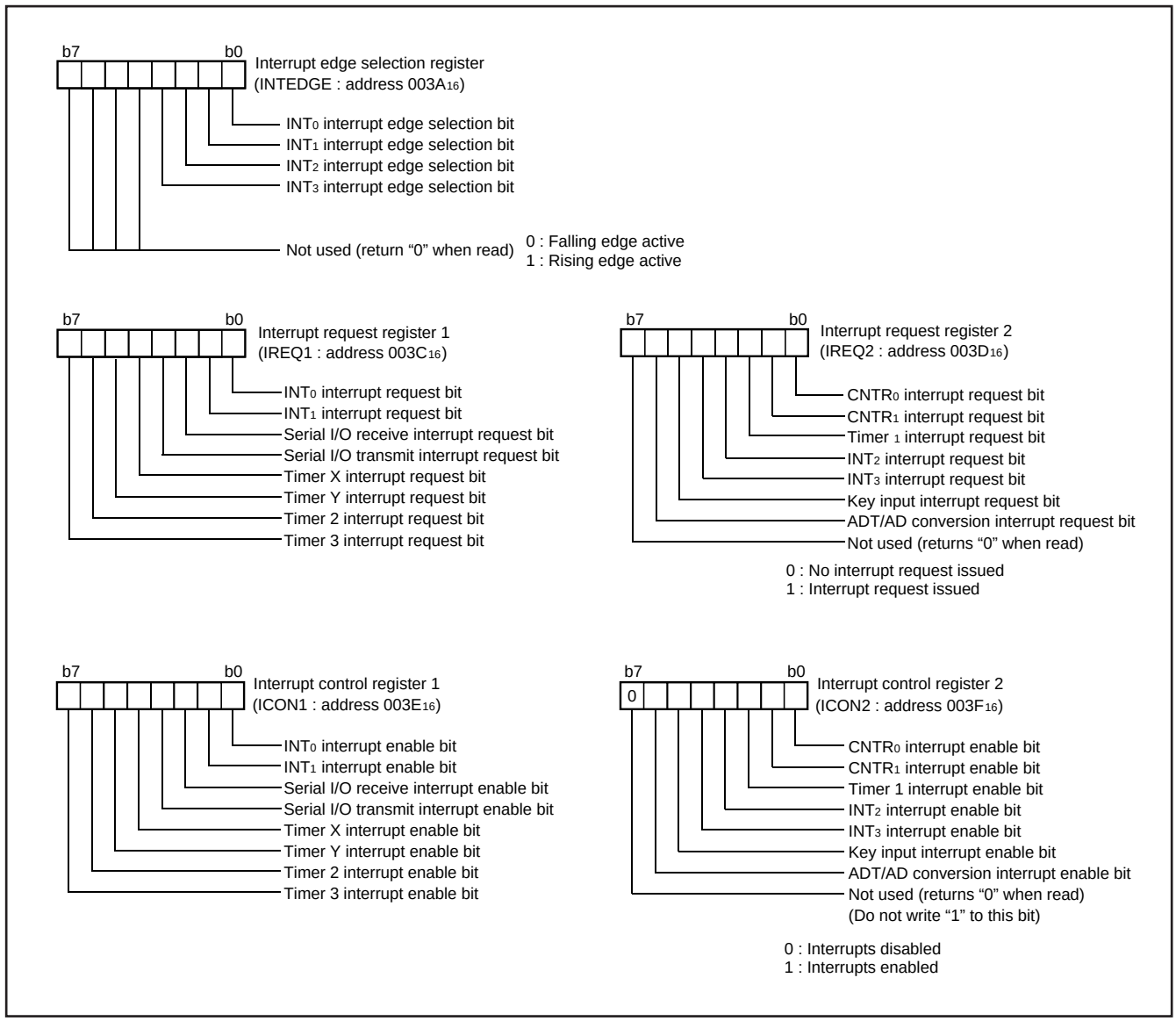


Fig. 15 Structure of interrupt-related registers

Key Input Interrupt (Key-on Wake Up)

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

An example of using a key input interrupt is shown in Figure 10, where an interrupt request is generated by pressing one of the keys consisted as an active-low key matrix which inputs to ports P20–P23.

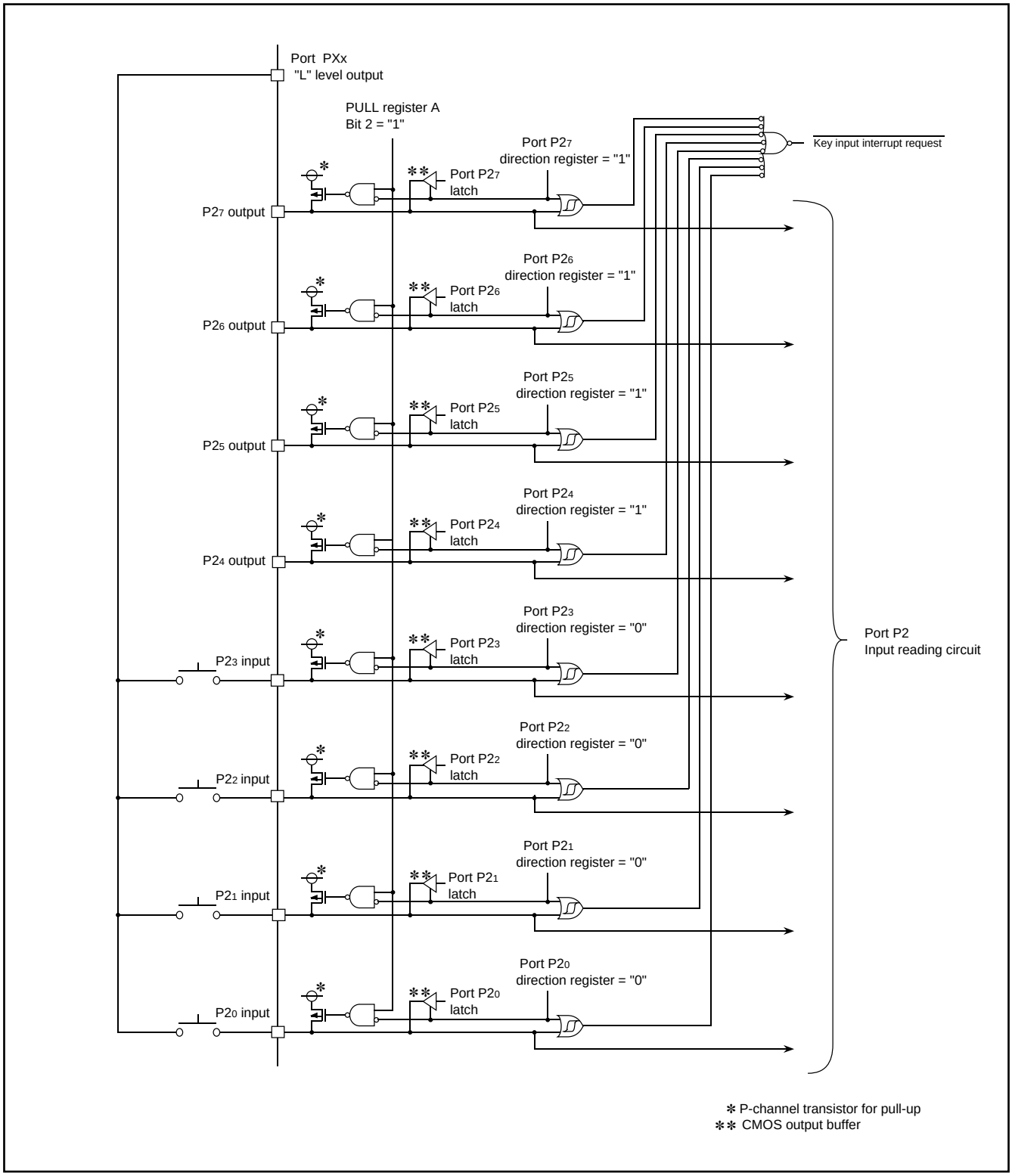


Fig. 16 Connection example when using key input interrupt and port P2 block diagram

TIMERS

The 3822 group has five timers: timer X, timer Y, timer 1, timer 2, and timer 3. Timer X and timer Y are 16-bit timers, and timer 1, timer 2, and timer 3 are 8-bit timers.

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

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

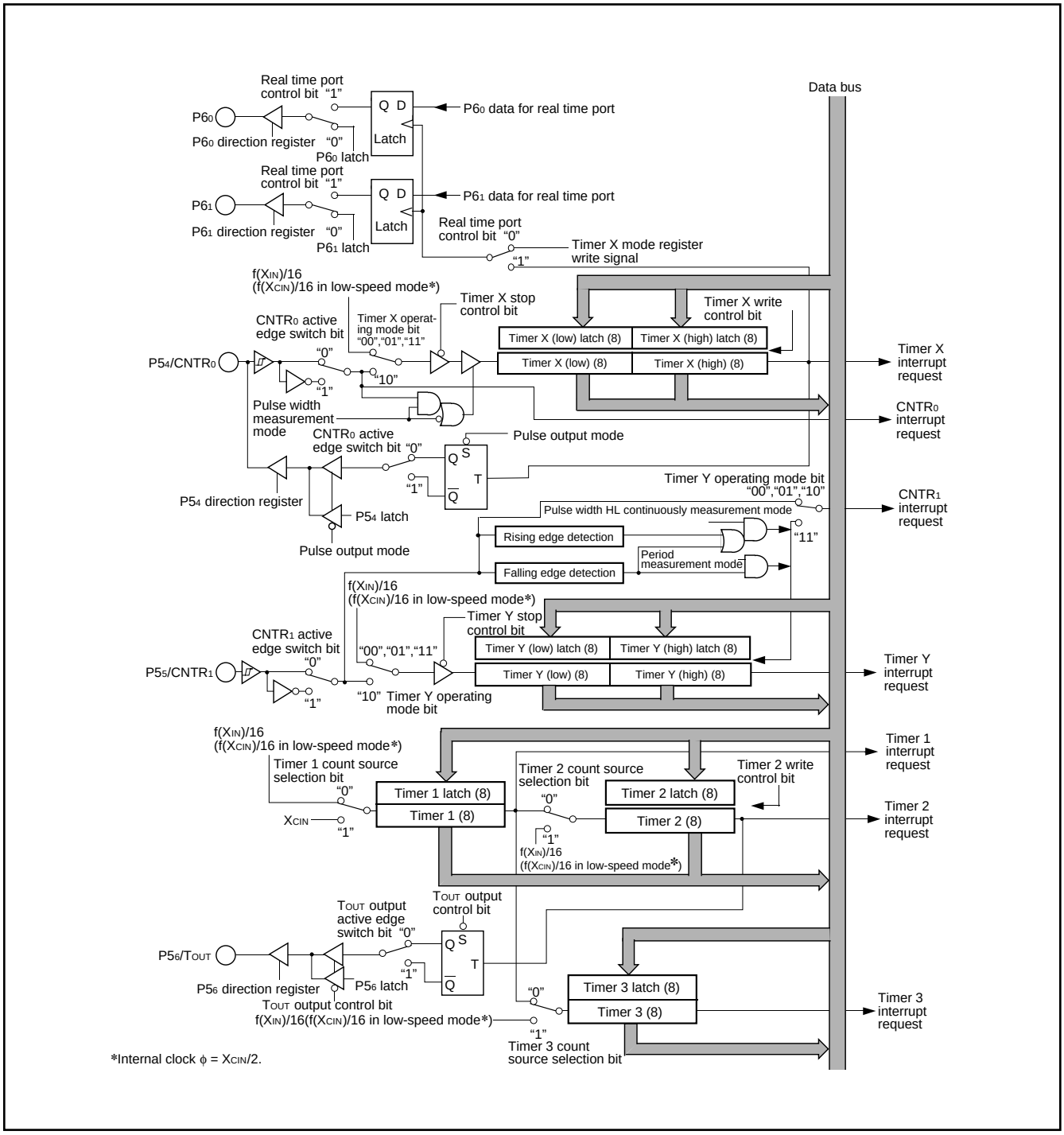


Fig. 17 Timer block diagram

Timer X

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

Timer mode

The timer counts f(XIN)/16 (or f(XCIN)/16 in low-speed mode).

Pulse output mode

Each time the timer underflows, a signal output from the CNTR0 pin is inverted. Except for this, the operation in pulse output mode is the same as in timer mode. When using a timer in this mode, set the corresponding port P54 direction register to output mode.

Event counter mode

The timer counts signals input through the CNTR0 pin.
Except for this, the operation in event counter mode is the same as in timer mode. When using a timer in this mode, set the corresponding port P54 direction register to input mode.

Pulse width measurement mode

The count source is f(XIN)/16 (or f(XCIN)/16 in low-speed mode). If CNTR0 active edge switch bit is “0”, the timer counts while the input signal of CNTR0 pin is at “H”. If it is “1”, the timer counts while the input signal of CNTR0 pin is at “L”. When using a timer in this mode, set the corresponding port P54 direction register to input mode.

Timer X Write Control

If the timer X write control bit is “0”, when the value is written in the address of timer X, the value is loaded in the timer X and the latch at the same time.
If the timer X write control bit is “1”, when the value is written in the address of timer X, the value is loaded only in the latch. The value in the latch is loaded in timer X after timer X underflows.
If the value is written in latch only, unexpected value may be set in the high-order counter when the writing in high-order latch and the underflow of timer X are performed at the same timing.

Note on CNTR0 Interrupt Active Edge Selection

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

Real Time Port Control

While the real time port function is valid, data for the real time port are output from ports P52 and P53 each time the timer X underflows. (However, after rewriting a data for real time port, if the real time port control bit is changed from “0” to “1”, data are output without the timer X.) If the data for the real time port is changed while the real time port function is valid, the changed data are output at the next underflow of timer X.
Before using this function, set the corresponding port direction registers to output mode.

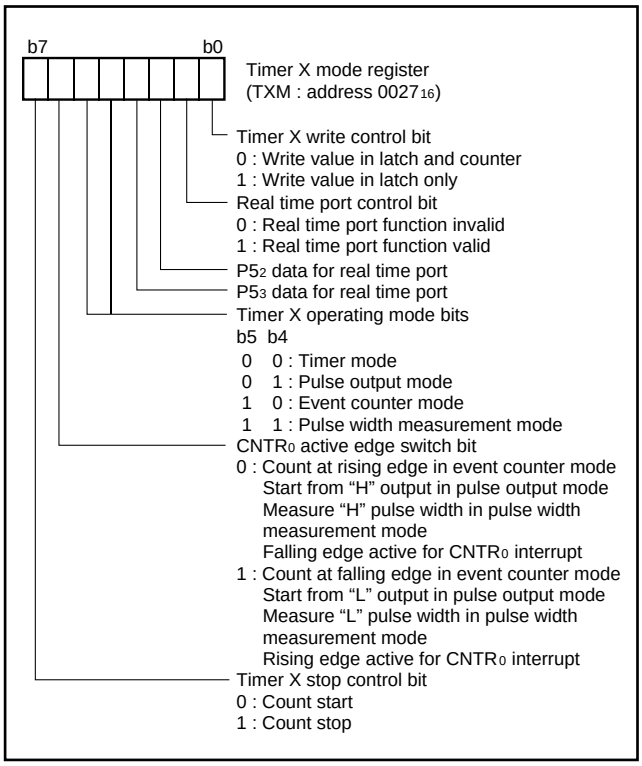


Fig. 18 Structure of timer X mode register

Timer Y

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

Timer mode

The timer counts f(XIN)/16 (or f(XCIN)/16 in low-speed mode).

Period measurement mode

CNTR1 interrupt request is generated at rising/falling edge of CNTR1 pin input signal. Simultaneously, the value in timer Y latch is reloaded in timer Y and timer Y continues counting down/Except for the above-mentioned, the operation in period measurement mode is the same as in timer mode.

The timer value just before the reloading at rising/falling of CNTR1 pin input signal is retained until the timer Y is read once after the reload.

The rising/falling timing of CNTR1 pin input signal is found by CNTR1 interrupt. When using a timer in this mode, set the corresponding port P55 direction register to input mode.

Event counter mode

The timer counts signals input through the CNTR1 pin.

Except for this, the operation in event counter mode is the same as in timer mode. When using a timer in this mode, set the corresponding port P55 direction register to input mode.

Pulse width HL continuously measurement mode

CNTR1 interrupt request is generated at both rising and falling edges of CNTR1 pin input signal. Except for this, the operation in pulse width HL continuously measurement mode is the same as in period measurement mode. When using a timer in this mode, set the corresponding port P55 direction register to input mode.

Note on CNTR1 Interrupt Active Edge Selection

CNTR1 interrupt active edge depends on the CNTR1 active edge switch bit. However, in pulse width HL continuously measurement mode, CNTR1 interrupt request is generated at both rising and falling edges of CNTR1 pin input signal regardless of the setting of CNTR1 active edge switch bit.

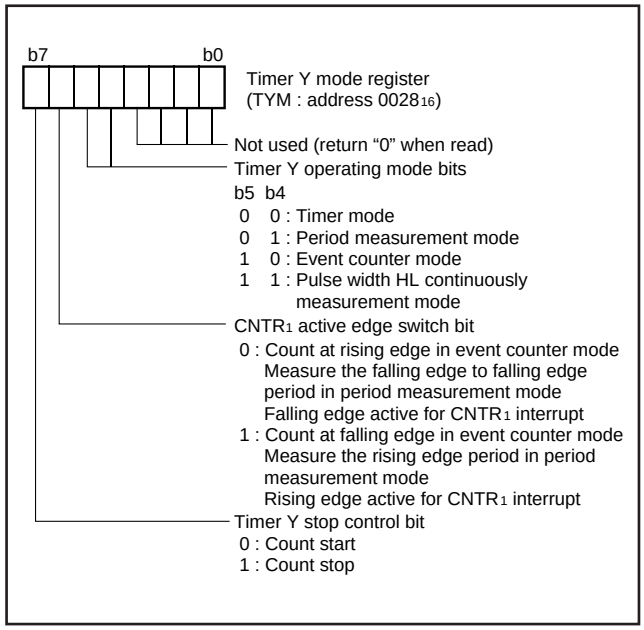


Fig. 19 Structure of timer Y mode register

Timer 1, Timer 2, Timer 3

Timer 1, timer 2, and timer 3 are 8-bit timers. The count source for each timer can be selected by timer 123 mode register. The timer latch value is not affected by a change of the count source. However, because changing the count source may cause an inadvertent count down of the timer. Therefore, rewrite the value of timer whenever the count source is changed.

Timer 2 Write Control

If the timer 2 write control bit is “0”, when the value is written in the address of timer 2, the value is loaded in the timer 2 and the latch at the same time.

If the timer 2 write control bit is “1”, when the value is written in the address of timer 2, the value is loaded only in the latch. The value in the latch is loaded in timer 2 after timer 2 underflows.

Timer 2 Output Control

When the timer 2 (TOUT) is output enabled, an inversion signal from pin TOUT is output each time timer 2 underflows.

In this case, set the port P56 shared with the port TOUT to the output mode.

Note on Timer 1 to Timer 3

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

Therefore, set the value of timer in the order of timer 1, timer 2 and timer 3 after the count source selection of timer 1 to 3.

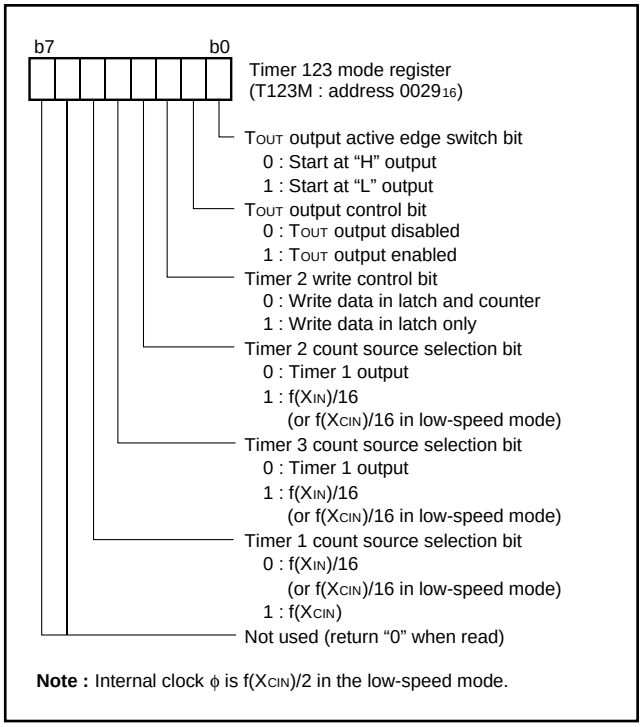


Fig. 20 Structure of timer 123 mode register

SERIAL I/O

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

Clock Synchronous Serial I/O Mode

Clock synchronous serial I/O mode can be selected by setting the mode selection bit of the serial I/O control register 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 (address 0018_{16}).

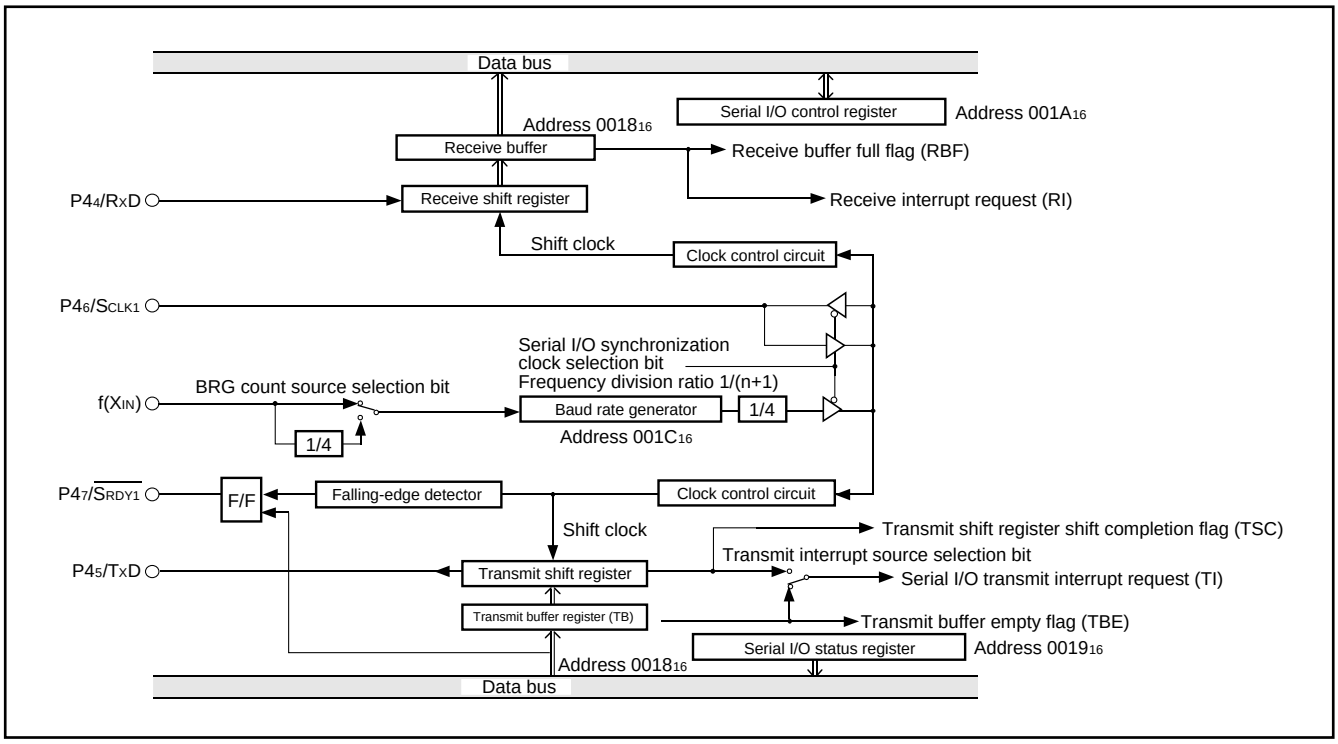
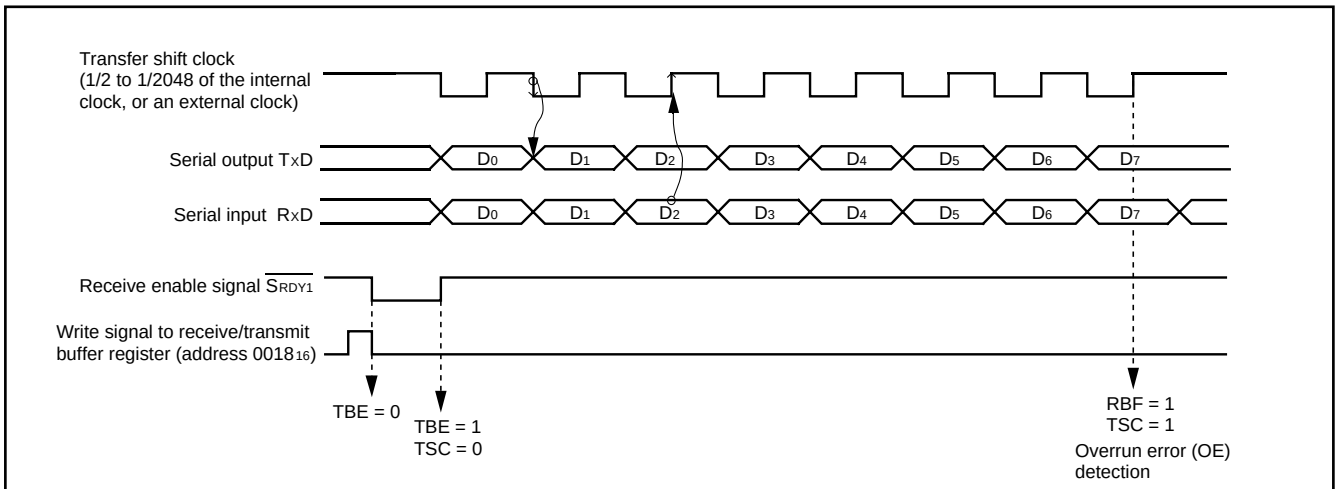


Fig. 21 Block diagram of clock synchronous serial I/O



Notes 1 : The transmit interrupt (TI) can be selected to occur either when the transmit buffer register has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O 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. 22 Operation of clock synchronous serial I/O function

Asynchronous Serial I/O1 (UART) Mode

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

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

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

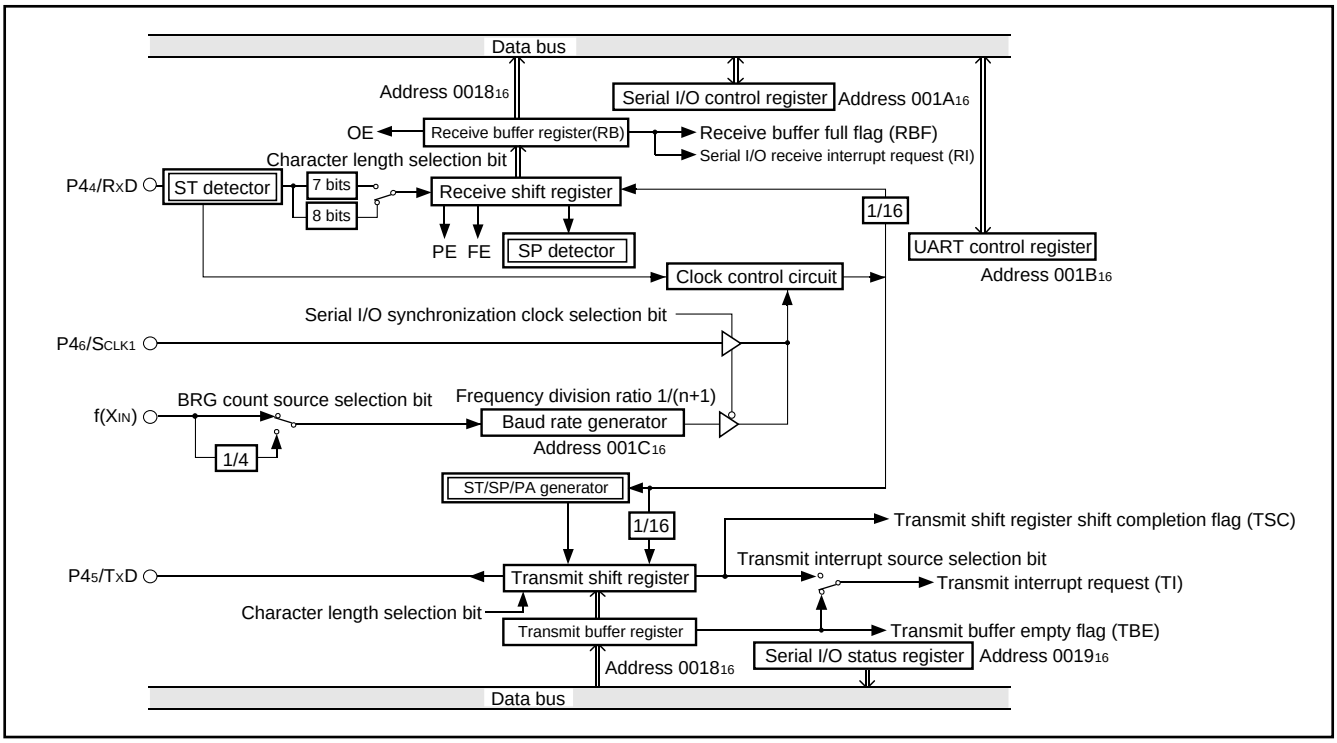
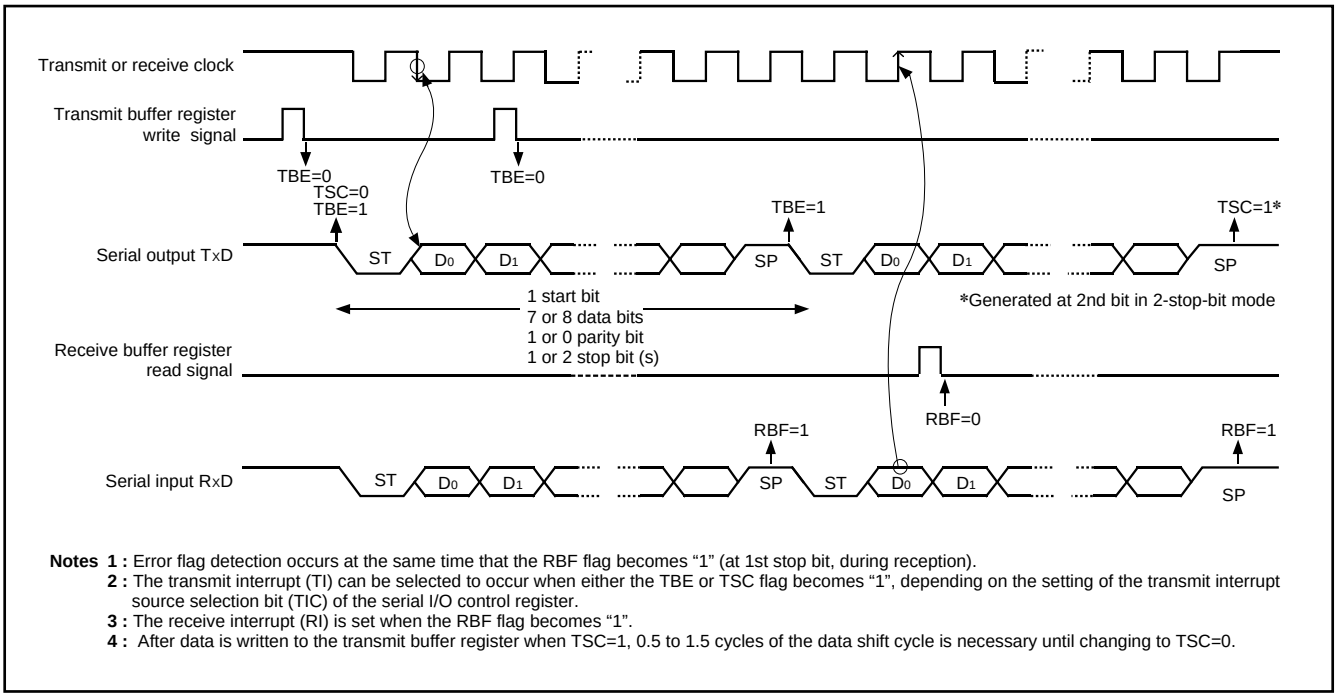


Fig. 23 Block diagram of UART serial I/O



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

Fig. 24 Operation of UART serial I/O function

Serial I/O Control Register (SIO1CON) 001A16

The serial I/O control register contains 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. One bit in this register (bit 4) is always valid and sets the output structure of the P45/TxD pin.

Serial I/O Status Register (SIO1STS) 001916

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

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

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

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

All bits of the serial I/O status register are initialized to “0” at reset, but if the transmit enable bit (bit 4) of the serial I/O 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/Receive Buffer Register (TB/RB) 001816

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

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.

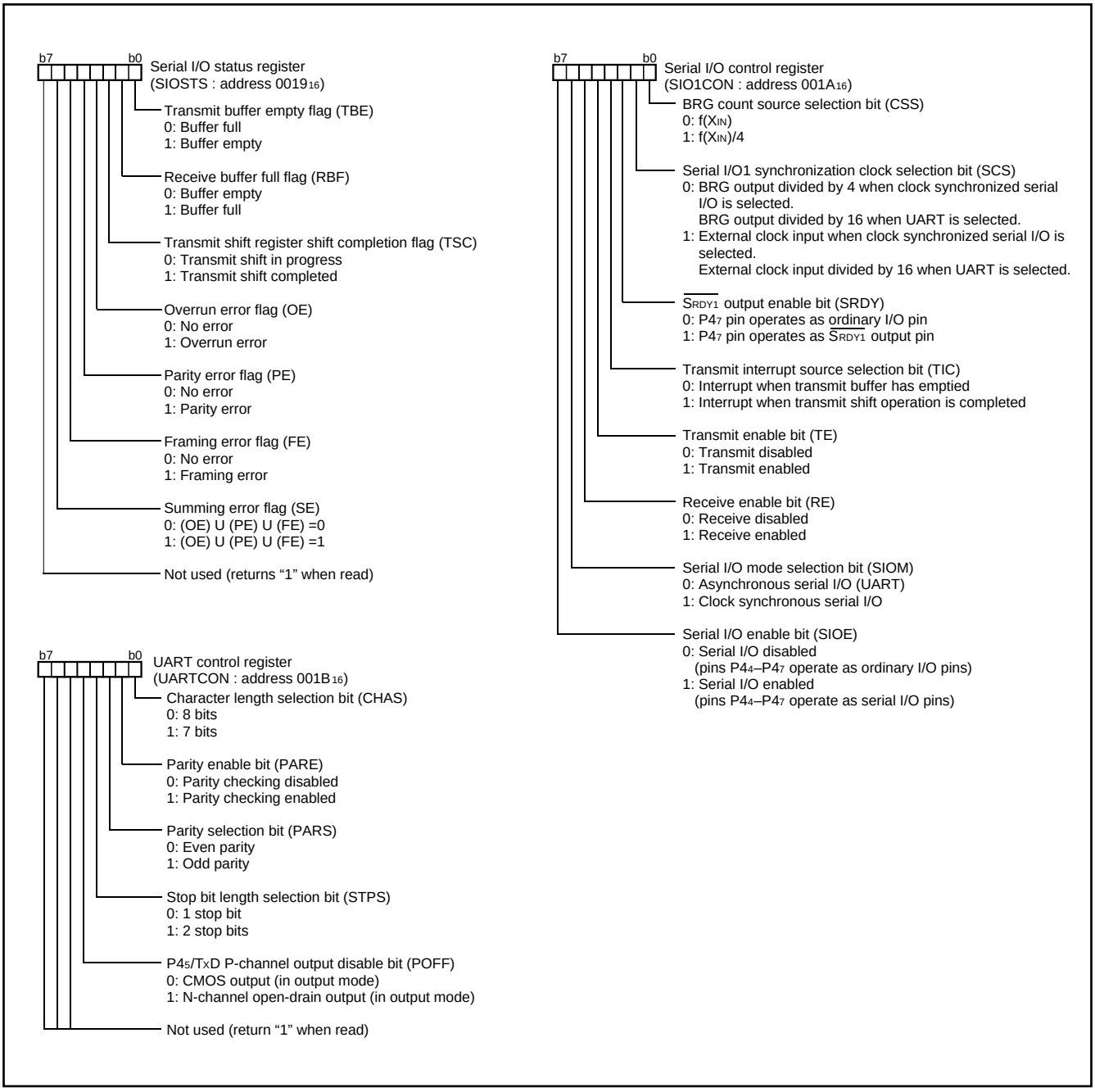


Fig. 25 Structure of serial I/O control registers

A-D CONVERTER

The functional blocks of the A-D converter are described below.

A-D Conversion Register (AD) 0035$_{16}$

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

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

The A-D control register controls the A-D conversion process. Bits 0 to 2 of this register select specific analog input pins. Bit 3 signals the completion of an A-D conversion. The value of this bit remains at “0” during an A-D conversion, then changes to “1” when the A-D conversion is completed.

Writing “0” to this bit starts the A-D conversion. Bit 4 controls the transistor which breaks the through current of the resistor ladder. When bit 5, which is the AD external trigger valid bit, is set to “1”, this bit enables A-D conversion even by a falling edge of an ADT input. Set ports which share with ADT pins to input when using an A-D external trigger.

Comparison Voltage Generator

The comparison voltage generator divides the voltage between AVSS and VREF by 256, and outputs the divided voltages.

Channel Selector

The channel selector selects one of the input ports P67/AN7 to P60/AN0.

Comparator and Control Circuit

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

Note that the comparator is constructed linked to a capacitor, so set f(XIN) to at least 500 kHz during A-D conversion.

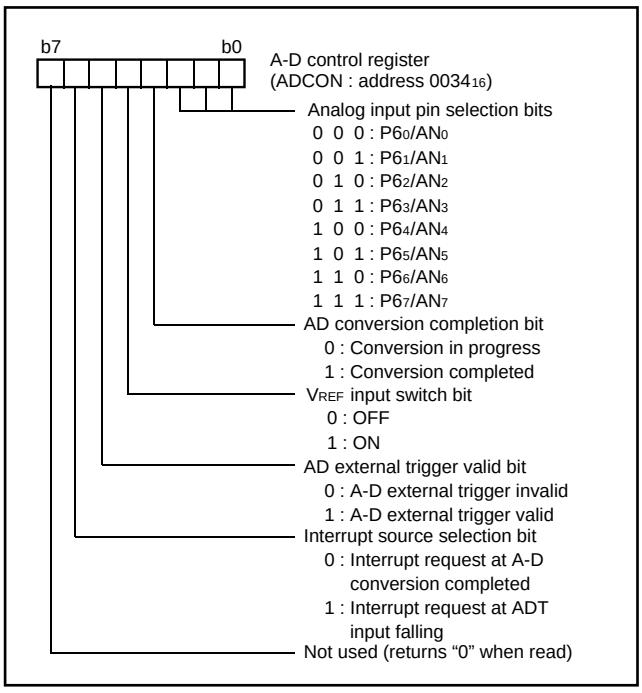


Fig. 26 Structure of A-D control register

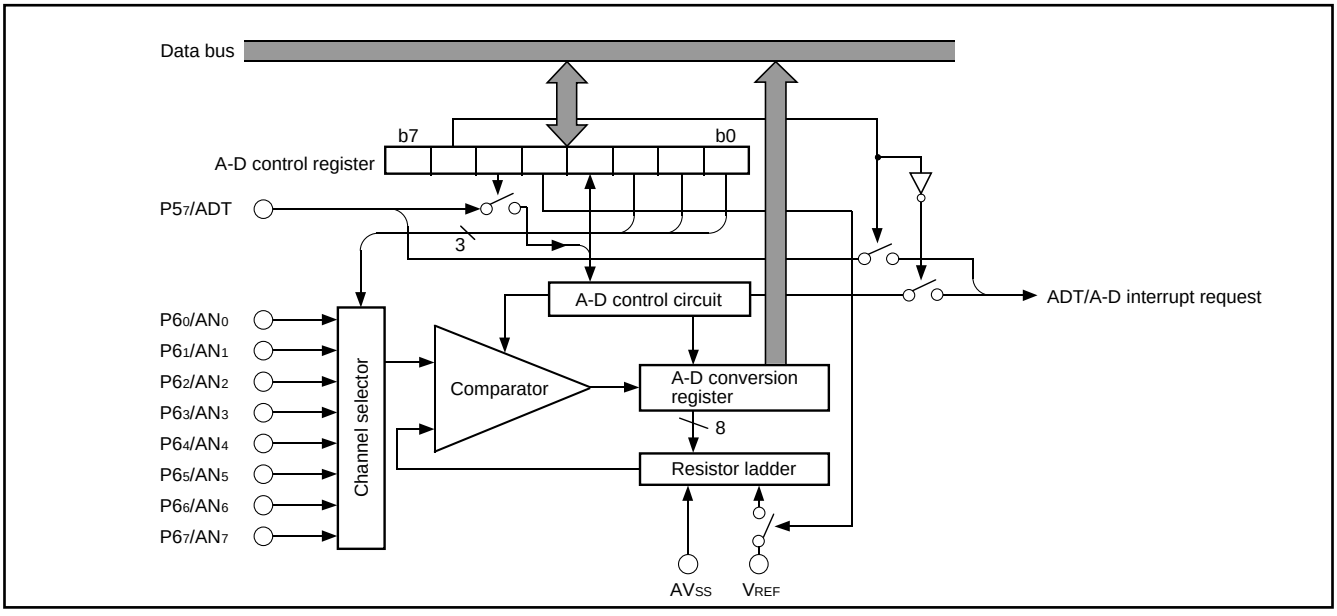


Fig. 27 A-D converter block diagram

LCD DRIVE CONTROL CIRCUIT

The 3822 group has the built-in Liquid Crystal Display (LCD) drive control circuit consisting of the following.

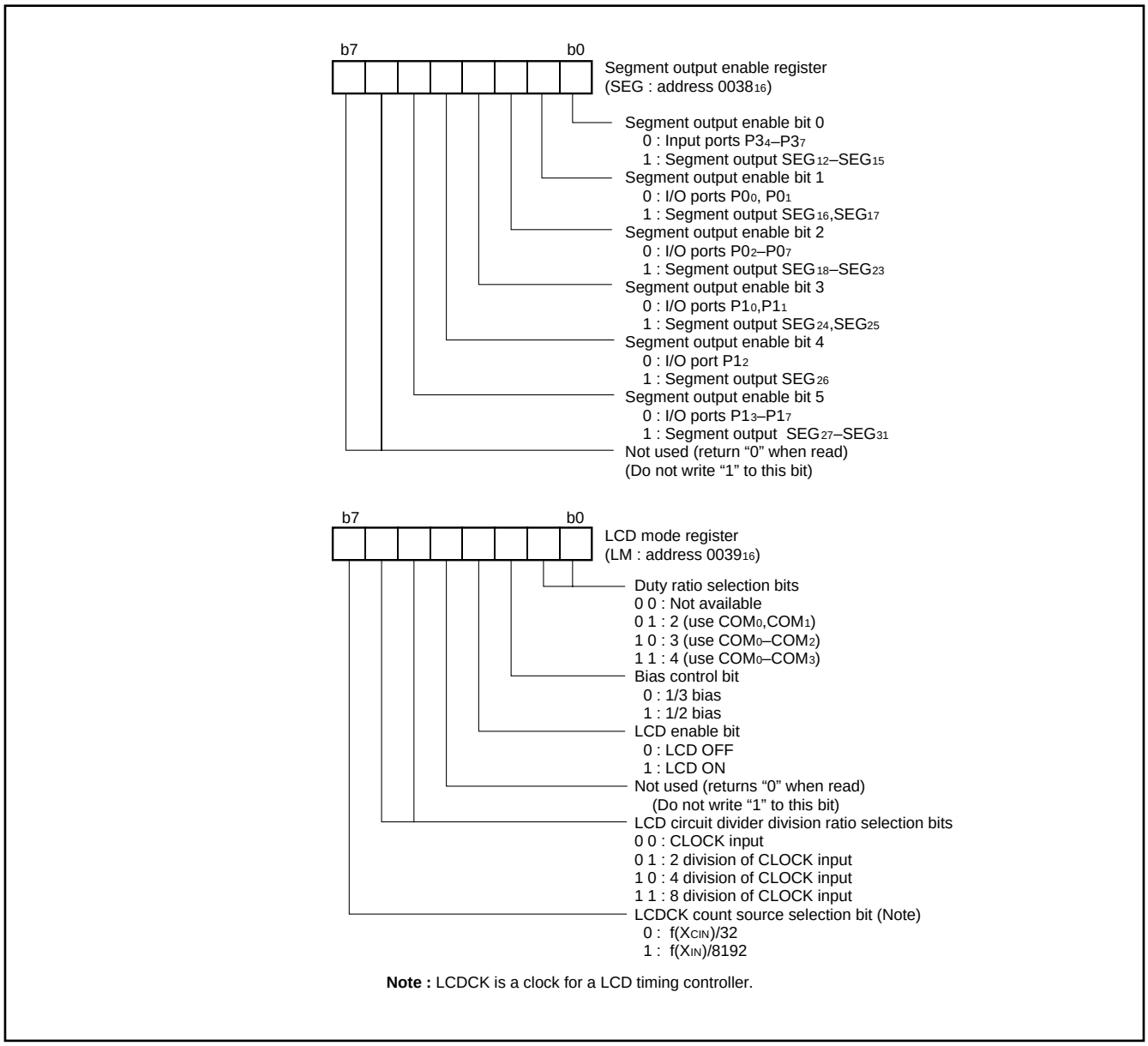
- LCD display RAM
- Segment output enable register
- LCD mode register
- Selector
- Timing controller
- Common driver
- Segment driver
- Bias control circuit

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

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

Table 2. Maximum number of display pixels at each duty ratio

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



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

Fig. 28 Structure of segment output enable register and LCD mode register

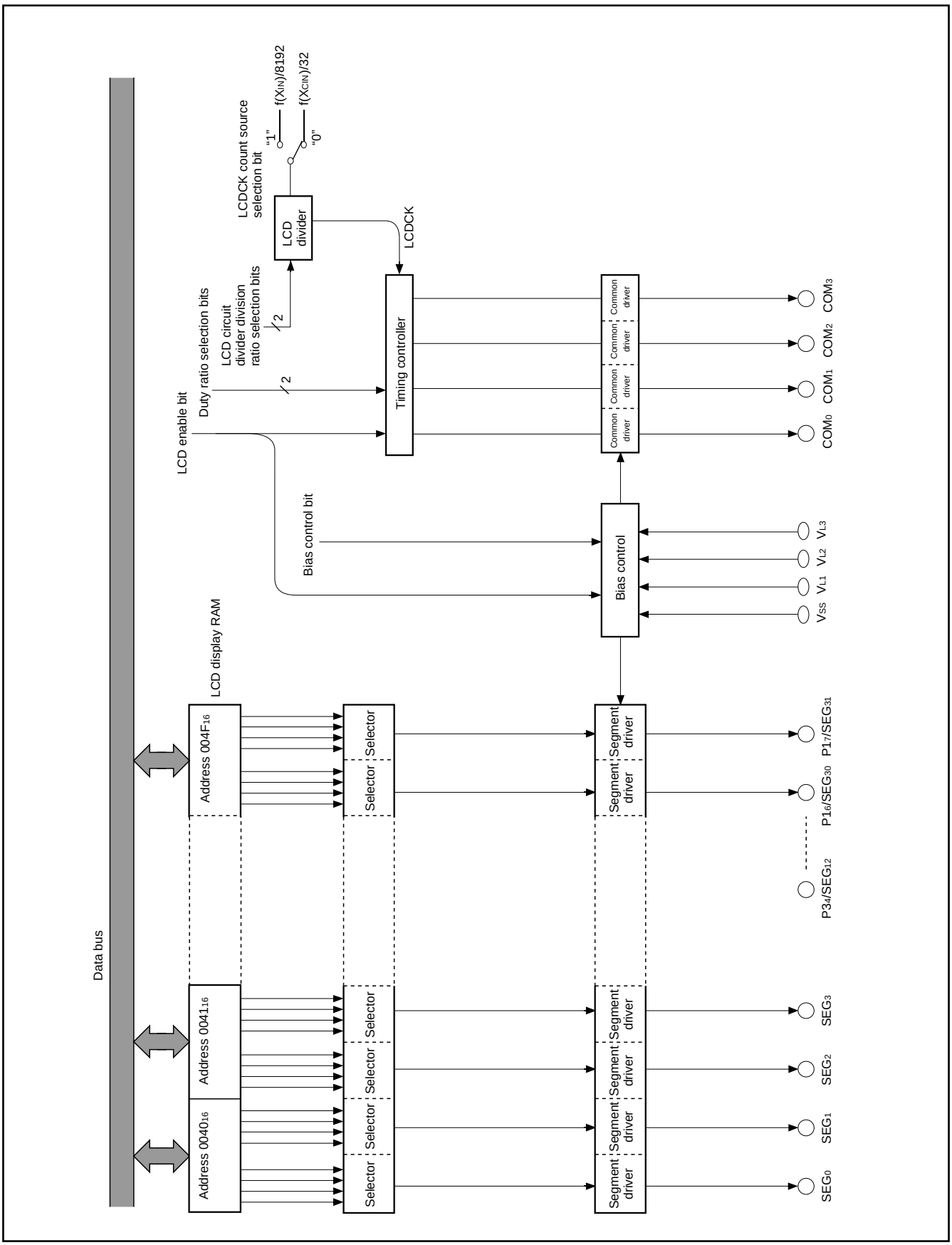


Fig. 29 Block diagram of LCD controller/driver

Bias Control and Applied Voltage to LCD Power Input Pins

To the LCD power input pins (V_{L1}–V_{L3}), apply the voltage shown in Table 3 according to the bias value.

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

Common Pin and Duty Ratio Control

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

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

Table 3. Bias control and applied voltage to V_{L1}–V_{L3}

Bias value	Voltage value
1/3 bias	$V_{L3}=V_{LCD}$ $V_{L2}=2/3\ V_{LCD}$ $V_{L1}=1/3\ V_{LCD}$
1/2 bias	$V_{L3}=V_{LCD}$ $V_{L2}=V_{L1}=1/2\ V_{LCD}$

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

Table 4. Duty ratio control and common pins used

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

Notes 1 : COM_2 and COM_3 are open
2 : COM_3 is open

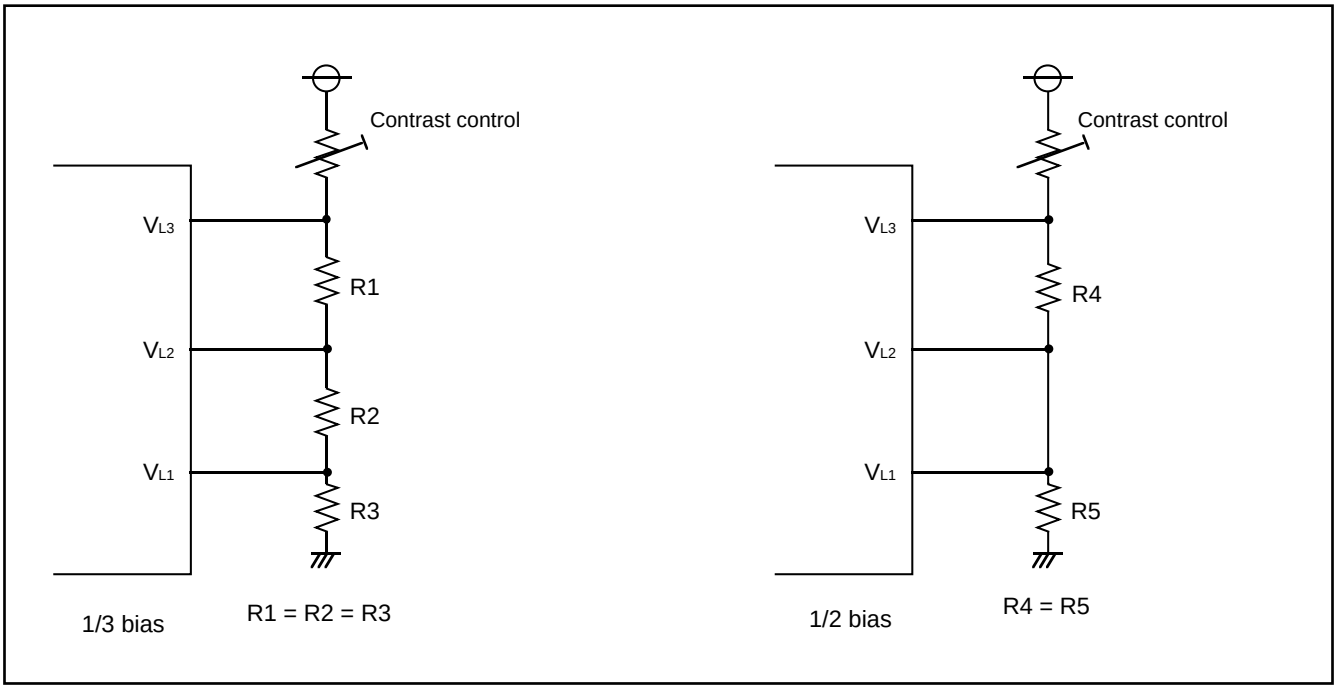


Fig. 30 Example of circuit at each bias

LCD Display RAM

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

LCD Drive Timing

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

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

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

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

Fig. 31 LCD display RAM map

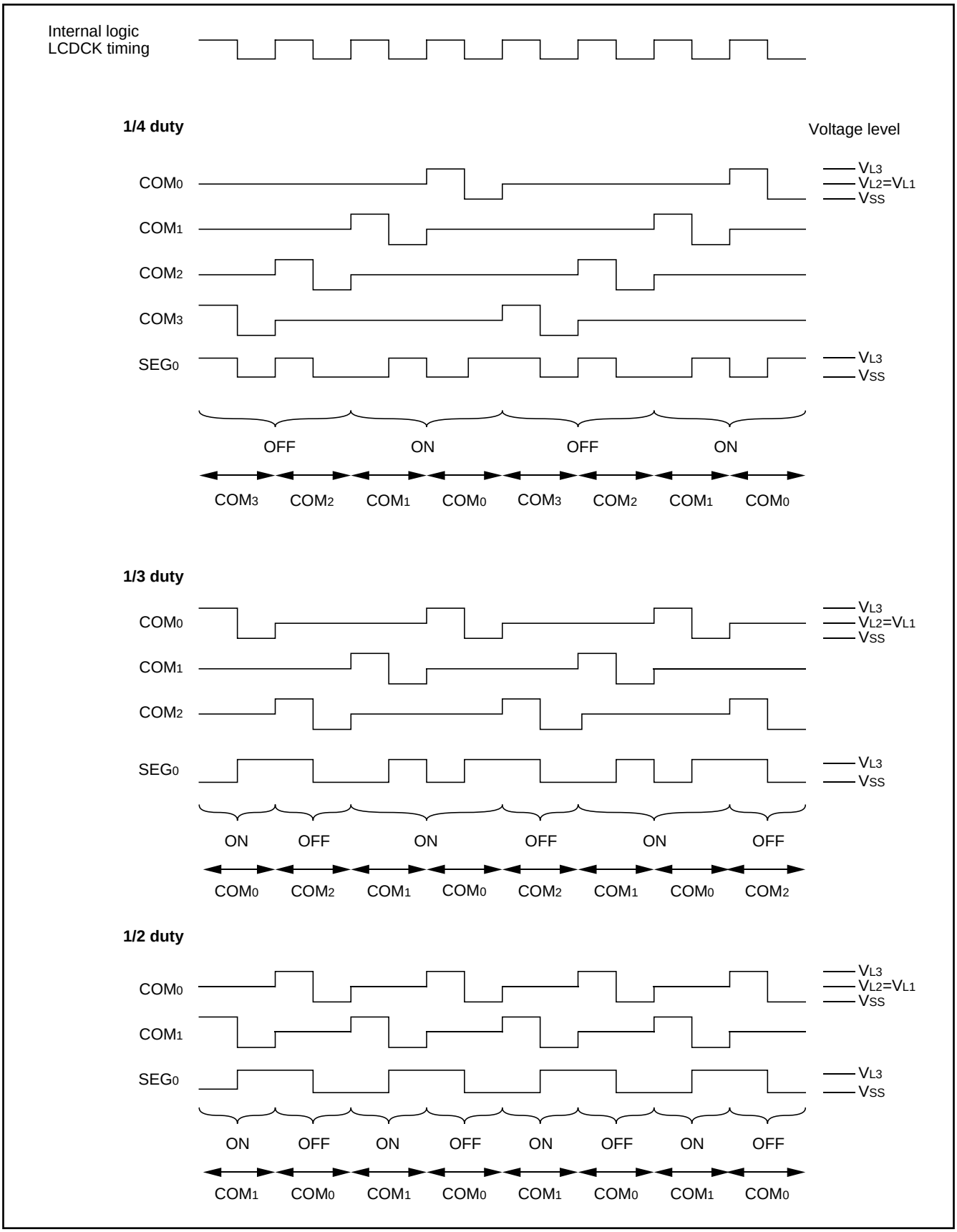


Fig. 32 LCD drive waveform (1/2 bias)

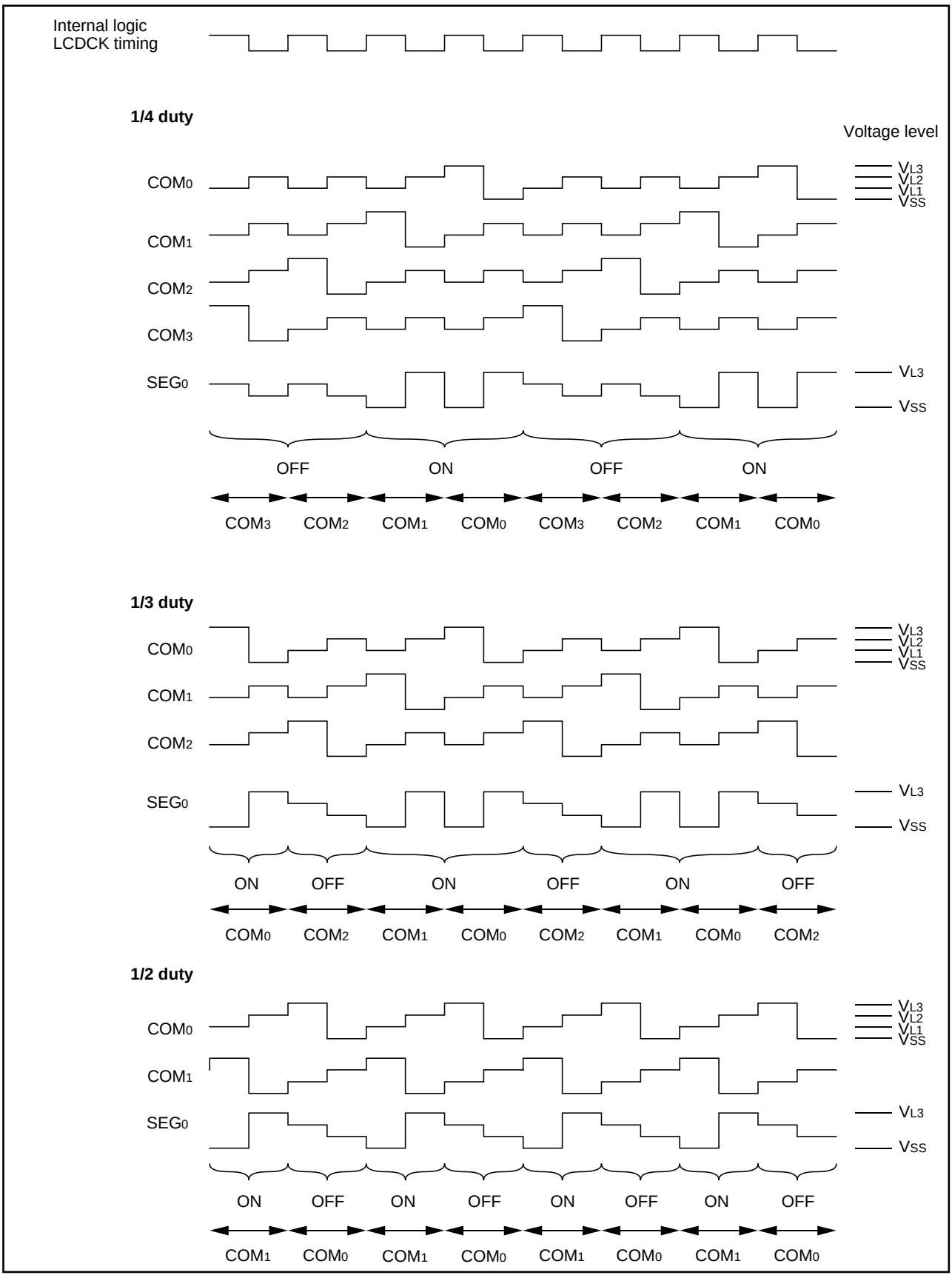


Fig. 33 LCD drive waveform (1/3 bias)

φ CLOCK OUTPUT FUNCTION

The internal system clock φ can be output from port P41 by setting the φ output control register. Set bit 1 of the port P4 direction register to when outputting φ clock.

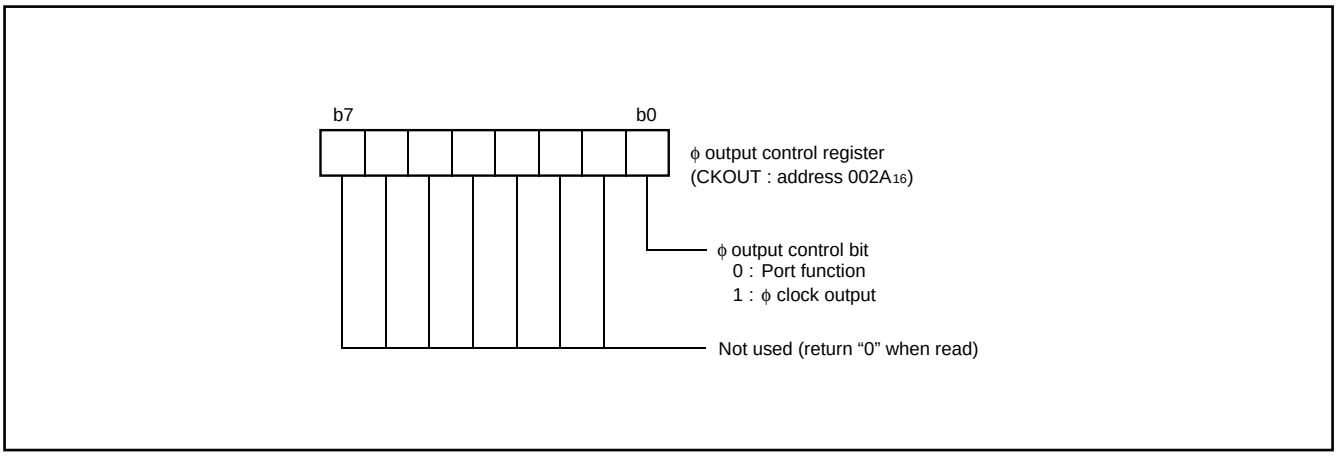


Fig. 34 Structure of φ output control register

RESET CIRCUIT

To reset the microcomputer, $\overline{\text{RESET}}$ pin should be held at an "L" level for 2 μs or more. Then the $\overline{\text{RESET}}$ pin is returned to an "H" level (the power source voltage should be between 2.5 V and 5.5 V, and the oscillation should be stable), reset is released. In order to give the XIN clock time to stabilize, internal operation does not begin until after 8200 XIN clock cycles (timer 1 and timer 2 are connected together and 512 cycles of f(XIN)/16) are complete. After the reset is completed, the program starts from the address contained in address FFFD16 (high-order byte) and address FFFC16 (low-order byte).

Make sure that the reset input voltage is less than 0.5 V for VCC of 2.5 V (Extended operating temperature version: the reset input voltage is less than 0.6V for VCC of 3.0V).

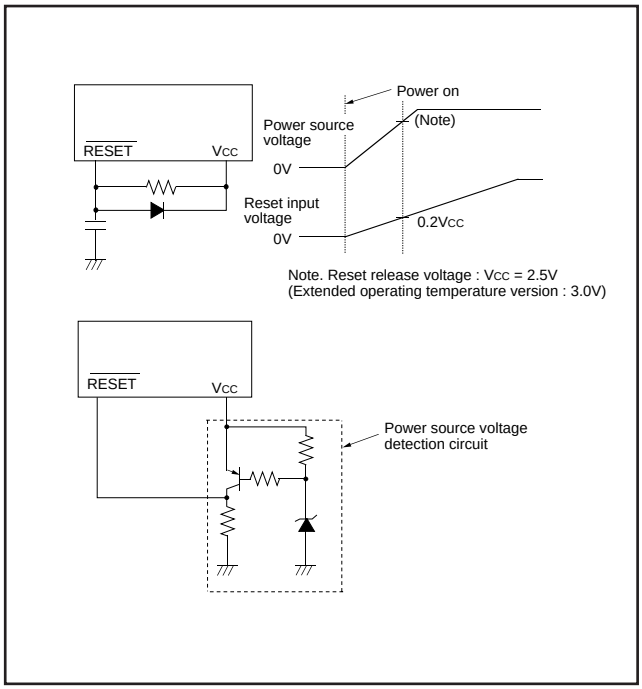


Fig. 35 Example of reset circuit

		Address	Register contents
(1)	Port P0 direction register	(0001_{16})	00_{16}
(2)	Port P1 direction register	(0003_{16})	00_{16}
(3)	Port P2 direction register	(0005_{16})	00_{16}
(4)	Port P4 direction register	(0009_{16})	00_{16}
(5)	Port P5 direction register	($000B_{16}$)	00_{16}
(6)	Port P6 direction register	($000D_{16}$)	00_{16}
(7)	Port P7 direction register	($000F_{16}$)	00_{16}
(8)	PULL register A	(0016_{16})	0 0 0 0 1 0 1 1
(9)	PULL register B	(0017_{16})	00_{16}
(10)	Serial I/O status register	(0019_{16})	1 0 0 0 0 0 0 0
(11)	Serial I/O control register	($001A_{16}$)	00_{16}
(12)	UART control register	($001B_{16}$)	1 1 1 0 0 0 0 0
(13)	Timer X (low)	(0020_{16})	FF_{16}
(14)	Timer X (high)	(0021_{16})	FF_{16}
(15)	Timer Y (low)	(0022_{16})	FF_{16}
(16)	Timer Y (high)	(0023_{16})	FF_{16}
(17)	Timer 1	(0024_{16})	FF_{16}
(18)	Timer 2	(0025_{16})	01_{16}
(19)	Timer 3	(0026_{16})	FF_{16}
(20)	Timer X mode register	(0027_{16})	00_{16}
(21)	Timer Y mode register	(0028_{16})	00_{16}
(22)	Timer 123 mode register	(0029_{16})	00_{16}
(23)	φ output control register	($002A_{16}$)	00_{16}
(24)	A-D control register	(0034_{16})	0 0 0 0 1 0 0 0
(25)	Segment output enable register	(0038_{16})	00_{16}
(26)	LCD mode register	(0039_{16})	00_{16}
(27)	Interrupt edge selection register	($003A_{16}$)	00_{16}
(28)	CPU mode register	($003B_{16}$)	0 1 0 0 1 0 0 0
(29)	Interrupt request register 1	($003C_{16}$)	00_{16}
(30)	Interrupt request register 2	($003D_{16}$)	00_{16}
(31)	Interrupt control register 1	($003E_{16}$)	00_{16}
(32)	Interrupt control register 2	($003F_{16}$)	00_{16}
(33)	Processor status register	(PS)	X X X X X 1 X X
(34)	Program counter	(PC_H)	Contents of address FFFD 16
		(PC_L)	Contents of address FFFC 16

Note X : Undefined
The contents of all other registers and RAM are undefined after reset, so they must be initialized by software.

Fig. 36 Internal state of microcomputer after reset

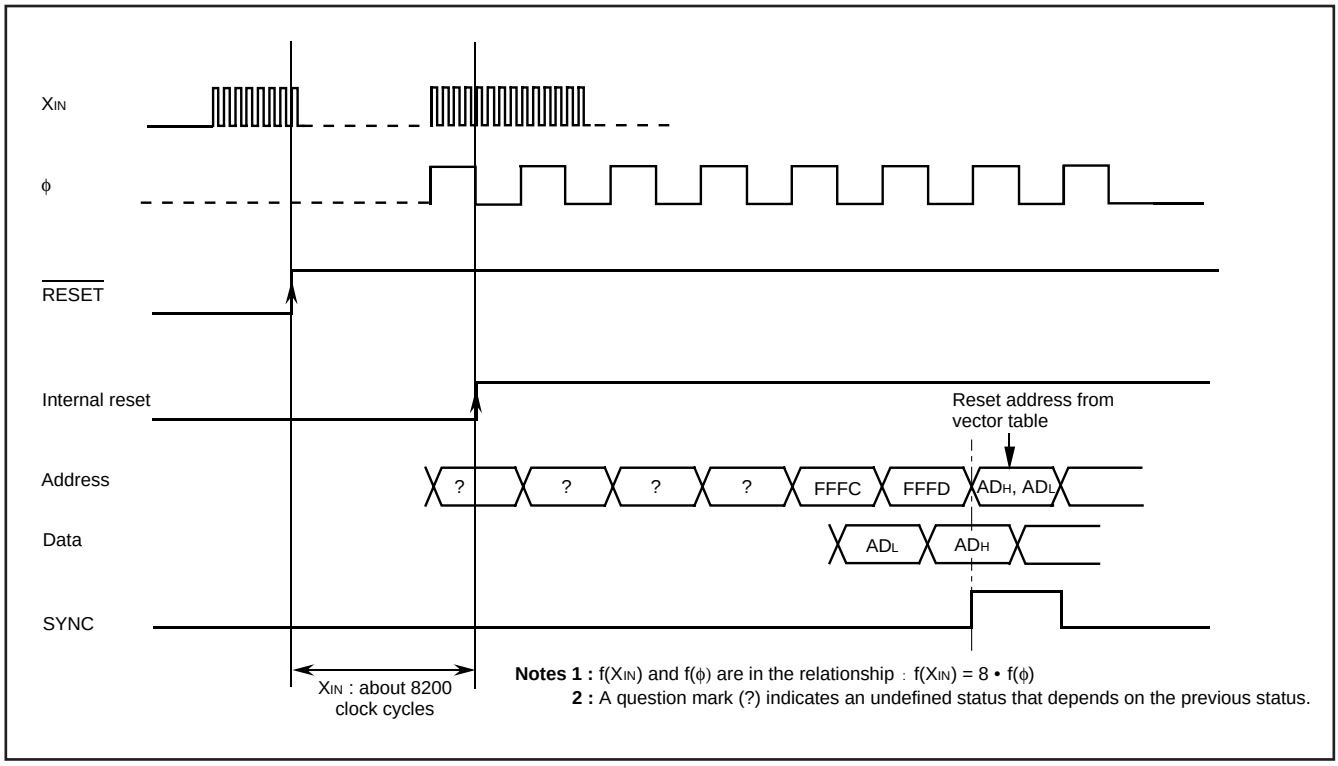


Fig. 37 Reset sequence

CLOCK GENERATING CIRCUIT

The 3822 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.

To supply a clock signal externally, input it to the XIN pin and make the XOUT pin open. The sub-clock XCIN-XCOUT oscillation circuit cannot directly input clocks that are externally generated. Accordingly, be sure to cause an external resonator to oscillate.

Immediately after poweron, only the XIN oscillation circuit starts oscillating, and XCIN and XCOUT pins function as I/O ports. The pull-up resistor of XCIN and XCOUT pins must be made invalid to use the sub-clock.

Frequency Control

Middle-speed mode

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

High-speed mode

The internal clock ϕ is half the frequency of XIN.

Low-speed mode

- The internal clock ϕ is half the frequency of XCIN.
- A low-power consumption operation can be realized by stopping the main clock XIN in this mode. To stop the main clock, set bit 5 of the CPU mode register to "1".
  When the main clock XIN is restarted, set enough time for oscillation to stabilize by programming.

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 poweron and at returning from stop mode. When switching the mode between middle/high-speed and low-speed, set the frequency on condition that f(XIN)>3f(XCIN).

Oscillation Control

Stop mode

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

Either XIN or XCIN divided by 16 is input to timer 1 as count source, and the output of timer 1 is connected to timer 2.

The bits of the timer 123 mode register except bit 4 are cleared to "0". Set the timer 1 and timer 2 interrupt enable bits to disabled ("0") before executing the STP instruction.

Oscillator restarts at reset or when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU until timer 2 underflows. This allows time for the clock circuit oscillation to stabilize.

Wait mode

If the WIT instruction is executed, the internal clock ϕ stops at an "H" level. The states of XIN and XCIN are the same as the state before the executing the WIT instruction. The internal clock restarts at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

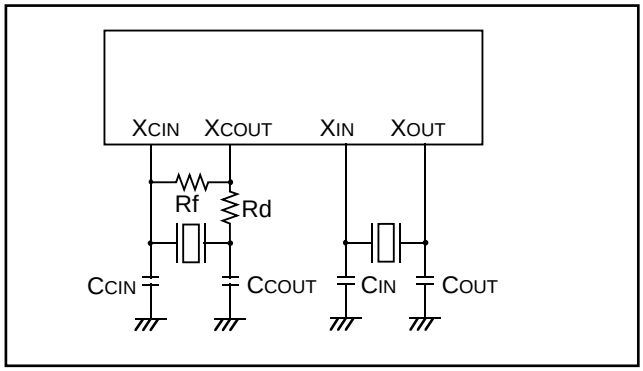


Fig. 38 Ceramic resonator circuit

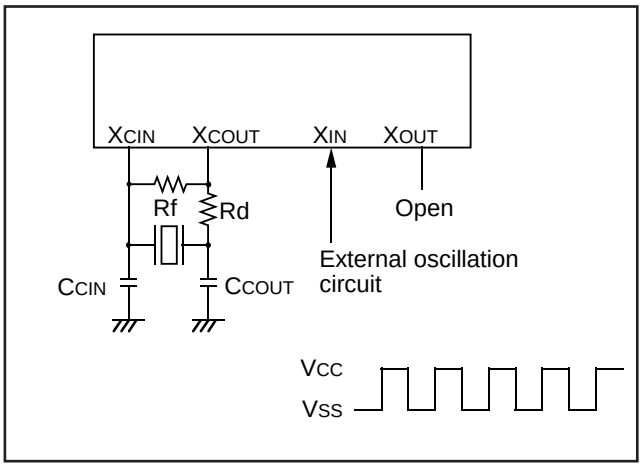
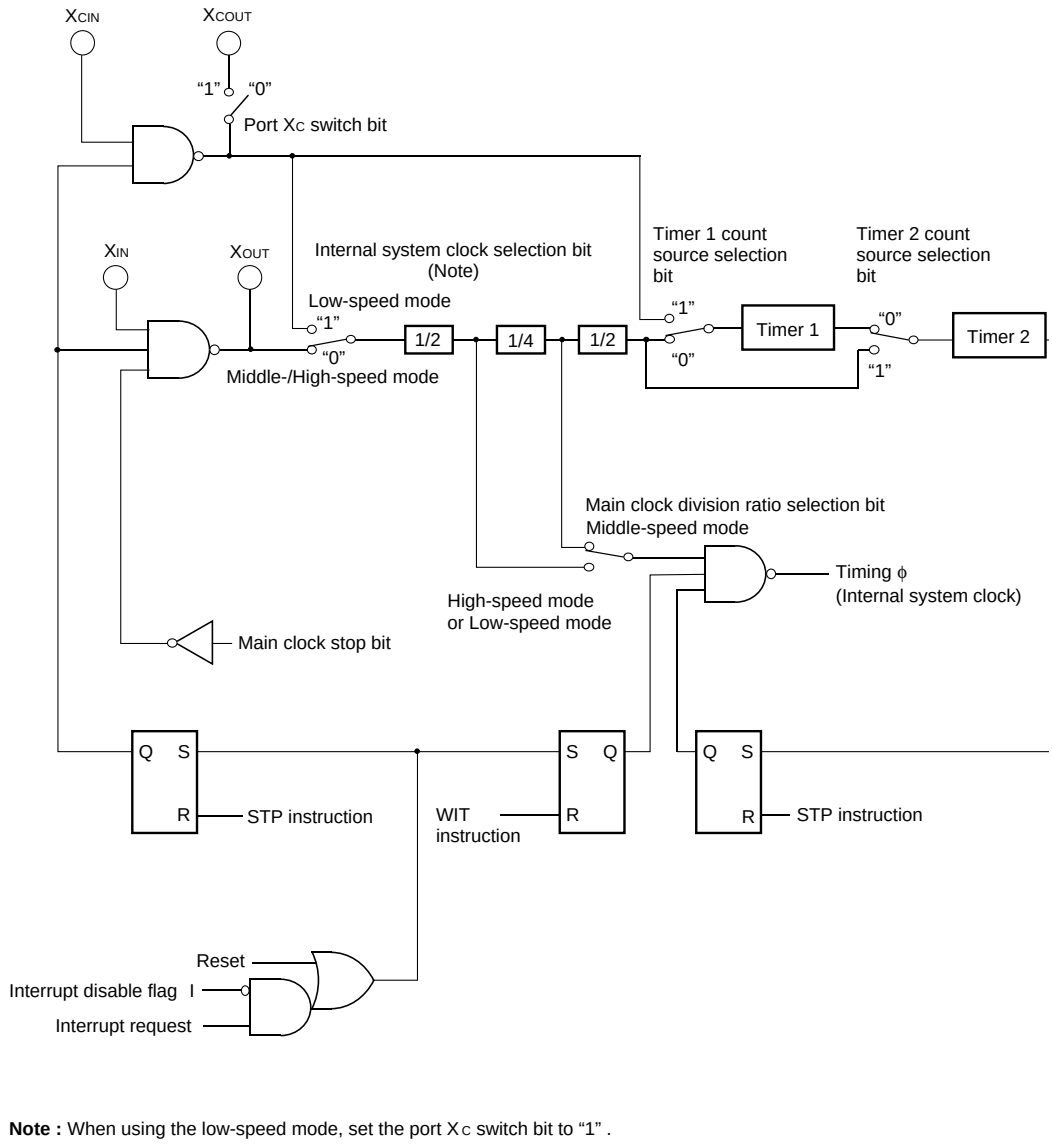
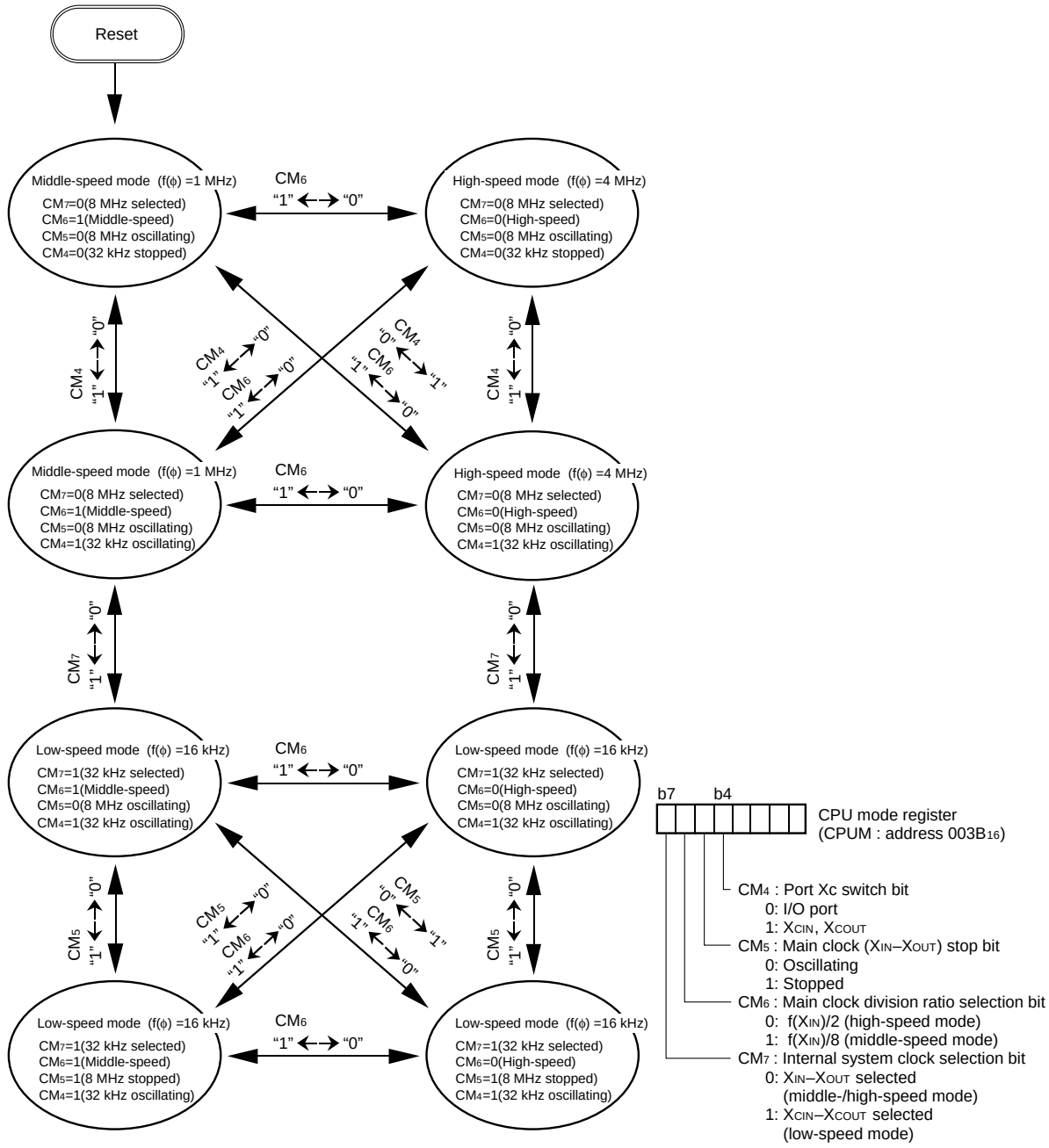


Fig. 39 External clock input circuit



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

Fig. 40 Clock generating circuit block diagram



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

Fig. 41 State transitions of internal clock

NOTES ON PROGRAMMING

Processor Status Register

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

Interrupt

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. Only the ADC and SBC instructions yield proper decimal results. 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.
The carry flag can be used to indicate whether a carry or borrow has occurred. Initialize the carry flag before each calculation. Clear the carry flag before an ADC and set the flag before an SBC.

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

Ports

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

- The data transfer instruction (LDA, etc.)
- The operation instruction when the index X mode flag (T) is "1"
- The addressing mode which uses the value of a direction register as an index
- The bit-test instruction (BBC or BBS, etc.) to a direction register
- The read-modify-write instruction (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 $\overline{\text{SRDY}}$ signal, set the transmit enable bit, the receive enable bit, and the $\overline{\text{SRDY}}$ output enable bit to "1".
Serial I/O continues to output the final bit from the TxD pin after transmission is completed.

A-D Converter

The comparator uses internal capacitors whose charge will be lost if the clock frequency is too low.
Make sure that f(XIN) is at least 500 kHz during an A-D conversion.
Do not execute the STP or WIT instruction during an A-D conversion.

Instruction Execution Time

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

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

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

ROM PROGRAMMING METHOD

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

Package	Name of Programming Adapter
80P6N-A	PCA4738F-80A
80P6S-A	PCA4738G-80
80P6D-A	PCA4738H-80
80D0	PCA4738L-80A

The PROM of the blank One Time PROM version is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Figure 36 is recommended to verify programming.

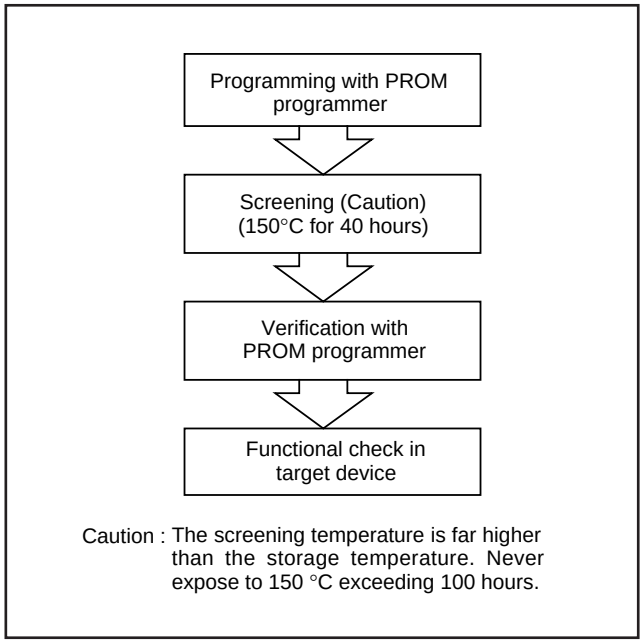


Fig. 42 Programming and testing of One Time PROM version

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltage	All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to 7.0	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_4$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$		–0.3 to V_{CC} +0.3	V
V_I	Input voltage V_{L1}		–0.3 to V_{L2}	V
V_I	Input voltage V_{L2}		V_{L1} to V_{L3}	V
V_I	Input voltage V_{L3}		V_{L2} to V_{CC} +0.3	V
V_I	Input voltage $\overline{RESET}$, X_{IN}		–0.3 to V_{CC} +0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$	At output port	–0.3 to V_{CC} +0.3	V
		At segment output	–0.3 to V_{L3} +0.3	V
V_O	Output voltage $P3_4$–$P3_7$	At segment output	–0.3 to V_{L3} +0.3	V
V_O	Output voltage $P2_0$–$P2_7$, $P4_1$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$		–0.3 to V_{CC} +0.3	V
V_O	Output voltage SEG_0–SEG_{11}		–0.3 to V_{L3} +0.3	V
V_O	Output voltage X_{OUT}		–0.3 to V_{CC} +0.3	V
Pd	Power dissipation	T_a = 25 °C	300	mW
Topr	Operating temperature		–20 to 85 (Note 1)	°C
Tstg	Storage temperature		–40 to 125 (Note 2)	°C

Notes 1 : Extended operating temperature version : –40 to 85°C
2 : Extended operating temperature version : –65 to 150°C

RECOMMENDED OPERATING CONDITIONS

(V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C, unless otherwise noted.
Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5V, T_a = –20 to 85°C)

Symbol	Parameter			Limits			Unit
				Min.	Typ.	Max.	
V_{CC}	Power source voltage	High-speed mode f(X_{IN})=8 MHz		4.0	5.0	5.5	V
		Middle-speed mode f(X_{IN})=8 MHz	T_a = –20 to 85°C	2.5	5.0	5.5	
			T_a = –40 to –20°C	3.0	5.0	5.5	
		Low-speed mode	T_a = –20 to 85°C	2.5	5.0	5.5	
			T_a = –40 to –20°C	3.0	5.0	5.5	
V_{SS}	Power source voltage				0		V
V_{REF}	A-D conversion reference input voltage			2		V_{CC}	V
AV_{SS}	Analog power source voltage				0		V
V_{IA}	Analog input voltage AN_0–AM_7			AV_{SS}		V_{CC}	V
V_{IH}	“H” input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_4$–$P3_7$, $P4_0$, $P4_1$, $P4_5$, $P4_7$, $P5_2$, $P5_3$, $P5_6$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (CM_4=0)		0.7 V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage	$P2_0$–$P2_7$, $P4_2$–$P4_4$, $P4_6$, $P5_0$, $P5_1$, $P5_4$, $P5_5$, $P5_7$		0.8 V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage	$\overline{RESET}$		0.8 V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage	X_{IN}		0.8 V_{CC}		V_{CC}	V
V_{IL}	“L” input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_4$–$P3_7$, $P4_0$, $P4_1$, $P4_5$, $P4_7$, $P5_2$, $P5_3$, $P5_6$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (CM_4=0)		0		0.3 V_{CC}	V
V_{IL}	“L” input voltage	$P2_0$–$P2_7$, $P4_2$–$P4_4$, $P4_6$, $P5_0$, $P5_1$, $P5_4$, $P5_5$, $P5_7$		0		0.2 V_{CC}	V
V_{IL}	“L” input voltage	$\overline{RESET}$		0		0.2 V_{CC}	V
V_{IL}	“L” input voltage	X_{IN}		0		0.2 V_{CC}	V

RECOMMENDED OPERATING CONDITIONS

(V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C, unless otherwise noted.
Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$\Sigma I_{OH(peak)}$	"H" total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$ (Note 1)			–40	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current	$P4_1$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)			–40	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$ (Note 1)			40	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current	$P4_1$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)			40	mA
$\Sigma I_{OH(avg)}$	"H" total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$ (Note 1)			–20	mA
$\Sigma I_{OH(avg)}$	"H" total average output current	$P4_1$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)			–20	mA
$\Sigma I_{OL(avg)}$	"L" total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$ (Note 1)			20	mA
$\Sigma I_{OL(avg)}$	"L" total average output current	$P4_1$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)			20	mA
$I_{OH(peak)}$	"H" peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$ (Note 2)			–2	mA
$I_{OH(peak)}$	"H" peak output current	$P2_0$–$P2_7$, $P4_1$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 2)			–5	mA
$I_{OL(peak)}$	"L" peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$ (Note 2)			5	mA
$I_{OL(peak)}$	"L" peak output current	$P2_0$–$P2_7$, $P4_1$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 2)			10	mA
$I_{OH(avg)}$	"H" average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$ (Note 3)			–1.0	mA
$I_{OH(avg)}$	"H" average output current	$P2_0$–$P2_7$, $P4_1$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 3)			–2.5	mA
$I_{OL(avg)}$	"L" average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$ (Note 3)			2.5	mA
$I_{OL(avg)}$	"L" average output current	$P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 3)			5.0	mA
$f(CNTR_0)$ $f(CNTR_1)$	Input frequency for timers X and Y (duty cycle 50 %)	4.0 V ≤ V_{CC} ≤ 5.5 V			4.0	MHz
		2.5 V ≤ V_{CC} ≤ 4.0 V			$(2 \times V_{CC})$–4	MHz
$f(X_{IN})$	Main clock input oscillation frequency (Note 4)	High-speed mode (4.0 V ≤ V_{CC} ≤ 5.5 V)			8.0	MHz
		High-speed mode (2.5 V ≤ V_{CC} ≤ 4.0 V)			$(4 \times V_{CC})$–8	MHz
		Middle-speed mode			8.0	MHz
$f(X_{CIN})$	Sub-clock input oscillation frequency (Note 4, 5)			32.768	50	kHz

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

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

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

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

5 : When using the microcomputer in low-speed mode, make sure that the sub-clock input oscillation frequency on condition that $f(X_{CIN}) < f(X_{IN})/3$.

ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.0 to 5.5 V, T_a = –20 to 85°C, unless otherwise noted.
Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C)

Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
V_{OH}	"H" output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$	I_{OH} = –2.5 mA		V_{CC}–2.0			V
			I_{OH} = –0.6 mA V_{CC} = 2.5 V		V_{CC}–1.0			V
V_{OH}	"H" output voltage	$P2_0$–$P2_7$, $P4_1$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)	I_{OH} = –5 mA		V_{CC}–2.0			V
			I_{OH} = –1.25 mA		V_{CC}–0.5			V
			I_{OH} = –1.25 mA V_{CC} = 2.5 V		V_{CC}–1.0			V
V_{OL}	"L" output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$	I_{OL} = 5 mA				2.0	V
			I_{OL} = 1.25 mA				0.5	V
			I_{OL} = 1.25 mA V_{CC} = 2.5 V				1.0	V
V_{OL}	"L" output voltage	$P2_0$–$P2_7$, $P4_1$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)	I_{OL} = 10 mA				2.0	V
			I_{OL} = 2.5 mA				0.5	V
			I_{OL} = 2.5 mA V_{CC} = 2.5 V				1.0	V
$V_{T+} - V_{T-}$	Hysteresis	$CNTR_0$, $CNTR_1$, INT_0–INT_3, $P2_0$–$P2_7$				0.5		V
$V_{T+} - V_{T-}$	Hysteresis	RxD, SCLK				0.5		V
$V_{T+} - V_{T-}$	Hysteresis	$\overline{RESET}$	$\overline{RESET}$: V_{CC}=2.5 V to 5.5 V			0.5		V
I_{IH}	"H" input current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_7$	V_I = V_{CC} Pull-downs "off"				5.0	µA
			V_{CC}= 5.0 V, V_I = V_{CC} Pull-downs "on"	T_a = –20 to 85°C	30	70	140	µA
				T_a = –40 to –20°C		70	170	
			V_{CC}= 3.0 V, V_I = V_{CC} Pull-downs "on"	T_a = –20 to 85°C	6.0	25	45	µA
				T_a = –40 to –20°C		25	55	
I_{IH}	"H" input current	$P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	V_I = V_{CC}				5.0	µA
I_{IH}	"H" input current	$\overline{RESET}$	V_I = V_{CC}				5.0	µA
I_{IH}	"H" input current	X_{IN}	V_I = V_{CC}			4.0		µA
I_{IL}	"L" input current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_4$–$P3_7$, $P4_0$					–5.0	µA
I_{IL}	"L" input current	$P2_0$–$P2_7$, $P4_1$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$	V_I = V_{SS} Pull-ups "off"				–5.0	µA
			V_{CC}= 5.0 V, V_I = V_{SS} Pull-ups "on"		–30	–70	–140	µA
			V_{CC}= 3.0 V, V_I = V_{SS} Pull-ups "on"		–6	–25	–45	µA
I_{IL}	"L" input current	$\overline{RESET}$	V_I = V_{SS}				–5.0	µA
I_{IL}	"L" input current	X_{IN}	V_I = V_{SS}			–4.0		µA

Note 1: When "1" is set to port X_C switch bit (bit 4 of address $003B_{16}$) of CPU mode register, the drive ability of port $P7_0$ is different from the value above mentioned.

ELECTRICAL CHARACTERISTICS (V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C, unless otherwise noted.

Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{RAM}	RAM hold voltage	When clock is stopped	2.0		5.5	V
I_{CC}	Power source current	• High-speed mode, V_{CC} = 5 V $f(X_{IN})$ = 8 MHz $f(X_{CIN})$ = 32.768 kHz Output transistors “off” A-D converter in operating		6.4	13	mA
		• High-speed mode, V_{CC} = 5 V $f(X_{IN})$ = 8 MHz (in WIT state) $f(X_{CIN})$ = 32.768 kHz Output transistors “off” A-D converter stopped		1.6	3.2	mA
		• Low-speed mode, V_{CC} = 5 V, $T_a \leq 55°C$ $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz Output transistors “off”		25	36	µA
		• Low-speed mode, V_{CC} = 5 V, T_a = 25°C $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz (in WIT state) Output transistors “off”		7.0	14.0	µA
		• Low-speed mode, V_{CC} = 3 V, $T_a \leq 55°C$ $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz Output transistors “off”		15	22	µA
		• Low-speed mode, V_{CC} = 3 V, T_a = 25°C $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz (in WIT state) Output transistors “off”		4.5	9.0	µA
		All oscillation stopped (in STP state) Output transistors “off” — T_a = 25 °C		0.1	1.0	µA
		All oscillation stopped (in STP state) Output transistors “off” — T_a = 85 °C			10	µA

A-D CONVERTER CHARACTERISTICS

(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85°C, 4 MHz ≤ f(X_{IN}) ≤ 8 MHz, middle-/high-speed mode, unless otherwise noted. Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy (excluding quantization error)	V_{CC} = V_{REF} = 5 V			±2	LSB
t_{CONV}	Conversion time	f(X_{IN}) = 8 MHz		12.5 (Note)		µs
R_{LADDER}	Ladder resistor		12	35	100	kΩ
V_{REF}	Reference input current	V_{REF} = 5 V	50	150	200	µA
I_{IA}	Analog port input current				5.0	µA

Note : When an internal trigger is used in middle-speed mode, it is 14 µs.

TIMING REQUIREMENTS 1

(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85°C, unless otherwise noted.
Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{w(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{c(X_{IN})}$	Main clock input cycle time (X_{IN} input)	125			ns
$t_{wH(X_{IN})}$	Main clock input “H” pulse width	45			ns
$t_{wL(X_{IN})}$	Main clock input “L” pulse width	40			ns
$t_{c(CNTR)}$	$CNTR_0$, $CNTR_1$ input cycle time	200			ns
$t_{wH(CNTR)}$	$CNTR_0$, $CNTR_1$ input “H” pulse width	80			ns
$t_{wL(CNTR)}$	$CNTR_0$, $CNTR_1$ input “L” pulse width	80			ns
$t_{wH(INT)}$	INT_0 to INT_3 input “H” pulse width	80			ns
$t_{wL(INT)}$	INT_0 to INT_3 input “L” pulse width	80			ns
$t_{c(S_{CLK})}$	Serial I/O clock input cycle time (Note)	800			ns
$t_{wH(S_{CLK})}$	Serial I/O clock input “H” pulse width (Note)	370			ns
$t_{wL(S_{CLK})}$	Serial I/O clock input “L” pulse width (Note)	370			ns
$t_{su(R_{XD}–S_{CLK})}$	Serial I/O input set up time	220			ns
$t_{h(S_{CLK}–R_{XD})}$	Serial I/O input hold time	100			ns

Note : When f(X_{IN}) = 8 MHz and bit 6 of address $001A_{16}$ is “1” (clock synchronous).
Divide this value by four when f(X_{IN}) = 8 MHz and bit 6 of address $001A_{16}$ is “0” (UART).

TIMING REQUIREMENTS 2

(V_{CC} = 2.5 to 4.0 V, V_{SS} = 0 V, T_a = –20 to 85°C, unless otherwise noted.
Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{w(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{c(X_{IN})}$	Main clock input cycle time (X_{IN} input)	125			ns
$t_{wH(X_{IN})}$	Main clock input “H” pulse width	45			ns
$t_{wL(X_{IN})}$	Main clock input “L” pulse width	40			ns
$t_{c(CNTR)}$	$CNTR_0$, $CNTR_1$ 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 to INT_3 input “H” pulse width	230			ns
$t_{wL(INT)}$	INT_0 to INT_3 input “L” pulse width	230			ns
$t_{c(S_{CLK})}$	Serial I/O clock input cycle time (Note)	2000			ns
$t_{wH(S_{CLK})}$	Serial I/O clock input “H” pulse width (Note)	950			ns
$t_{wL(S_{CLK})}$	Serial I/O clock input “L” pulse width (Note)	950			ns
$t_{su(R_{XD}–S_{CLK})}$	Serial I/O input set up time	400			ns
$t_{h(S_{CLK}–R_{XD})}$	Serial I/O input hold time	200			ns

Note : When f(X_{IN}) = 2 MHz and bit 6 of address $001A_{16}$ is “1” (clock synchronous).
Divide this value by four when f(X_{IN}) = 2 MHz and bit 6 of address $001A_{16}$ is “0” (UART).

SWITCHING CHARACTERISTICS 1 (V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85°C, unless otherwise noted.

Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{wH(SCLK)}$	Serial I/O clock output “H” pulse width	$t_{c(SCLK)}$/2–30			ns
$t_{wL(SCLK)}$	Serial I/O clock output “L” pulse width	$t_{c(SCLK)}$/2–30			ns
$t_{d(SCLK-TXD)}$	Serial I/O output delay time (Note 1)			140	ns
$t_{v(SCLK-TXD)}$	Serial I/O output valid time (Note 1)	–30			ns
$t_{r(SCLK)}$	Serial I/O clock output rising time			30	ns
$t_{f(SCLK)}$	Serial I/O clock output falling time			30	ns
$t_{r(CMOS)}$	CMOS output rising time (Note 2)		10	30	ns
$t_{f(CMOS)}$	CMOS output falling time (Note 2)		10	30	ns

Notes 1 : When the P45/TXD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.

2 : XOUT and XCOUT pins are excluded.

SWITCHING CHARACTERISTICS 2 (V_{CC} = 2.5 to 4.0 V, V_{SS} = 0 V, T_a = –20 to 85°C, unless otherwise noted.

Extended operating temperature version : V_{CC} = 3.0 to 5.5 V, T_a = –40 to –20°C and V_{CC} = 2.5 to 5.5 V, T_a = –20 to 85°C)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{wH(SCLK)}$	Serial I/O clock output “H” pulse width	$t_{c(SCLK)}$/2–50			ns
$t_{wL(SCLK)}$	Serial I/O clock output “L” pulse width	$t_{c(SCLK)}$/2–50			ns
$t_{d(SCLK-TXD)}$	Serial I/O output delay time (Note 1)			350	ns
$t_{v(SCLK-TXD)}$	Serial I/O output valid time (Note 1)	–30			ns
$t_{r(SCLK)}$	Serial I/O clock output rising time			50	ns
$t_{f(SCLK)}$	Serial I/O clock output falling time			50	ns
$t_{r(CMOS)}$	CMOS output rising time (Note 2)		20	50	ns
$t_{f(CMOS)}$	CMOS output falling time (Note 2)		20	50	ns

Notes 1 : When the P45/TXD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.

2 : XOUT and XCOUT pins are excluded.

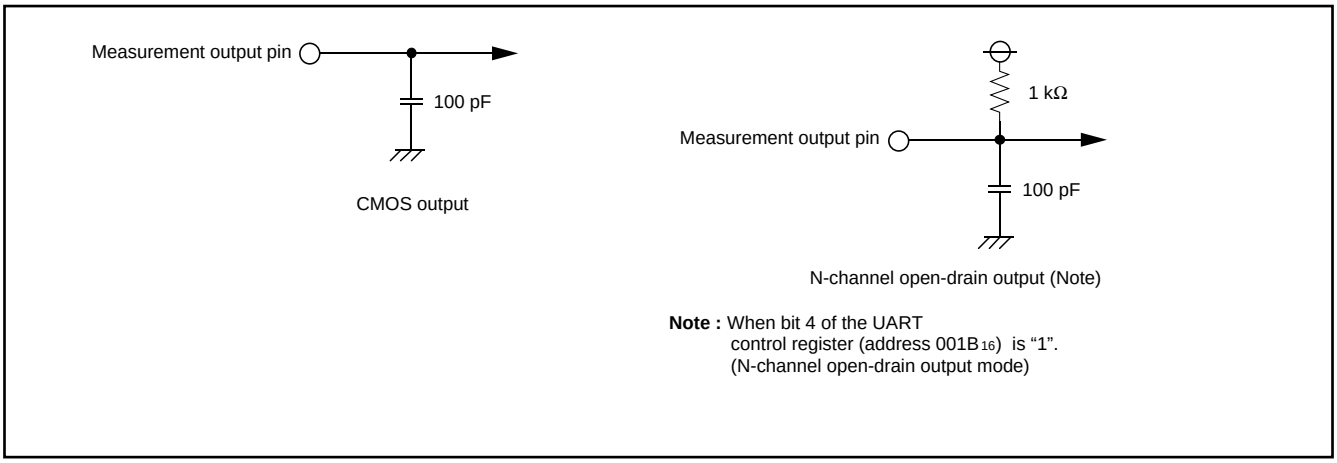


Fig. 43 Circuit for measuring output switching characteristics (1)

TIMING DIAGRAM

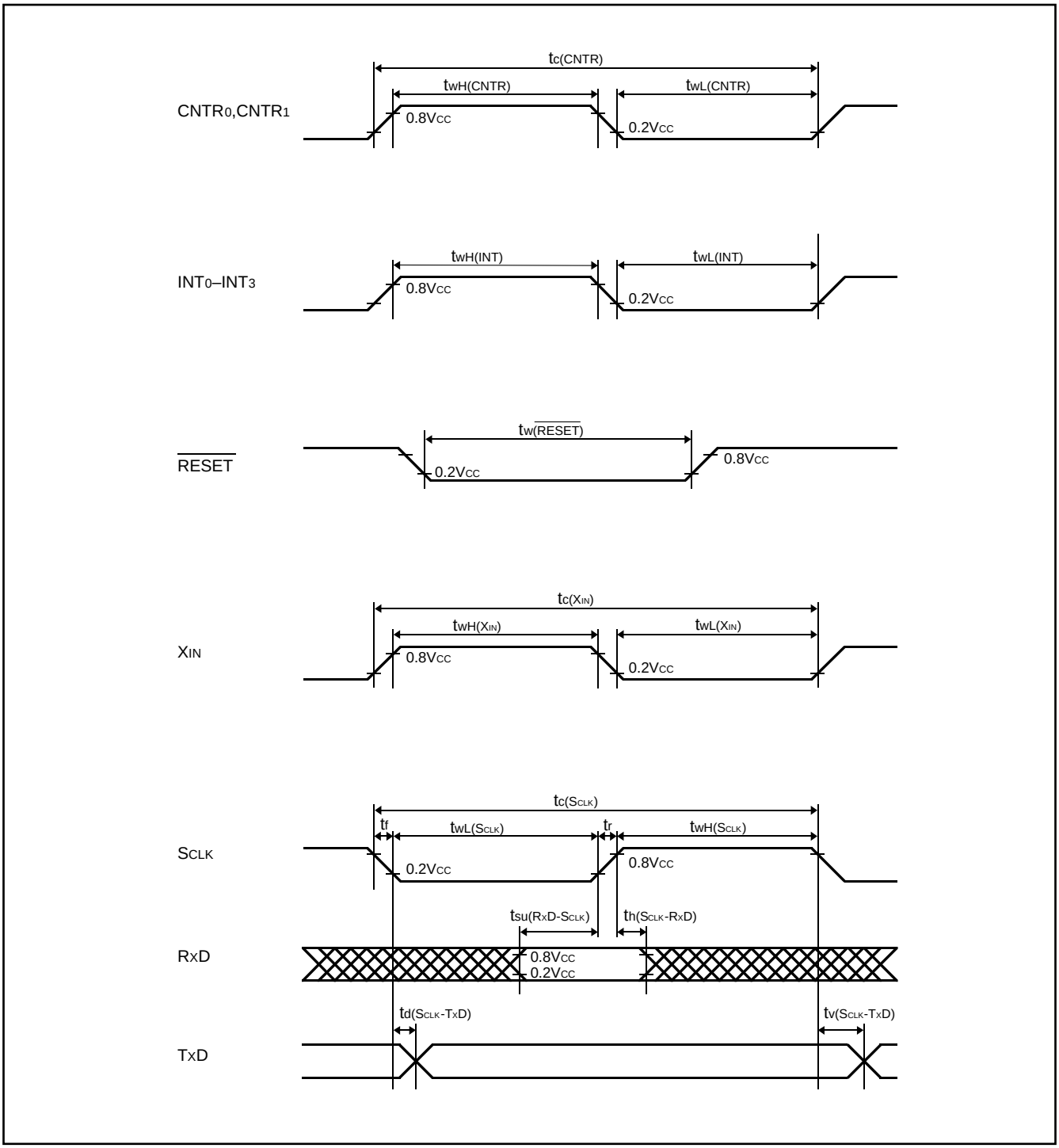


Fig. 44 Timing diagram

CHAPTER 2
APPLICATION

2.1 I/O pins

2.1.1 I/O ports

(1) I/O port write and read

■The input-only ports and programmable I/O ports set for the input mode

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

In writing data into the port register corresponding to each port, the data is only written to the port register but the pin remains in the floating state.

■Output-only ports and programmable I/O ports set for the output mode

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

In reading the data of the port transistor corresponding to each port, the pin state is not read but the value written to the port register is read. Accordingly, even if the output "H" voltage is reduced or the output "L" voltage is increased by external load, the previous output value is correctly read.

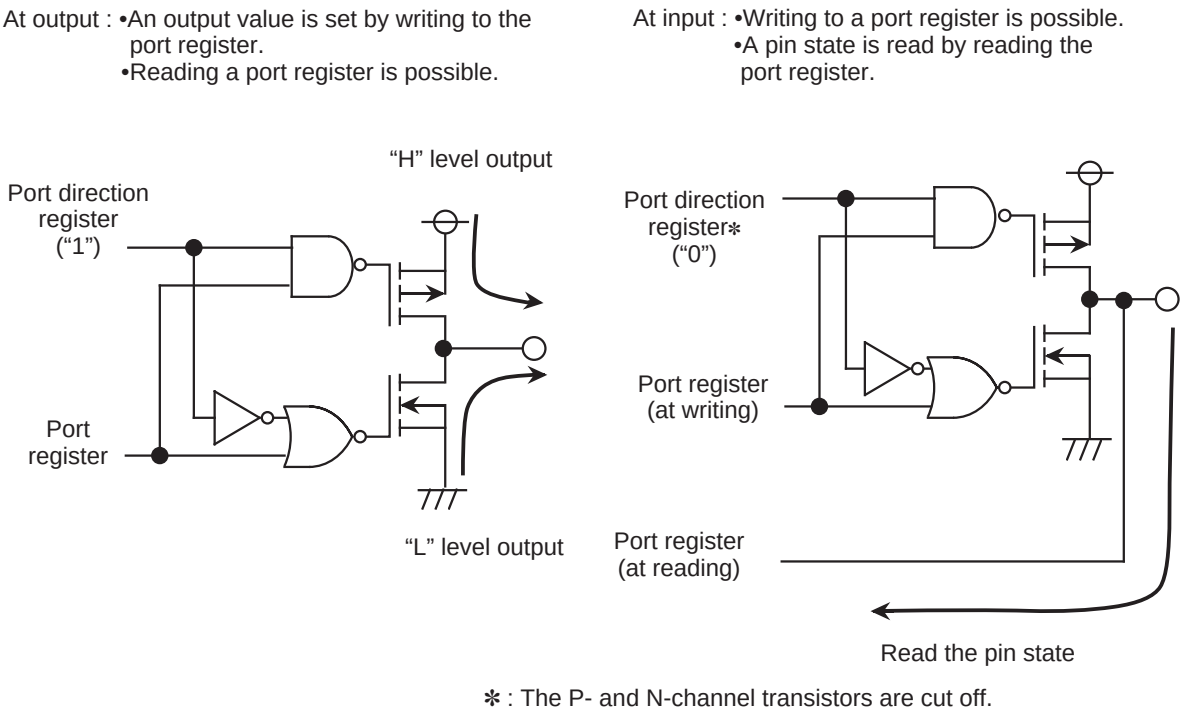


Fig. 2.1.1 I/O port write and read

Table 2.1.1 shows the memory allocation of the port registers corresponding to each port.

Table 2.1.1 Memory allocation of port registers

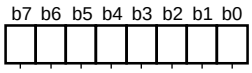
Port	Port register address
P0	0000_{16}
P1	0002_{16}
P2	0004_{16}
P3	0006_{16}
P4	0008_{16}
P5	$000A_{16}$
P6	$000C_{16}$
P7	$000E_{16}$

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

Input/output switching of the programmable I/O ports is performed by the port direction register corresponding to each port (**Note**). Figure 2.1.2 shows the structure of the port Pi (i = 2, 4 to 7) direction register, and Table 2.1.2 shows the memory allocation of the port direction registers corresponding to each port. Figure 2.1.4 shows a port direction register setting example.

Note: In ports P0 and P1, input/output switching is performed by a port unit. By setting bit 0 of the corresponding direction register to "0," the port is set for the input mode. By setting to "1," the port is set for the output mode. Figure 2.1.3 shows the structure of the ports P0 and P1 direction registers.

Port Pi direction register



Port Pi direction register (PiD) (i = 2, 4 to 7)
[Address 05_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, $0F_{16}$]

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

Notes 1: Nothing is allocated bit 0 of port P4 direction register and bit 2 to bit 7 of port P7 direction register. These bits cannot be written to.

2: The contents of the port Pi direction register cannot be read out (refer to **"2.1.4 Notes on use"**)

Fig. 2.1.2 Structure of port Pi (i = 2, 4 to 7) direction register

2.1 I/O pins

Port P0 direction register, port P1 direction register

Port P0 direction register (P0D) [Address 01 16]
Port P1 direction register (P1D) [Address 03 16]

B	Name	Functions	At reset	R	W
0	Port P0 direction register / Port P1 direction register	0 : All bits are input mode 1 : All bits are output mode	0	×	○
1 to 7	Nothing is allocated. These bits cannot be written to and be read out.		0	×	×

Note: In ports P0 and P1, input/output switching is performed by a port unit. By setting bit 0 of the corresponding port direction register to "0", the port is set for the input mode. By setting to "1", the port is set for the output mode. Nothing is allocated for bits 1 to 7 of the port P0 direction register, and these bits cannot be written to.

Fig. 2.1.3 Structure of ports P0 and P1 direction registers

Table 2.1.2 Memory allocation of port direction registers

Port	Port direction register address
P0	0001₁₆
P1	0003₁₆
P2	0005₁₆
P4	0009₁₆
P5	000B₁₆
P6	000D₁₆
P7	000F₁₆

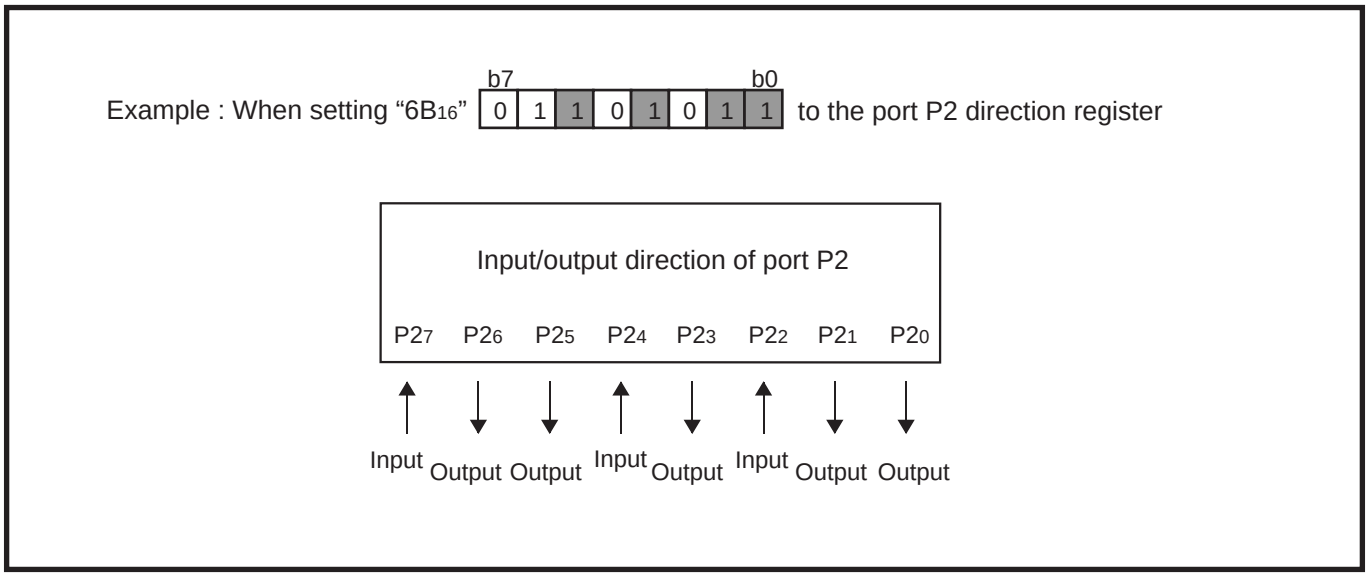


Fig. 2.1.4 Port direction register setting example

(3) Pull-up control and pull-down control

The ports shown in Table 2.1.3 are controlled for pull-up and pull-down by software. Either pull-up or pull-down is controlled by the PULL register A (address 0016_{16}) and the PULL register B (address 0017_{16}). Figure 2.1.5 shows the structure of the PULL register A and Figure 2.1.6 shows the structure of the PULL register B.

Table 2.1.3 I/O ports which either pull-up or pull-down is controlled by software

Control	Ports
Pull-down	P0, P1, P3
Pull-up	P2, $P4_1$ to $P4_7$, P5 to P7

2.1 I/O pins

PULL register A

b7 b6 b5 b4 b3 b2 b1 b0

PULL register A (PULLA) [Address 16 16]

B	Name	Functions	At reset	R	W
0	Port $P0_0$–$P0_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
1	Port $P1_0$–$P1_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
2	Port $P2_0$–$P2_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
3	Port $P3_0$–$P3_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
4	Port $P7_0$, $P7_1$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
5 to 7	Nothing is allocated. These bits cannot be written to and are fixed to “0” at reading.		0	0	×

Note: For ports set for the output mode, pull-up or pull-down is impossible.

Fig. 2.1.5 Structure of PULL register A

PULL register B

b7 b6 b5 b4 b3 b2 b1 b0

PULL register B (PULLB) [Address 17 16]

B	Name	Functions	At reset	R	W
0	Port $P4_1$-$P4_3$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
1	Port $P4_4$-$P4_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
2	Port $P5_0$-$P5_3$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
3	Port $P5_4$-$P5_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
4	Port $P6_0$-$P6_3$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
5	Port $P6_4$-$P6_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
6, 7	Nothing is allocated. These bits cannot be written to and are fixed to “0” at reading.		0	0	×

Note: For ports set for the output mode, pull-up is impossible.

Fig. 2.1.6 Structure of PULL register B

2.1.2 Function pins

Each function pin except I/O ports is described below.

(1) V_{CC} pin and V_{SS} pin

Power source input pins. In the high-speed mode, apply 5.5 V to the V_{CC} pin. In the middle-speed mode or the low-speed mode, apply 2.5 V to 5.5 V to the V_{CC} pin.
In all modes, apply 0 V to the V_{SS} pin.

(2) V_{REF} pin

The reference voltage input pin for the A-D converter. Apply 2 V to V_{CC} to the V_{REF} pin.

(3) AV_{SS} pin

The GND input pin for the A-D converter. Apply the same voltage as that applied to the V_{SS} pin to the AV_{SS} pin.

(4) V_{L1} pin, V_{L2} pin and V_{L3} pin

Power source input pins for LCD. Apply $0 \leq V_{L1} \leq V_{L2} \leq V_{L3} \leq V_{CC}$ of voltage to these pins.

(5) Pins X_{IN} and X_{OUT}

An input pin and an output pin for the main clock generating circuit.

(6) $\overline{\text{RESET}}$ pin

The 3822 group is reset internally by keeping the level of this pin at “L” for 2 μs or more. Reset state is released by returning the level of this pin to “H”.

(7) Pins SEG_0 to SEG_{11}

Segment signal output pins for LCD.

(8) Pins COM_0 to COM_3

Common signal output pins for LCD.

APPLICATION

2.1 I/O pins

2.1.3 Application examples

The basic structure for key input without a pull-up resistor and an application examples of it are described below.

In contrast to a method which uses a pull-up resistor, dissipating current incessantly, this method requires only a charging current for a very small capacitance, so it is especially suitable for a battery-driven unit. In the following description, ports A, B, C and D are only tentative names and differ from the real port names.

(1) Basic structure for key input

Figure 2.1.7 shows a connection example 1 for key input without a pull-up resistor and Figure 2.1.8 shows the key input control procedure 1. Figure 2.1.9 shows a timing diagram 1 where switch A is pressed.

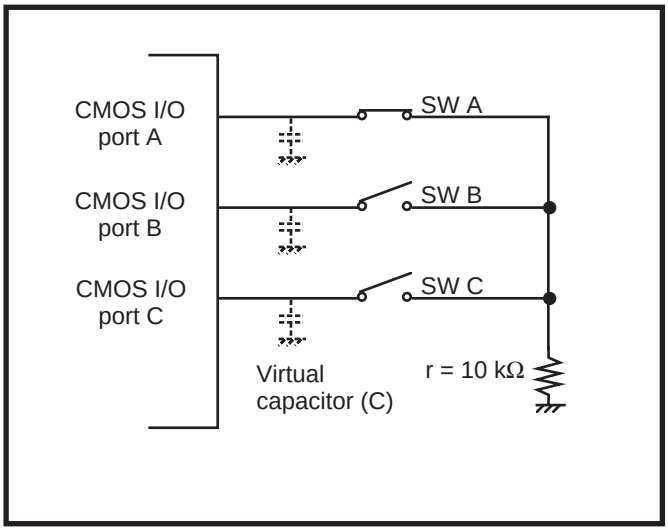


Fig. 2.1.7 Connection example 1 for key input

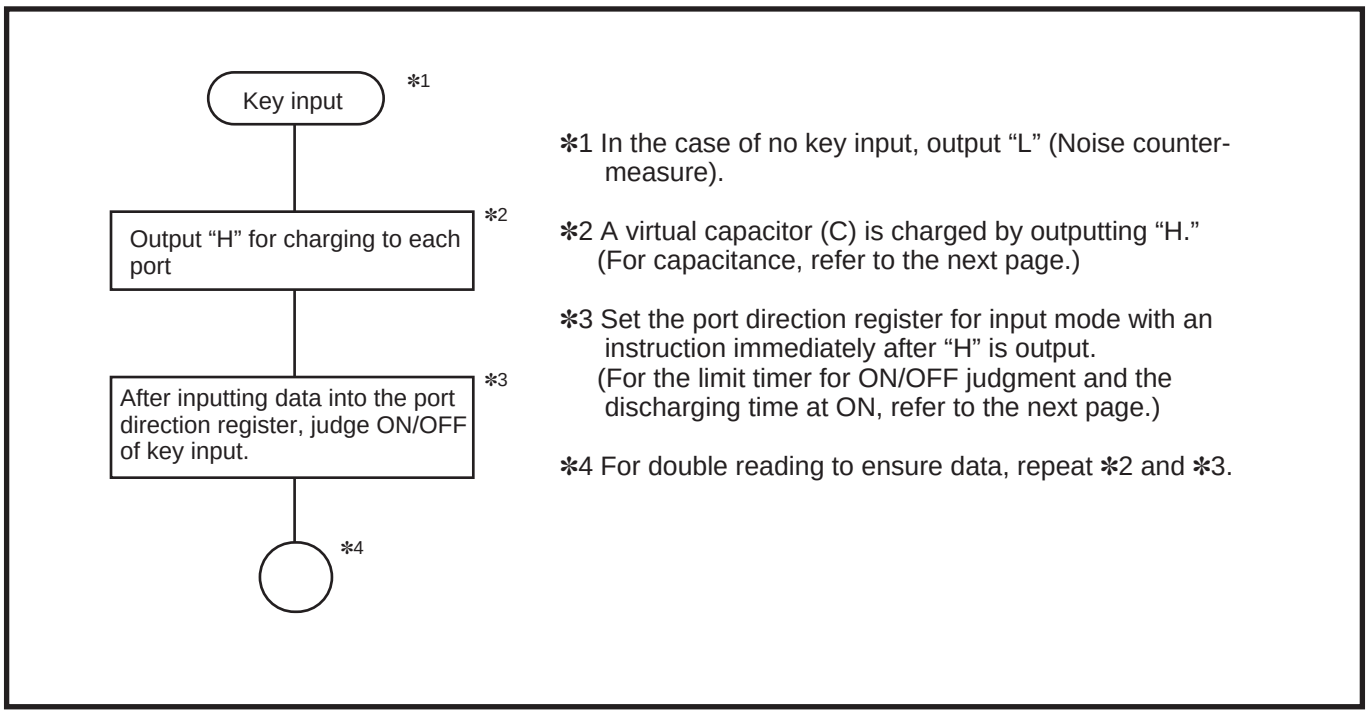


Fig. 2.1.8 Key input control procedure 1

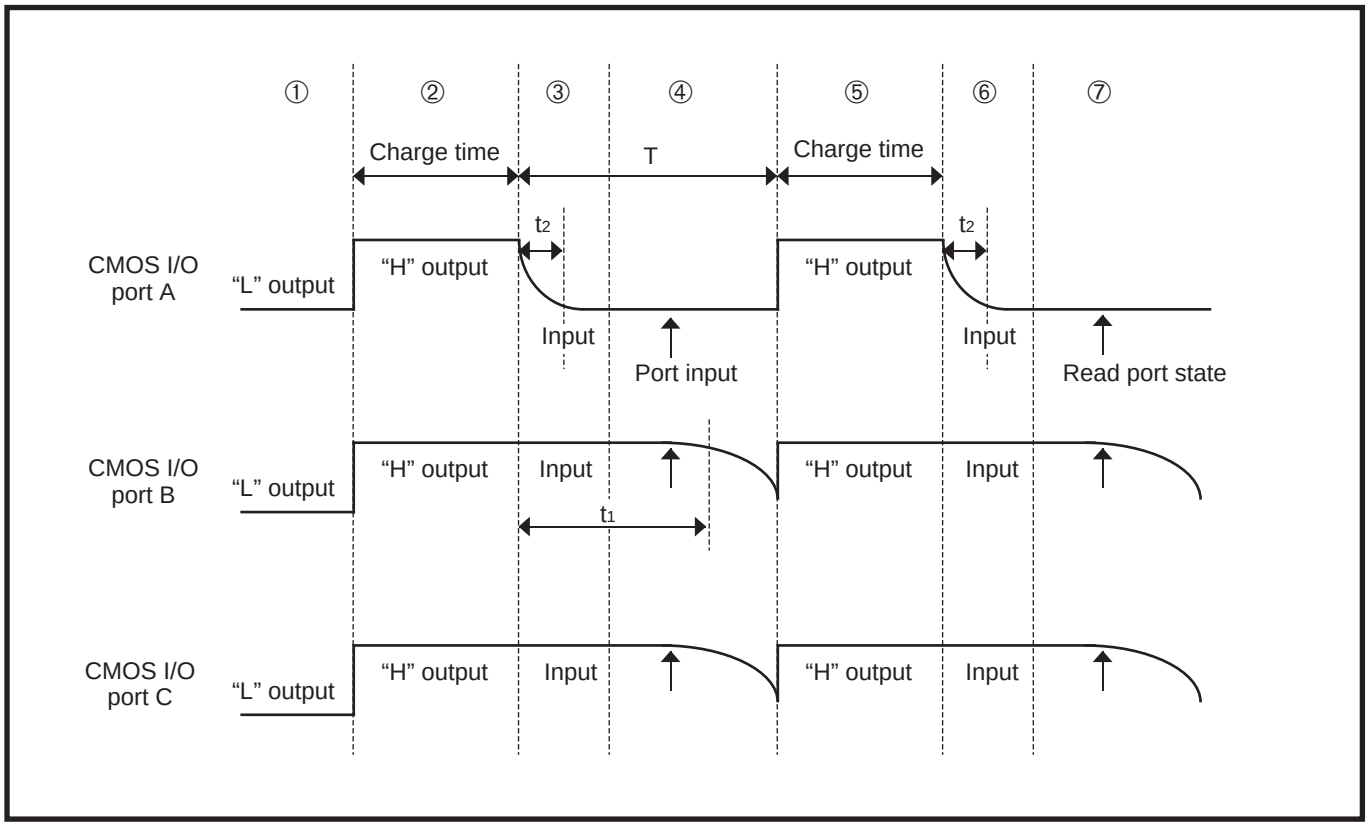


Fig. 2.1.9 Timing diagram 1 where switch A is pressed

The discharging time (③,④) after completion of charge in Figure 2.1.9 is shown with the following expression.

The discharging time (T) is obtained with T = CR.

●The capacitance of the virtual capacitor (C) is:

Capacitance of microcomputer output transistors and input transistors	Approx. 10 pF
Capacitance of package pin	Several pF
+ Capacitance of each key wiring	Several pF
	(minimum) Approx. 10 pF

●In the leak current standard at 5 V, the maximum value is 5 μA and the standard value is 0.05 μA. Accordingly, the minimum resistance (R) is 1 MΩ and the standard resistance is 100 MΩ.

In the above condition, the discharging time (T) is obtained as follows:

T (minimum) = 10 pF × 1 MΩ = 10 × 10^{-12} × 1 × 10^{6} = 10 × 10^{-6} (s)

T (standard) = 10 pF × 100 MΩ = 10 × 10^{-12} × 100 × 10^{6} = 1 × 10^{-3} (s)

Accordingly, the discharging time (T) is 10 μs (minimum) to 1 ms (standard).

∗The discharging time (t_2) at ON is obtained with t = Cr in the same way as the previous page, with the result of t = 100 ns.

∗Judge ON/OFF of key input within the time (t_1) which is obtained as follows:
After the completion of "H" output,

$$V_{t1} = V_O \times e^{-t1/T}$$

$$\downarrow$$

$$t_1 = -T \times 1n \frac{V_{t1}}{V_O}$$

V_O : "H" output voltage
V_{t1} : Input voltage after t_1(s)

<Example> The standard time at V_O = 5.0 V, V_{t1} = 3.5 V

$$t_1 = -1 \times 10^{-3} \times 1n \frac{3.5}{5.0} \fallingdotseq 357\ \mu s$$

(2) Key input application example

According to the key input without a pull-up resistor described in (1), an effective application example where there are enough ports is shown below. This method reduces both current dissipation and quantity of parts compared with the example shown in (1).
Figure 2.1.10 shows a connection example 2 for key input using port D and Figure 2.1.11 shows the key input control procedure 2. Figure 2.1.12 shows a timing diagram 2 where switch A is pressed.

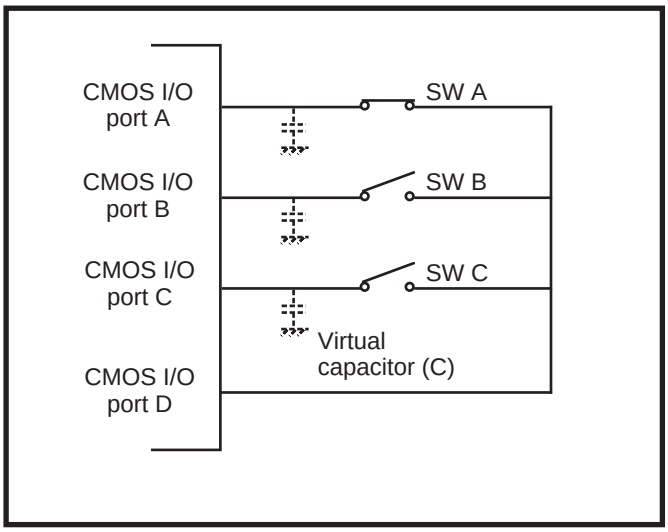


Fig. 2.1.10 Connection example 2 for key input

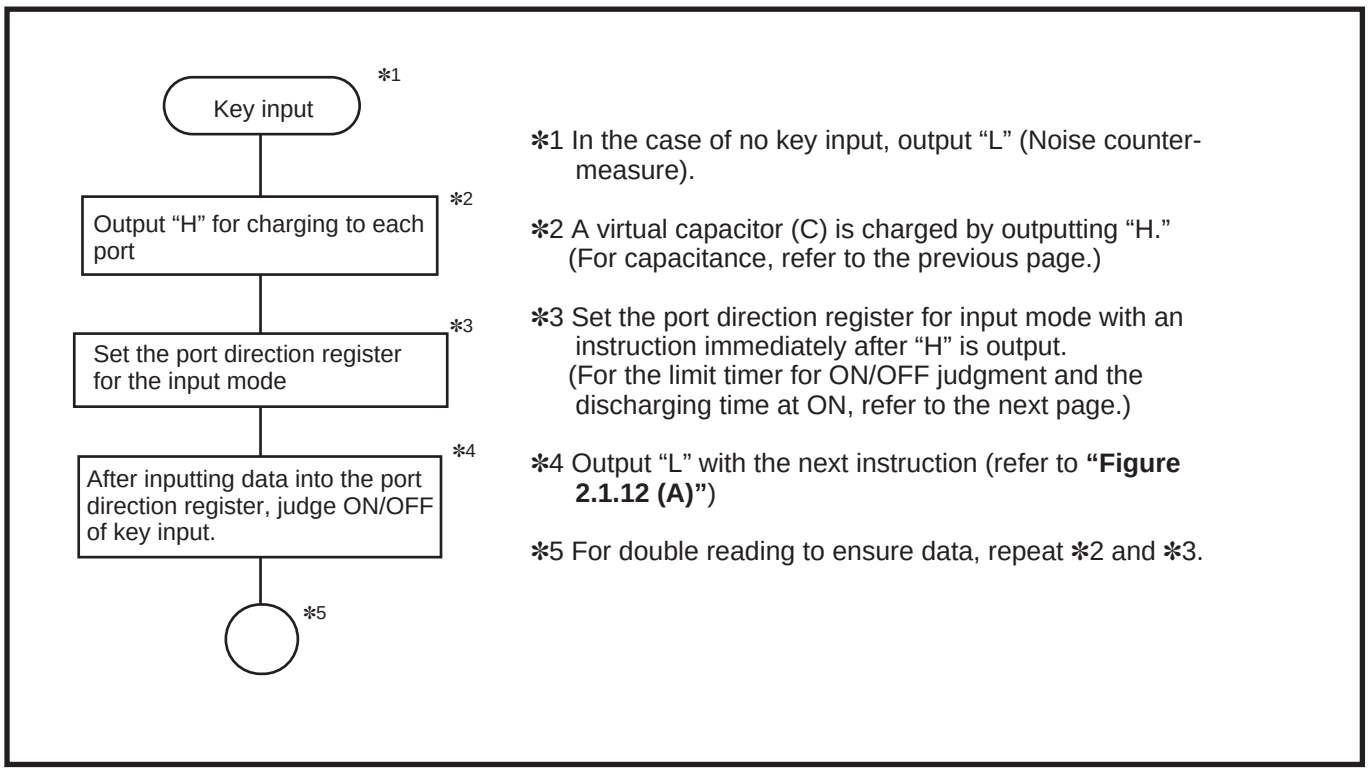


Fig. 2.1.11 Key input control procedure 2

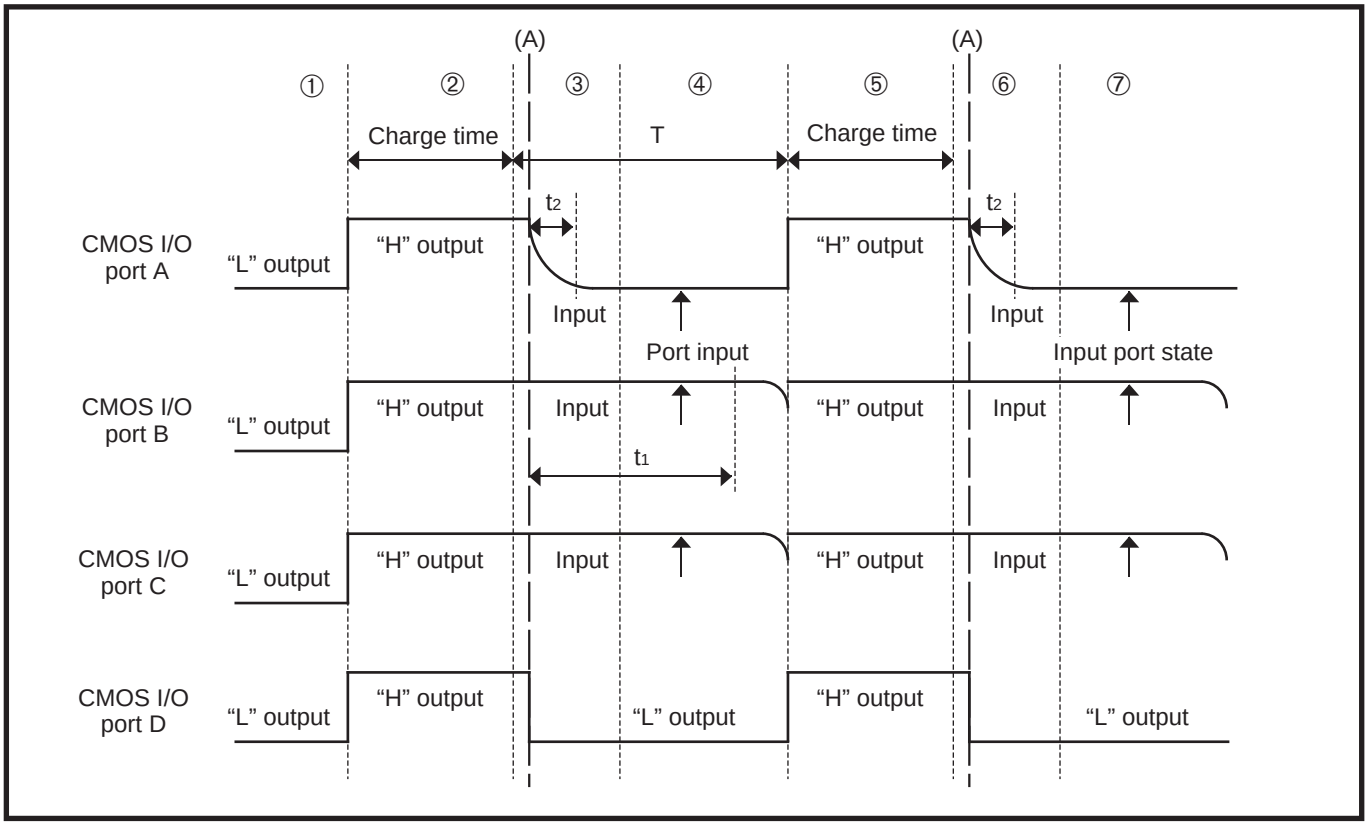


Fig. 2.1.12 Timing diagram 2 where switch A is pressed

With the exception that "L" is output using port D for key input (refer to **"Figure 2.1.12 (A)"**), the basic structure is the same as that shown in (1).
The examples shown in (1) and (2) are already put into practical use. However, be sure to evaluate them on the user's side.
In this example, the ports are the same structure as the equivalent circuit which a pull-up resistor of about 1 kΩ is connected.

2.1.4 Notes on use

When using I/O ports, note the following.

(1) Reading the port direction register

The value of the port direction register is not readable. The following cannot be used:

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

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

(2) When the data register (port latch) of an I/O port is modified with the bit managing instruction

When the data register (port latch) of an I/O port is modified with the bit managing instruction*1, the value of the unspecified bit may be changed.

REASON

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

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

*1 bit managing instructions : **SEB** and **CLB** instruction

(3) Pull-up control and pull-down control

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

- When ports P0, P1 and P3 are used as segment output pins for LCD, the settings of the pull-down bits corresponding to these ports of the PULL register A are invalid (pull-down is impossible).
- When ports P0–P2, $P4_1$–$P4_7$ and P5–P7 are set for the output mode, the settings of the bits corresponding to these ports of the PULL register A and PULL register B are invalid (pull-up or pull-down are impossible).

(4) Notes in standby state

In standby state*2 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 P-channel and 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 as an option, note on varied current values.

- When setting as an input port : Fix its input level
- When setting as an output port : Prevent current from flowing out to external

REASON

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

- P-channel.....when the content of the data register (port latch) is "0"
- N-channel.....when the content of the data register (port latch) is "1"

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

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

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

2.1 I/O pins

(5) Termination of unused pins

Table 2.1.4 shows termination of unused pins.

Table 2.1.4 Termination of unused pins

Pins	Terminations
$P2_0$–$P2_7$	①After set for the input mode and put the built-in pull-up resistor in the ON state, open.*1 ②Set for the output mode and open at "L" or "H".*2
$P4_1$/ϕ	
$P4_4$/RxD	
$P4_5$/TxD	
$P4_6$/SCLK	
$P4_7$/$\overline{SRDY}$	
$P5_2$/RTP_0	
$P5_3$/RTP_1	
$P5_4$/$CNTR_0$	
$P5_5$/$CNTR_1$	
$P5_6$/TOUT	
$P5_7$/ADT	
$P6_0$/AN_0–$P6_7$/AN_7	
$P7_0$/XCOUT	
$P7_1$/XCIN	
$P0_0$/SEG_{16}–$P0_7$/SEG_{23}	①After set for the input mode and put the built-in pull-down resistor in the ON state, open.*1 ②Set for the output mode and open at "L" or "H".*2
$P1_0$/SEG_{24}–$P1_7$/SEG_{31}	
$P4_0$	Connect each pin to VCC or VSS through each resistor of 1kΩ to 10 kΩ.
$P4_2$/INT_0	①After disabling INT interrupts, set for the input mode, and put the pull-up built-in resistor in the ON state, open.*1 ②Set for the output mode and open at "L" or "H".*2
$P4_3$/INT_1	
$P5_0$/INT_2	
$P5_1$/INT_3	
V_{L1}–V_{L3}	Connect to VSS level
COM_0–COM_3	Open
SEG_0–SEG_{11}	
$P3_4$/SEG_{12}–$P3_7$/SEG_{15}	

✻1 After reset and before the built-in pull-up (pull-down) resistor is put in the ON state by software, the built-in pull-up (pull-down) resistor is in the OFF state. Because of this, the potential at these pins are "undefined" and the power source current may increase. Since the direction register setup may be changed for the output mode because of a program runaway or noise, set direction register for the input mode periodically. And make the length of wiring which is connected I/O ports within 2 cm.

✻2 After reset and before I/O ports are switched for the output mode by software, I/O ports are set for the input mode. Because of this, the potential at these pins are "undefined" and the power source current may increase in the input mode. Since the direction register setup may be changed for the input mode because of a program runaway or noise, set direction register for the output mode periodically. And make the length of wiring which is connected I/O ports within 2 cm.

2.2 Interrupts

2.2.1 Explanation of operations

When an interrupt request is accepted, the contents immediately before acceptance of the interrupt requests of the following registers is automatically pushed onto the stack area in the order of ①, ② and ③.
①High-order (PC_H) contents of program counter
②Low-order (PC_L) contents of program counter
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 processing executed immediately before acceptance of the interrupts is continued.
Figure 2.2.1 shows an interrupt operation diagram.

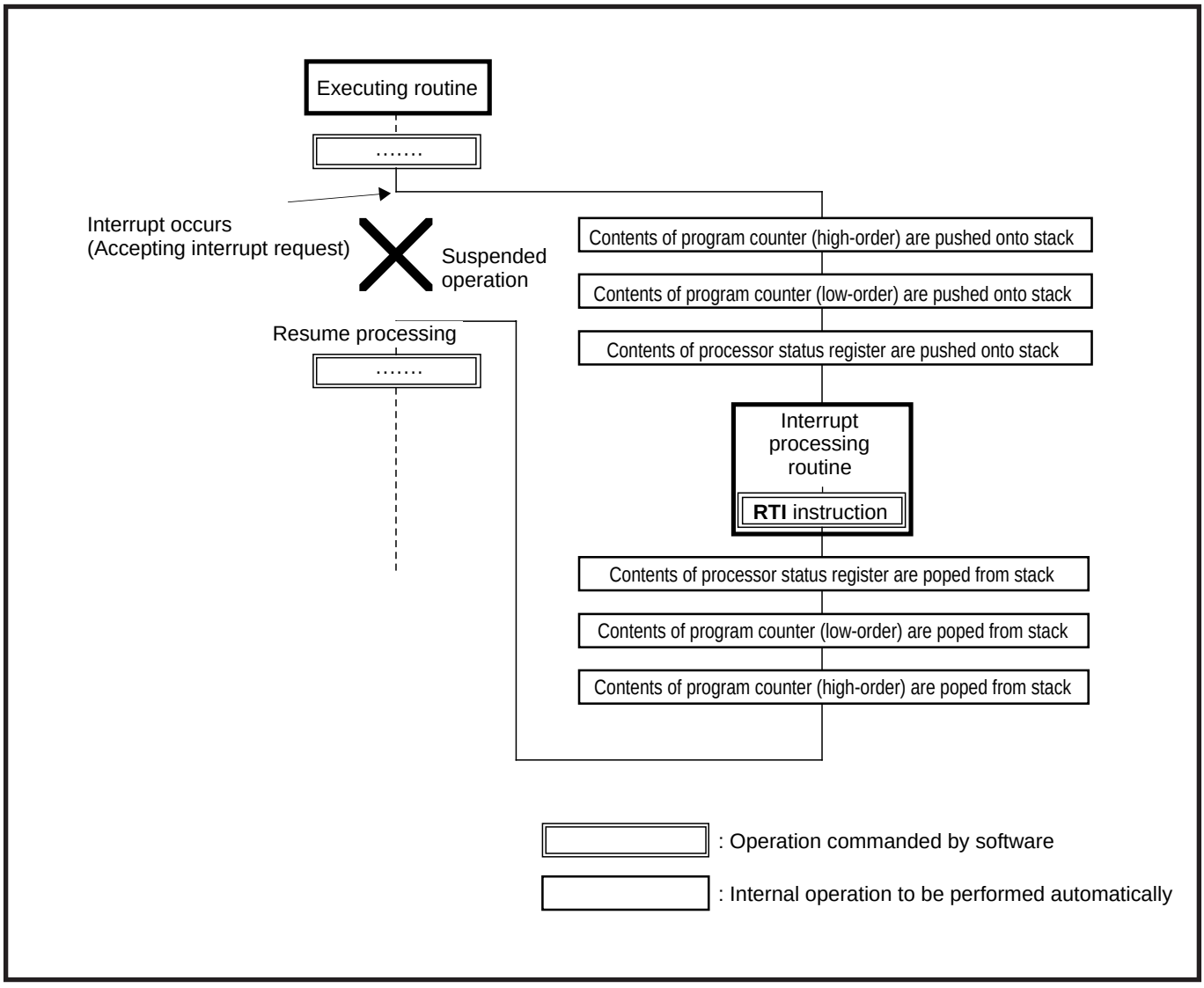


Fig. 2.2.1 Interrupt operation diagram

2.2 Interrupts

(1) Interrupt request generating conditions

Table 2.2.1 shows interrupt sources and interrupt request generating conditions.
The occurrence of an interrupt request causes the corresponding interrupt request bit to be set to "1."
When the following conditions are satisfied in this state, the interrupt request is accepted.
For details, refer to "**2.2.2 Control**".
①Interrupt disable flag = "0" (interrupts enabled)
②Interrupt enable bit = "1" (interrupts enabled)

Table 2.2.1 Interrupt sources and interrupt request generating conditions

Interrupt sources	Interrupt request generating conditions	Reference
INT_0	At detection of either rising or falling edge of INT_0 input (Active edge selectable)	2.2.4 INT interrupts
INT_1	At detection of either rising or falling edge of INT_1 input (Active edge selectable)	
Serial I/O receive	At completion of serial I/O data reception	2.5 Serial I/O
Serial I/O transmit	At completion of serial I/O transmit shift or when transmit buffer is empty	
Timer X	At timer X underflow	2.3 Timer X and timer Y
Timer Y	At timer Y underflow	
Timer 2	At timer 2 underflow	2.4 Timer 1, timer 2, and timer 3
Timer 3	At timer 3 underflow	
$CNTR_0$	At detection of either rising or falling edge of $CNTR_0$ input (Active edge selectable)	2.3 Timer X and timer Y
$CNTR_1$	At detection of either rising or falling edge of $CNTR_1$ input (Active edge selectable)	
Timer 1	At timer 1 underflow	2.4 Timer 1, timer 2, and timer 3
INT_2	At detection of either rising or falling edge of INT_2 input (Active edge selectable)	2.2.4 INT interrupts
INT_3	At detection of either rising or falling edge of INT_3 input (Active edge selectable)	
Key input (Key-on wake up)	At falling of conjunction of input level for port P2 (at input mode)	2.2.5 Key input interrupt
ADT A-D conversion	At detection of falling edge of ADT input At completion of A-D conversion	2.6 A-D converter
BRK instruction	At BRK instruction execution	SERIES 740 <SOFTWARE> USER'S MANUAL

(2) Processing upon acceptance of an 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 2.2.2 shows 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.

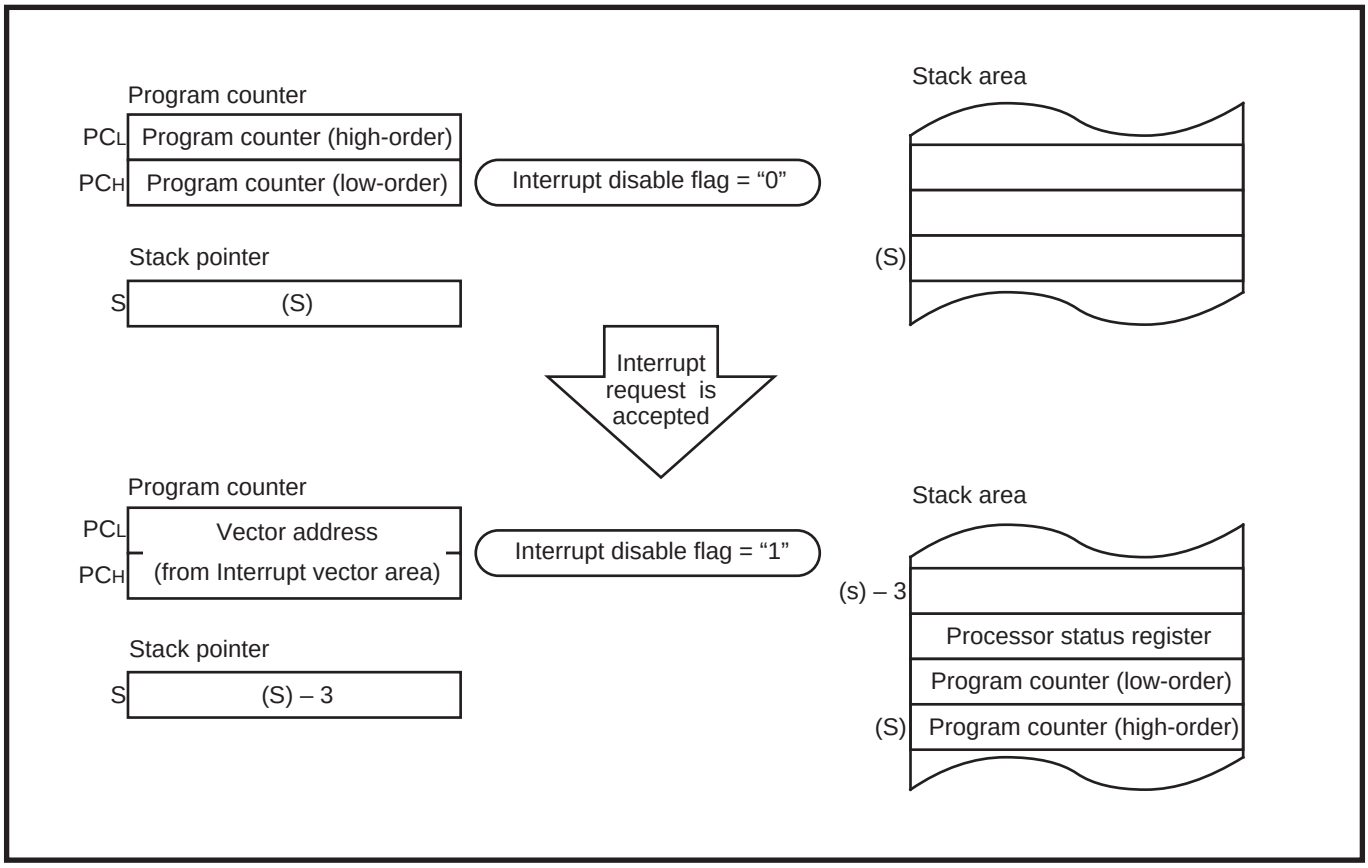
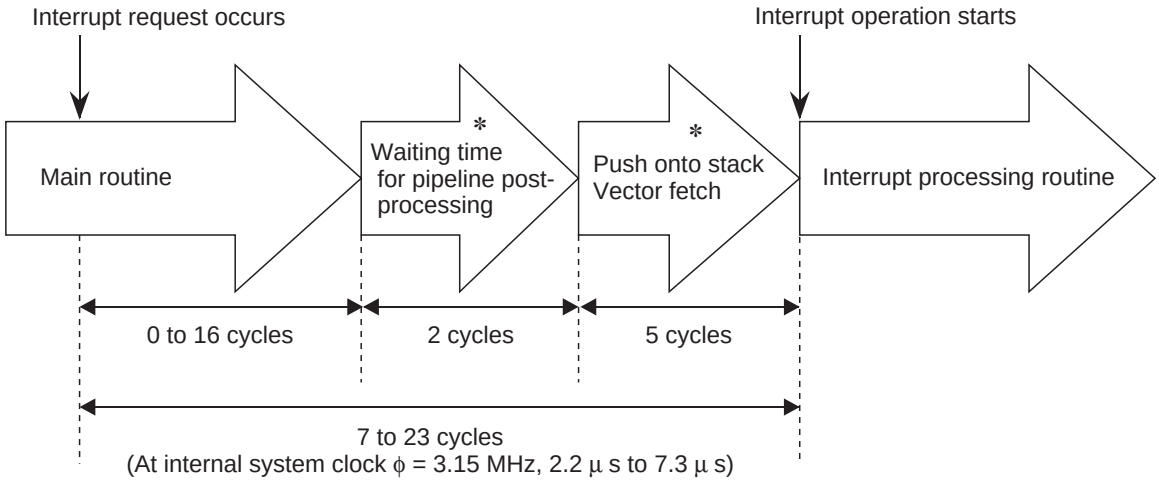


Fig. 2.2.2 Changes of stack pointer and program counter upon acceptance of interrupt request

2.2 Interrupts

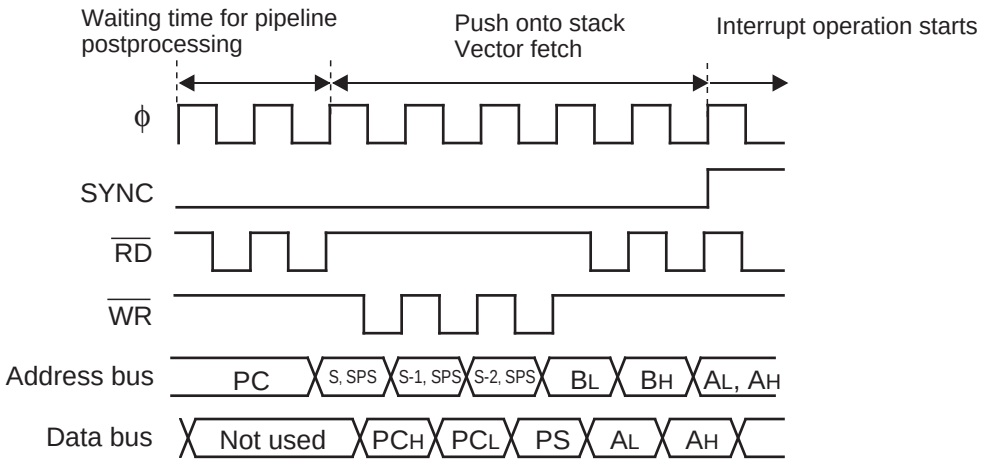
(3) Timing after acceptance of an interrupt request

The interrupt processing routine is started at the timing of machine cycle after completion of the executing instruction. Figure 2.2.3 shows the processing time up to the execution of an interrupt processing routine and Figure 2.2.4 shows timing after the acceptance of an interrupt request.



✻ : Refer to **"Figure 2.2.4"**

Fig. 2.2.3 Processing time up to execution of interrupt processing routine



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

Note: Refer to **"Table 6 in CHAPTER 1 HARDWARE ."**

Fig. 2.2.4 Timing after acceptance of interrupt request

2.2.2 Control

For interrupts except the **BRK** instruction interrupt, the acceptance of interrupt can be controlled by an interrupt request bit, an interrupt enable bit, and an interrupt disable flag. In this section, control of interrupts except the **BRK** instruction interrupt is described and Figure 2.2.5 shows an interrupt control diagram.

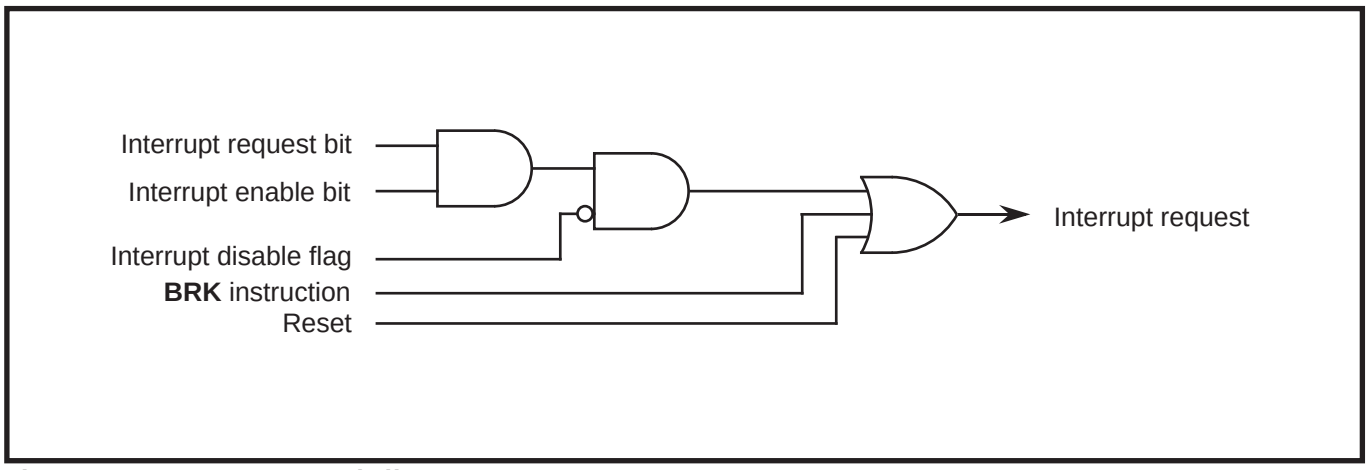


Fig. 2.2.5 Interrupt control diagram

An interrupt request bit, an interrupt enable bit and an 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 software, 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 bits for individual interrupt sources.

(1) Interrupt request bits

The interrupt request bits are allocated to the interrupt request register 1 (address $003C_{16}$) and interrupt request register 2 (address $003D_{16}$).

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" by software, but it 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 $003F_{16}$).

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 only set to "1" and this 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," this interrupt is accepted (at 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.

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 2.2.6 shows an example of multiple interrupts.

Table 2.2.2 List of interrupt bits for individual interrupt sources

Interrupt sources	Interrupt request bit		Interrupt enable bit	
	Address	Bit	Address	Bit
INT_0	$003C_{16}$	b0	$003E_{16}$	b0
INT_1	$003C_{16}$	b1	$003E_{16}$	b1
Serial I/O reception	$003C_{16}$	b2	$003E_{16}$	b2
Serial I/O transmission	$003C_{16}$	b3	$003E_{16}$	b3
Timer X	$003C_{16}$	b4	$003E_{16}$	b4
Timer Y	$003C_{16}$	b5	$003E_{16}$	b5
Timer 2	$003C_{16}$	b6	$003E_{16}$	b6
Timer 3	$003C_{16}$	b7	$003E_{16}$	b7
$CNTR_0$	$003D_{16}$	b0	$003F_{16}$	b0
$CNTR_1$	$003D_{16}$	b1	$003F_{16}$	b1
Timer 1	$003D_{16}$	b2	$003F_{16}$	b2
INT_2	$003D_{16}$	b3	$003F_{16}$	b3
INT_3	$003D_{16}$	b4	$003F_{16}$	b4
Key input	$003D_{16}$	b5	$003F_{16}$	b5
ADT/A-D conversion	$003D_{16}$	b6	$003F_{16}$	b6

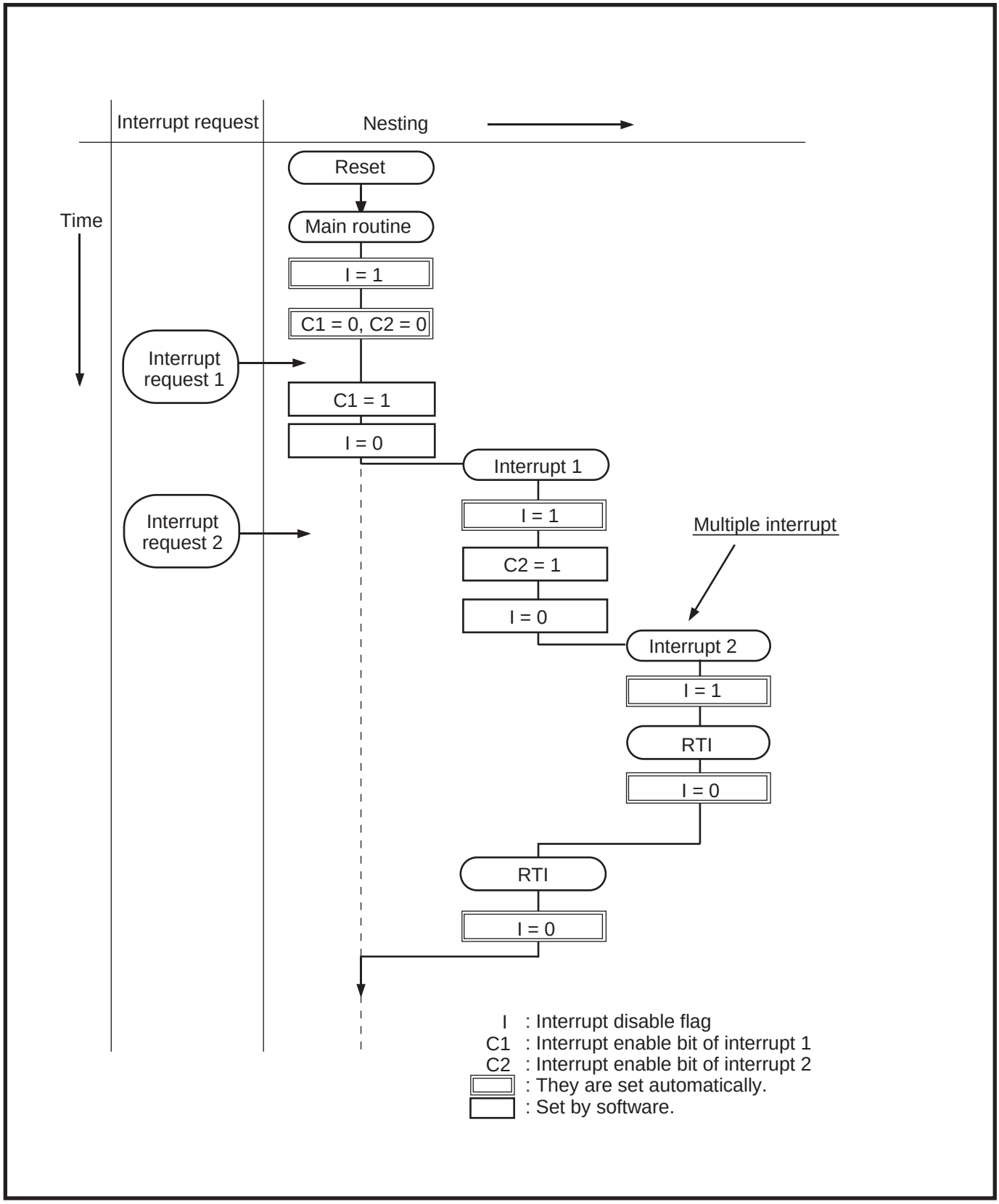


Fig. 2.2.6 Example of multiple interrupts

2.2 Interrupts

2.2.3 Related registers

Figure 2.2.7 shows memory allocation of interrupt-related registers. Each of these registers is described below.

Address	
003A16	Interrupt edge selection register (INTEDGE)
003B16	
003C16	Interrupt request register 1 (IREQ1)
003D16	Interrupt request register 2 (IREQ2)
003E16	Interrupt control register 1 (ICON1)
003F16	Interrupt control register 1 (ICON2)

Fig. 2.2.7 Memory allocation of interrupt-related registers

(1) Interrupt edge selection register (address 003A16)

The interrupt edge selection register selects an active edge of each INT interrupt.

Bit 0 to bit 3 select active edges of INT0–INT3 pins inputs. In the "0" state, the falling edge (↓) of the corresponding pin input is active. In the "1" state, the rising edge (↑) of the corresponding pin input is active. Figure 2.2.8 shows the structure of the interrupt edge selection register.

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register (INTEDGE) [Address 3A 16]

B	Name	Functions	At reset	R	W
0	INT0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	INT2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	INT3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4 to 7	Nothing is allocated. These bits cannot be written to and are fixed to "0" at reading.		0	0	×

Fig. 2.2.8 Structure of interrupt edge selection register

(2) Interrupt request register 1 (IREQ1) and interrupt request register 2 (IREQ2)

The interrupt request register 1 (address 003C16) and the interrupt request register 2 (address 003D16) indicate whether an interrupt request has occurred or not.

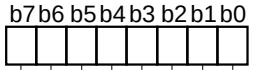
Figure 2.2.9 shows the structure of the interrupt request register 1 and Figure 2.2.10 shows the structure of the interrupt request register 2.

The occurrence of an interrupt request causes the corresponding bit to be set to "1." This interrupt request bit is automatically cleared to "0" by the acceptance of the interrupt request.

The interrupt request bits can be set to "0" by software, but it cannot be set to "1" by software.

The occurrence of each interrupt is controlled by the interrupt enable bits (refer to the next item).

Interrupt request register 1



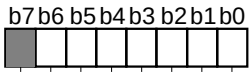
Interrupt request register 1 (IREQ1) [Address 3C 16]

B	Name	Functions	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT_1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

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

Fig. 2.2.9 Structure of interrupt request register 1

Interrupt request register 2



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

B	Name	Functions	At reset	R	W
0	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key input interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	ADT/A-D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated. This bit cannot be written to and is fixed to “0” at reading.		0	0	×

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

Fig. 2.2.10 Structure of interrupt request register 2

(3) Interrupt control register 1 (ICON1) and interrupt control register 2 (ICON2)

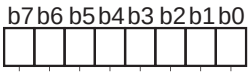
The interrupt control register 1 (address 003E16) and the interrupt control register 2 (address 003F16) control each interrupt request source.

Figure 2.2.11 shows the structure of the interrupt control register 1 and Figure 2.2.12 shows the structure of the interrupt control register 2.

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 only set to "1," and the interrupt request is not accepted.

When an interrupt enable bit is "1," the corresponding interrupt request is enabled. If an interrupt request occurs when this bit is "1," the interrupt request is accepted (at interrupt disable flag = "0"). Each interrupt enable bit can be set to "0" or "1" by software.

Interrupt control register 1



Interrupt control register 1 (ICON1) [Address 3E16]

B	Name	Functions	At reset	R	W
0	INT0 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	INT1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	Timer 2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Timer 3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○

Fig. 2.2.11 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	Functions	At reset	R	W
0	$CNTR_0$ interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	$CNTR_1$ interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Timer 1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	INT_2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Key input interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	ADT/A-D conversion interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Fix this bit to “0.”		0	0	0

Fig. 2.2.12 Structure of interrupt control register 2

(4) Processor status register

The processor status register is an 8-bit register. Figure 2.2.13 shows the structure of the processor status register. Bit 2 related to an interrupt is described below.

■Interrupt disable flag : bit 2

The interrupt disable flag controls the acceptance of interrupt requests except **BRK** instruction interrupt. When this flag is "1," the acceptance of an interrupt request is disabled. When this flag is "0," the acceptance of an interrupt request 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.

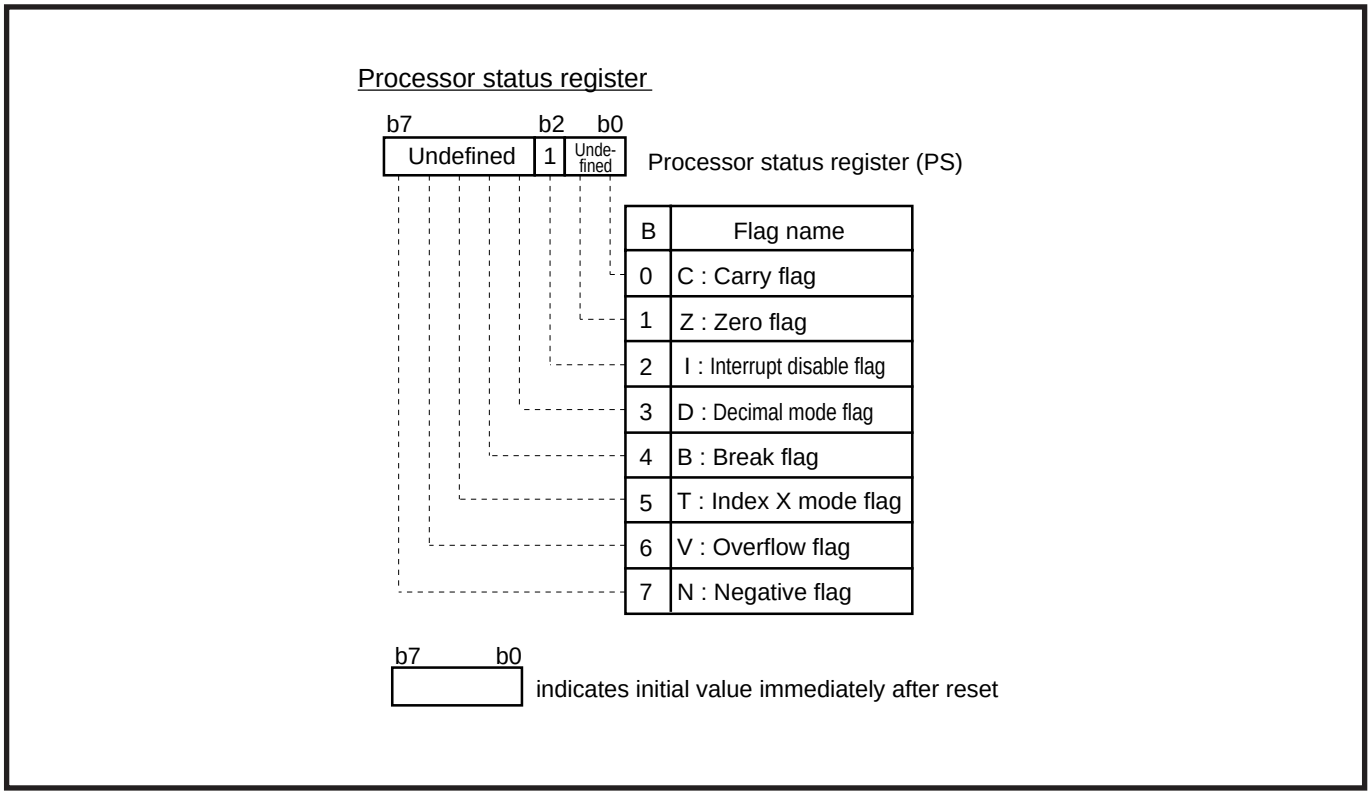


Fig. 2.2.13 Structure of processor status register

2.2.4 INT interrupts

The INT interrupt requests occur by detecting a level change of each INT pin (INT_0–INT_3).

(1) Active edge selection

As an active edge, falling edge (↓) detection or rising edge (↑) detection can be selected by bits 0 to 3 of the interrupt selection register (address $003A_{16}$).

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.

The pins INT_0 to INT_3 are also used as I/O ports $P4_2$, $P4_3$, $P5_0$, and $P5_1$, but no register to switch between INT pin and I/O port is available. When the port is an input port, the active edges of the port are always detected. Accordingly, when using ports $P4_2$, $P4_3$, $P5_7$ and $P6_0$ as input ports, put the corresponding INT interrupt into the disabled state. If this interrupt is not disabled, an INT interrupt is caused by pin level change, so that the program runs away.

Figure 2.2.14 shows the structure of the interrupt edge selection register.

Interrupt edge selection register

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

B	Name	Functions	At reset	R	W
0	INT_0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT_1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	INT_2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	INT_3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4 to 7	Nothing is allocated. These bits cannot be written to and are fixed to "0" at reading.		0	0	×

Fig. 2.2.14 Structure of interrupt edge selection register

2.2.5 Key input interrupt

The Key input interrupt request occurs when an “L” level voltage is applied to the pin set for the input mode of the port P2.

For interrupt sources except the INT interrupts and the Key input interrupt, refer to **“CHAPTER 1”**.

(1) Connection example when the Key input interrupt is used

When using the Key input interrupt, after set ports $P2_0$ to $P2_3$ for the input mode, configure an “L” level valid key-matrix.

Figure 2.2.15 shows a connection example when the key input interrupt is used, and a port P2 block diagram. In the connection example in Figure 2.2.15, an Key input interrupt request is caused by pressing the key corresponding to one of ports $P2_0$ to $P2_3$.

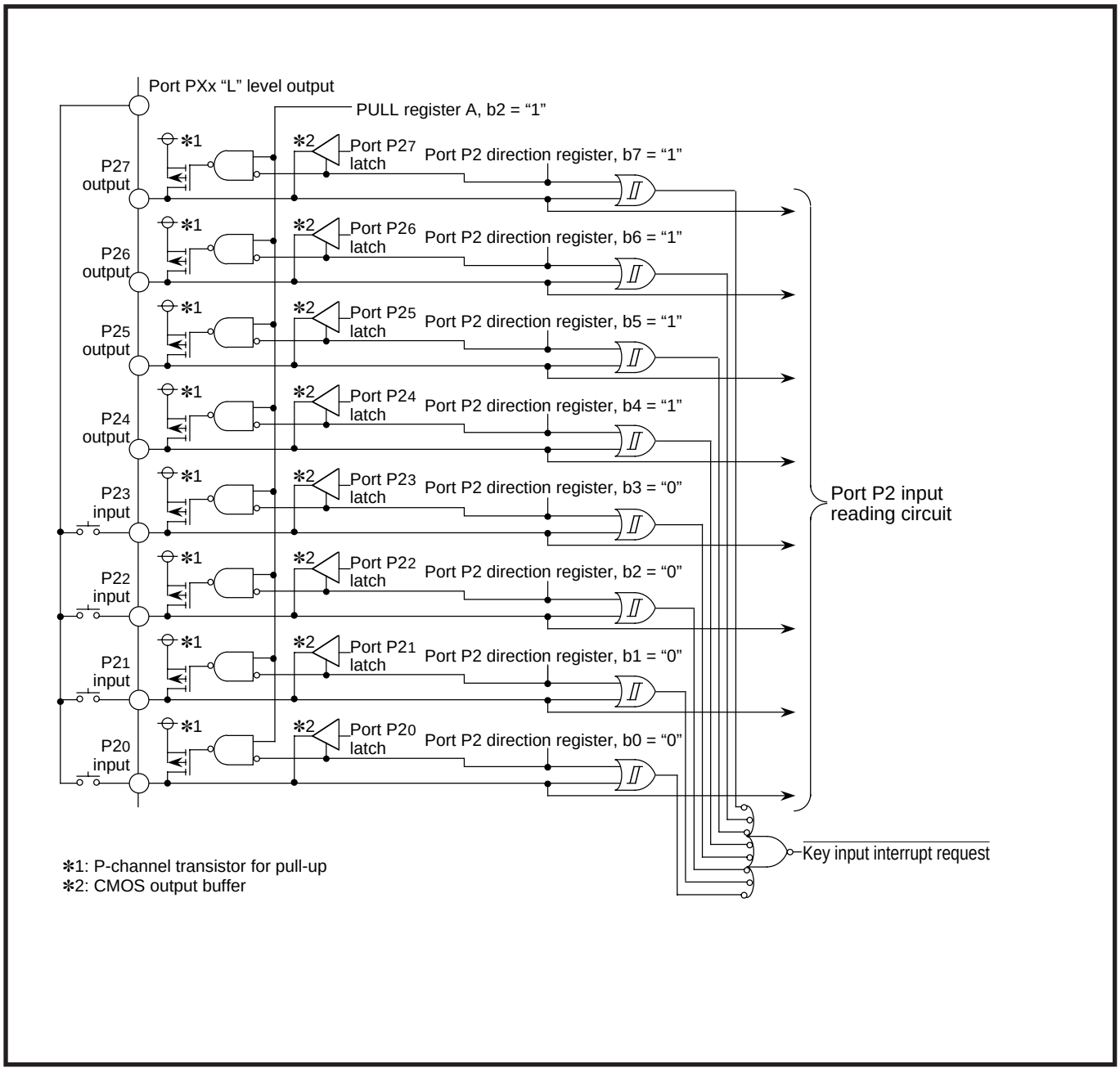


Fig. 2.2.15 Connection example when key input interrupt is used, and port P2 block diagram

(2) Set values of Key input interrupt-related registers

When using the Key input interrupt, set the following:

●Port P2 direction register (address 0005_{16})
●Bit 2 of PULL register A (address 0016_{16})
●Bit 5 of interrupt request register 2 (address $003D_{16}$) = "0"
●Bit 5 of interrupt control register 2 (address $003F_{16}$) (Note) = "1"

Figure 2.2.16 shows the setting values (corresponding to Figure 2.2.15) of the Key input interrupt-related registers.

Note: Fix bit 7 of the interrupt control register 2 (address $003F_{16}$) to "0".

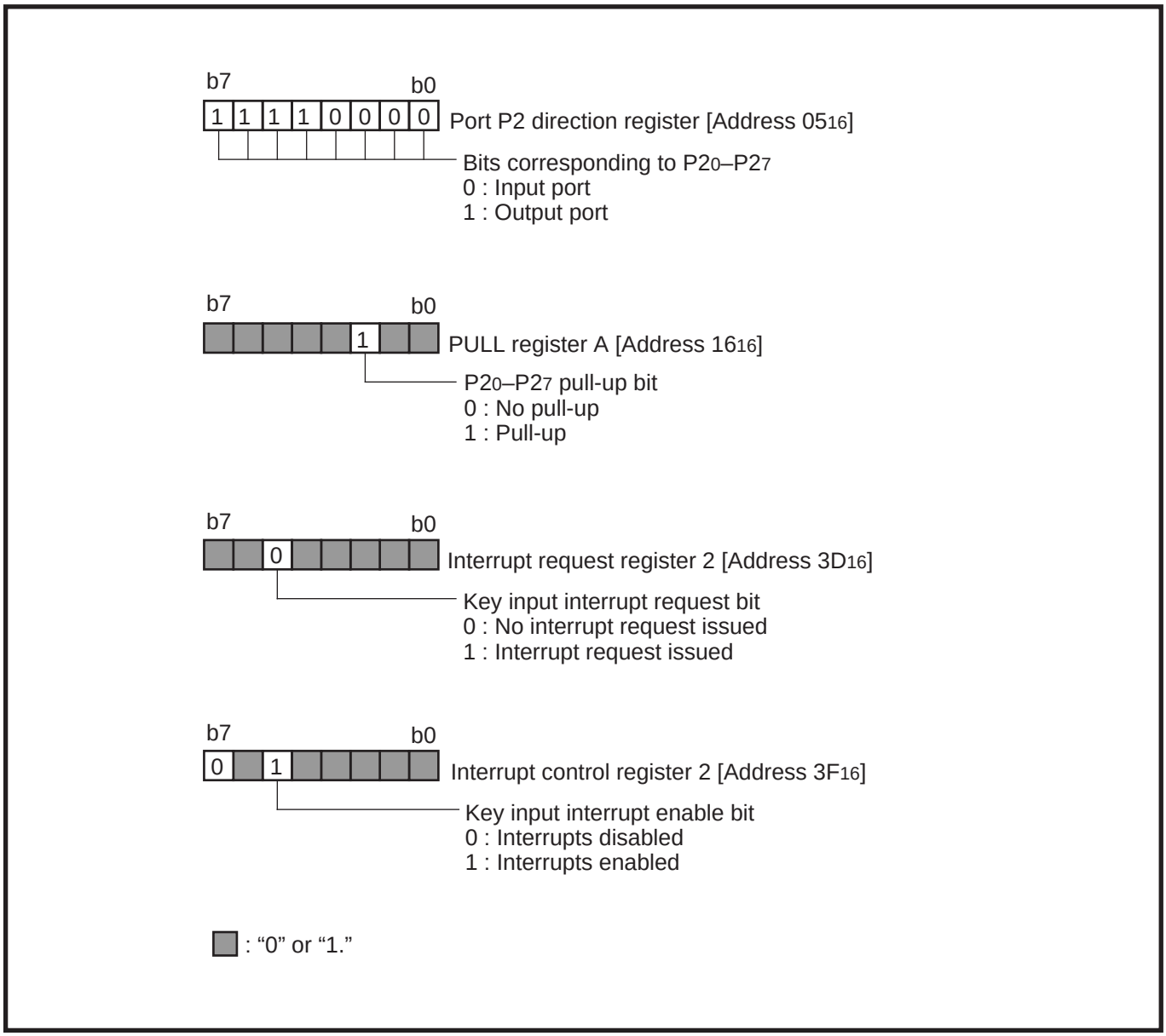


Fig. 2.2.16 Setting values (corresponding to Figure 2.2.15) of key input interrupt-related registers

2.2.6 Notes on use

When using interrupts, note the following.

(1) Register setting

■Fix bit 7 of the interrupt control register 2 (address $003F_{16}$) to "0." Nothing is allocated for this bit, however, do not write "1" to it.

■When using I/O ports $P4_2$, $P4_3$, $P5_0$ and $P5_1$ as input ports, put the INT interrupts corresponding to each port into the disabled state.

■When the active edges of the following interrupts are switched, the corresponding interrupt request bit may be set to "1." To avoid accepting an interrupt request, we recommend the register setting example shown in Figure 2.2.17.

●INT_0 interrupt to INT_3 interrupt

●$CNTR_0$ interrupt and $CNTR_1$ interrupt

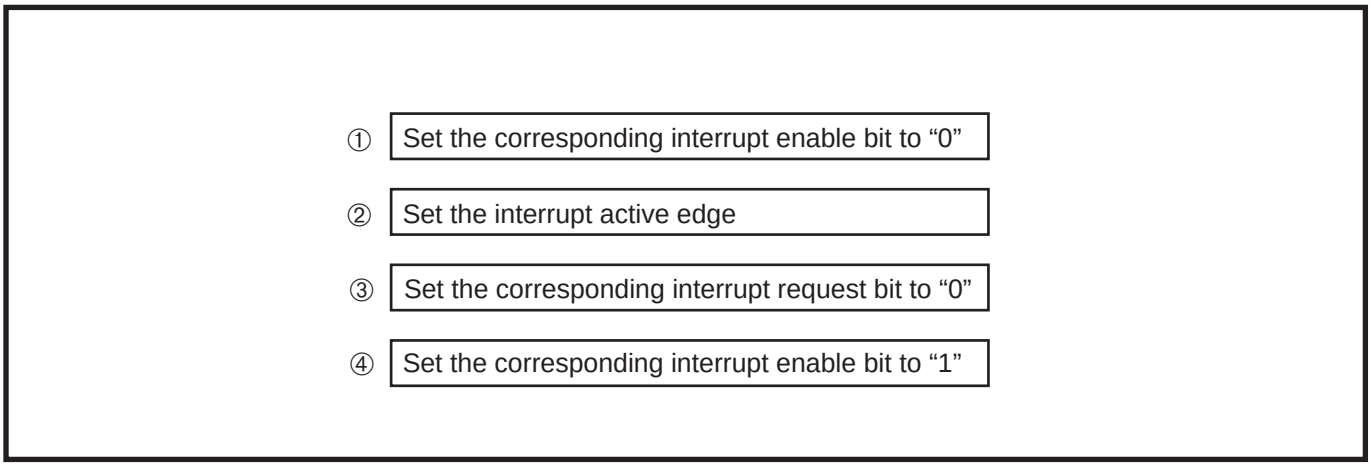


Fig. 2.2.17 Register setting example

2.3 Timer X and timer Y

2.3.1 Explanation of timer X operations

Timer X has 4 modes of operation.
Operation in each mode is described below.

(1) Timer Mode

Operation in the timer mode is described below.

①Start of count operation

Immediately after reset release, the timer X stop control bit is in the "0" state. For this reason, the count operation is automatically started after reset release.

The value of the timer X counter (referred as "the X counter") is decremented by 1 each time a count source is input.

The count source is f(XIN)/16 clock (low-speed mode ; f(XCIN)/16 clock).

②Reload operation

The X counter underflows at the first count pulse after the value of the X counter reaches "00_{16}." At this time, the value of the timer X latch (referred as "the X latch") is transferred (reloaded) to the X counter.

③Interrupt operation

An interrupt request occurs at the X counter underflow. At the same time, the timer X interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the timer X interrupt enable bit.

An interrupt request occurs each time the counter underflows. In other words, an interrupt request occurs every "the X counter initial value + 1" count of the rising edge of the count source.

④Stop of count operation

By writing "1" to the timer X stop control bit by software, the count operation is stopped.

The count operation is continued until "1" is set to the timer X stop control bit.

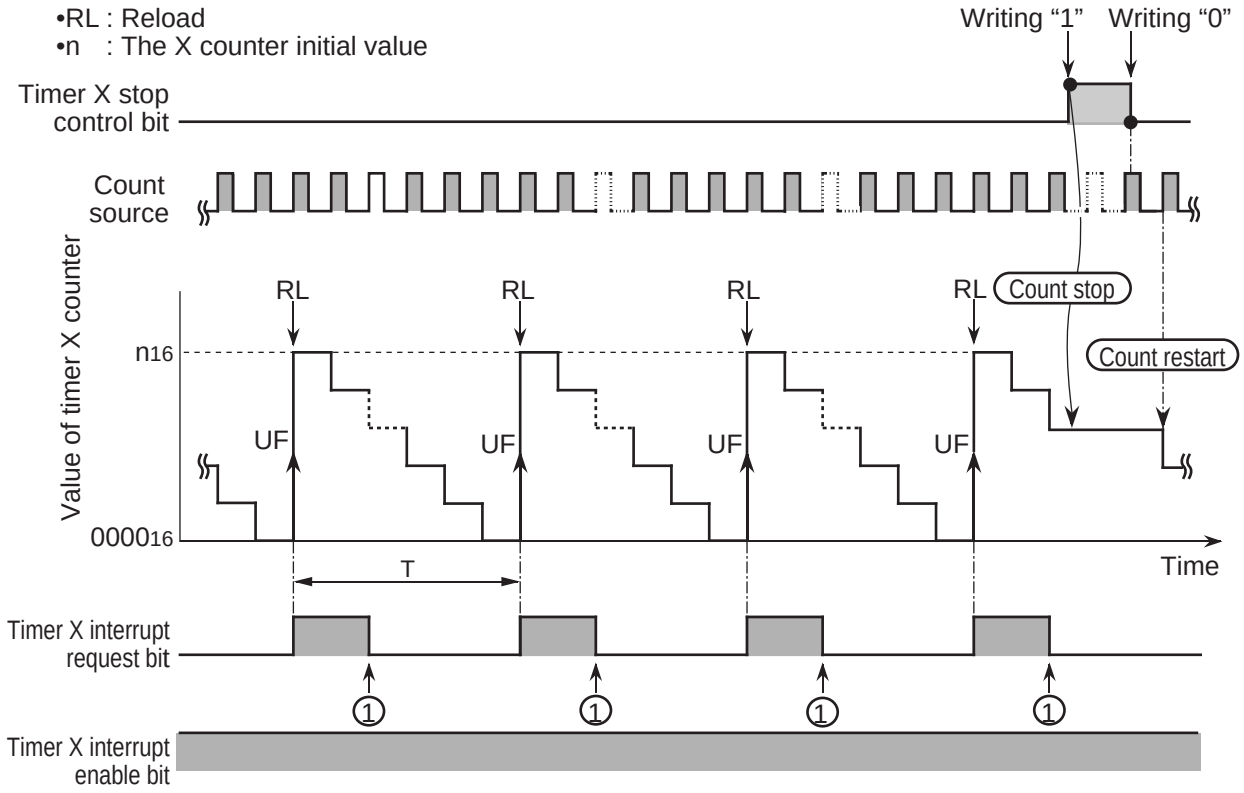
Figure 2.3.1 shows a timer mode operation example.

Count period

Count period T(s) = 1 ÷ count source frequency × (the X counter initial value + 1)

Timer mode operation example

•UF : Underflow
•RL : Reload
•n : The X counter initial value



① : •Clearing by writing "0" to the timer X interrupt request bit.
•Clearing by accepting the timer X interrupt request when the timer X interrupt enable bit is "1."

Fig. 2.3.1 Timer mode operation example

(2) Pulse output mode

The operation in the pulse output mode is the same as that in the timer mode, besides, which is added a pulse output operation. In this mode, a pulse whose polarity is reversed at every the X counter underflow is output from the P54/CNTR0 pin.

Operation in the pulse output mode is described below.

①Start of count operation

Immediately after reset release, the timer X stop control bit is in the "0" state. For this reason, the count operation is automatically started after reset release.

The value of the X counter is decremented by 1 each time a count source is input.

The count source is f(XIN)/16 clock (low-speed mode ; f(XCIN)/16 clock).

②Reload operation

The X counter underflows at the first count pulse after the value of the X counter reaches "0016." At this time, the value of the X latch is transferred (reloaded) to the X counter.

③Pulse output

A pulse whose polarity is reversed every the X counter underflow is output from the P54/CNTR0 pin. As a level at a start of pulse output, a "H" or "L" is selected by the CNTR0 active edge switch bit. At the time when the pulse output mode is selected by the timer X operating mode bits, a pulse output is started.

④Interrupt operation

■Counter underflow

An interrupt request occurs at the X counter underflow. At the same time, the timer X interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the timer X interrupt enable bit.

■Edge of pulse output

At the edge of the pulse output from the P54/CNTR0 pin, an interrupt request occurs. At the same time, the CNTR0 interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the CNTR0 interrupt enable bit.

As an active edge, the falling edge (↓) or rising edge (↑) is specified by the CNTR0 active edge switch bit.

⑤Stop of count operation

By writing "1" to the timer X stop control bit by software, the count operation is stopped.

The count operation is continued until "1" is set to the timer X stop control bit.

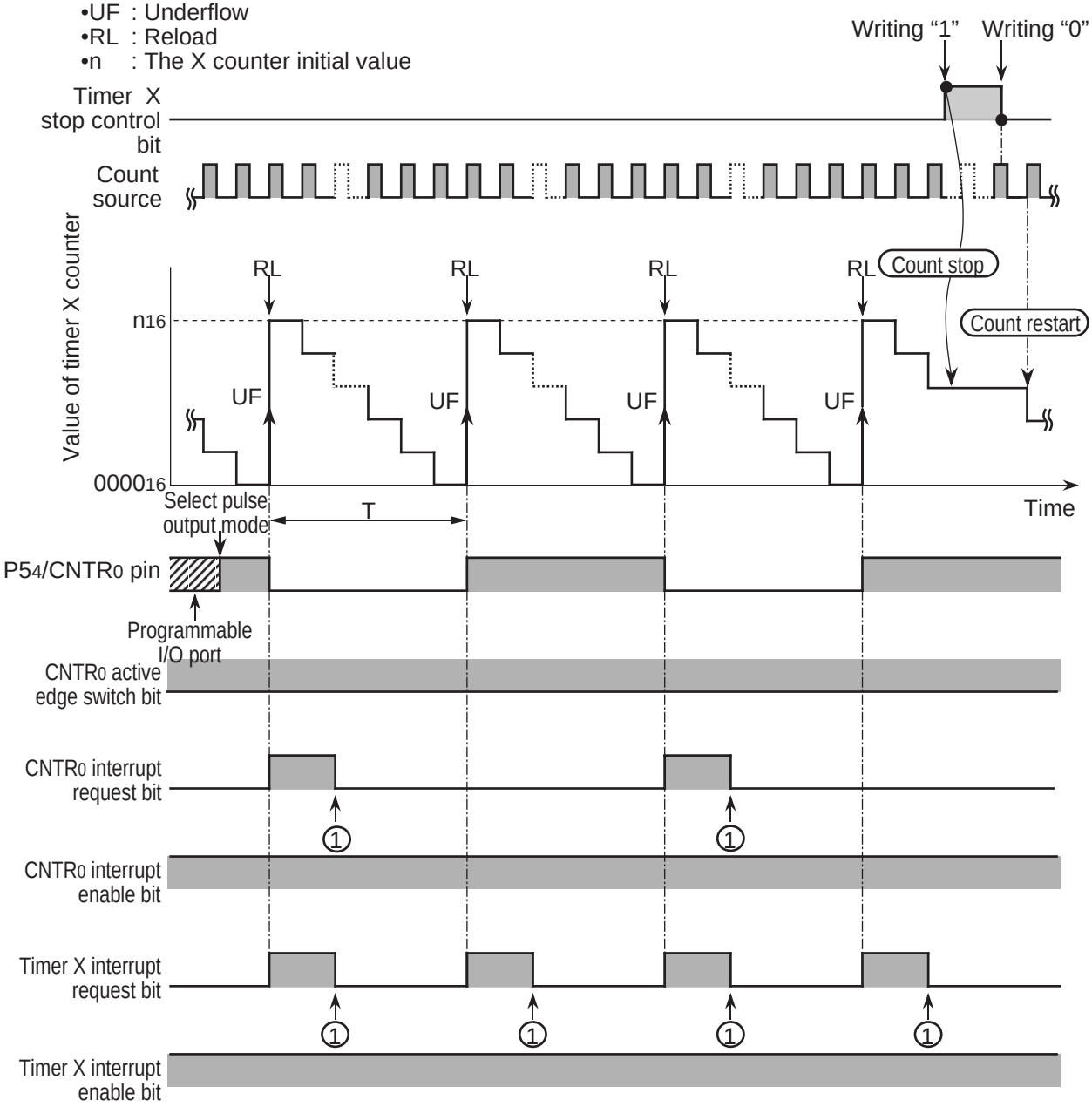
Figure 2.3.2 shows a pulse output mode operation example.

Count period

Count period T(s) = 1 ÷ count source frequency × (the X counter initial value + 1)

Pulse output mode operation example

•UF : Underflow
•RL : Reload
•n : The X counter initial value



① : •Clearing by writing "0" to the timer X interrupt request bit or the CNTR₀ interrupt request bit.
•Clearing by accepting the timer X interrupt request and the CNTR₀ interrupt request when the respective interrupt enable bits are "1."

✻ : When the CNTR₀ active edge switch bit is "1" ;
•The reverse-polarity pulse of above pulse is output.
•The CNTR₀ interrupt request occurs at the rising edge of the output pulse.

Fig. 2.3.2 Pulse output mode operation example

(3) Event counter mode

The operation in the event counter mode is the same as that in the timer mode except that the input signal to the CNTR0 pin is used as a count source.

Operation in the event counter mode is described below.

①Start of count operation

Immediately after reset release, the timer X stop control bit is in the "0" state. For this reason, the count operation is automatically started after reset release.

The value of the X counter is decremented by 1 each time a count source is input.

As an active edge, the falling edge (↓) or rising edge (↑) is specified by the CNTR0 active edge switch bit.

②Reload operation

The X counter underflows at the first count pulse after the value of the X counter reaches "00_{16}." At this time, the value of the X latch is transferred (reloaded) to the X counter.

③Interrupt operation

■Counter underflow

An interrupt request occurs at the X counter underflow. At the same time, the timer X interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the timer X interrupt enable bit.

■Edge of count source

At the edge of the count source input from the $P5_4$/CNTR0 pin, an interrupt request occurs. At the same time, the CNTR0 interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the CNTR0 interrupt enable bit.

As an active edge, the falling edge (↓) or rising edge (↑) is specified by the CNTR0 active edge switch bit.

④Stop of count operation

By writing "1" to the timer X stop control bit by software, the count operation is stopped.

The count operation is continued until "1" is set to the timer X stop control bit.

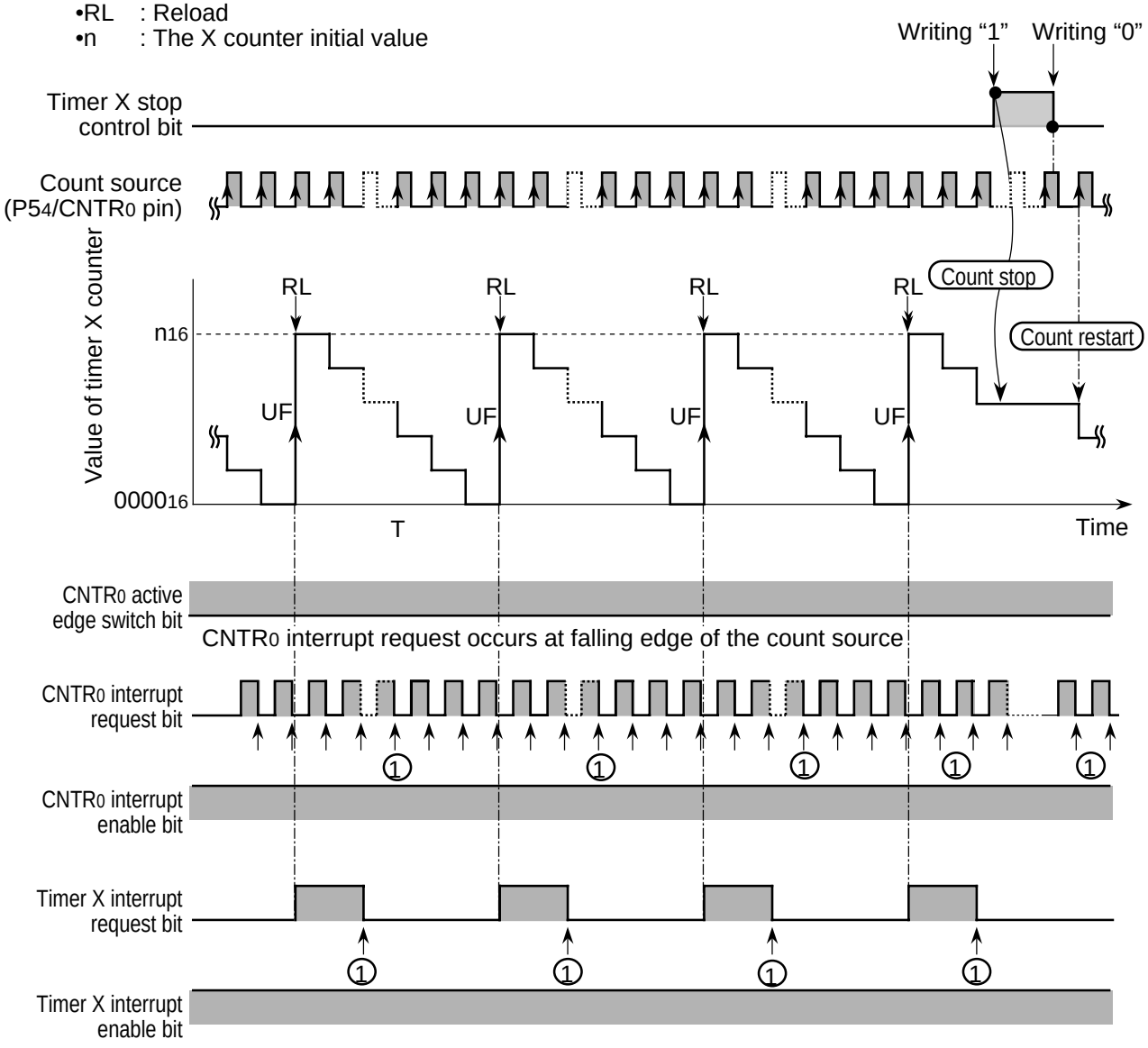
Figure 2.3.3 shows an event counter mode operation example.

Count period

Count period T(s) = 1 ÷ count source frequency × (the X counter initial value + 1)

Event counter mode operation example

•UF : Underflow
•RL : Reload
•n : The X counter initial value



① : •Clearing by writing "0" to the timer X interrupt request bit or the CNTR0 interrupt request bit.
•Clearing by accepting the timer X interrupt request and the CNTR0 interrupt request when the respective interrupt enable bits are "1."

✻ : When the CNTR0 active edge switch bit is "1" ;
•Falling edge of the count source is valid.
•The CNTR0 interrupt request occurs at the rising edge of the output pulse.

Fig. 2.3.3 Event counter mode operation example

(4) Pulse width measurement mode

In the pulse width measurement mode, the width ("H" or "L" level) of a pulse input from the P54/CNTR0 pin is measured.

Operation in the pulse width measurement mode is described below.

①Count operation

Immediately after reset, the timer X stop control bit is in the "0" state. In this state, a count operation is continued in the period in which the measurement level is input to the P54/CNTR0 pin.

The value of the X counter is decremented by 1 each time a count source is input.

The count source is f(XIN)/16 clock (low-speed mode ; f(XCIN)/16 clock).

②Reload operation

The X counter underflows at the first count pulse after the value of the X counter reaches "00_{16}." At this time, the value of the X latch is transferred (reloaded) to the X counter.

③Pulse width measurement

As a pulse measurement period, a "H" or "L" is selected by the CNTR0 active edge switch bit.

The difference between the initial value of the X counter and the X counter value at counter stop is a measured pulse width.

A reload operation by reading the count value is not performed automatically. Accordingly, to continue the measurement, set the initial value anew by software.

④Interrupt operation

■Edge of pulse measured

At the edge of the pulse input from the P54/CNTR0 pin, an interrupt request occurs. At the same time, the CNTR0 interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the CNTR0 interrupt enable bit.

The CNTR0 active edge switch bit specifies an active edge. When "H" level width is measured, the falling edge (↓) is active, when "L" level width is measured, the rising edge (↑) is active.

■Counter underflow

An interrupt request occurs at the X counter underflow. At the same time, the timer X interrupt request bit is set to "1." The occurrence of an interrupt is controlled by using the timer X interrupt enable bit.

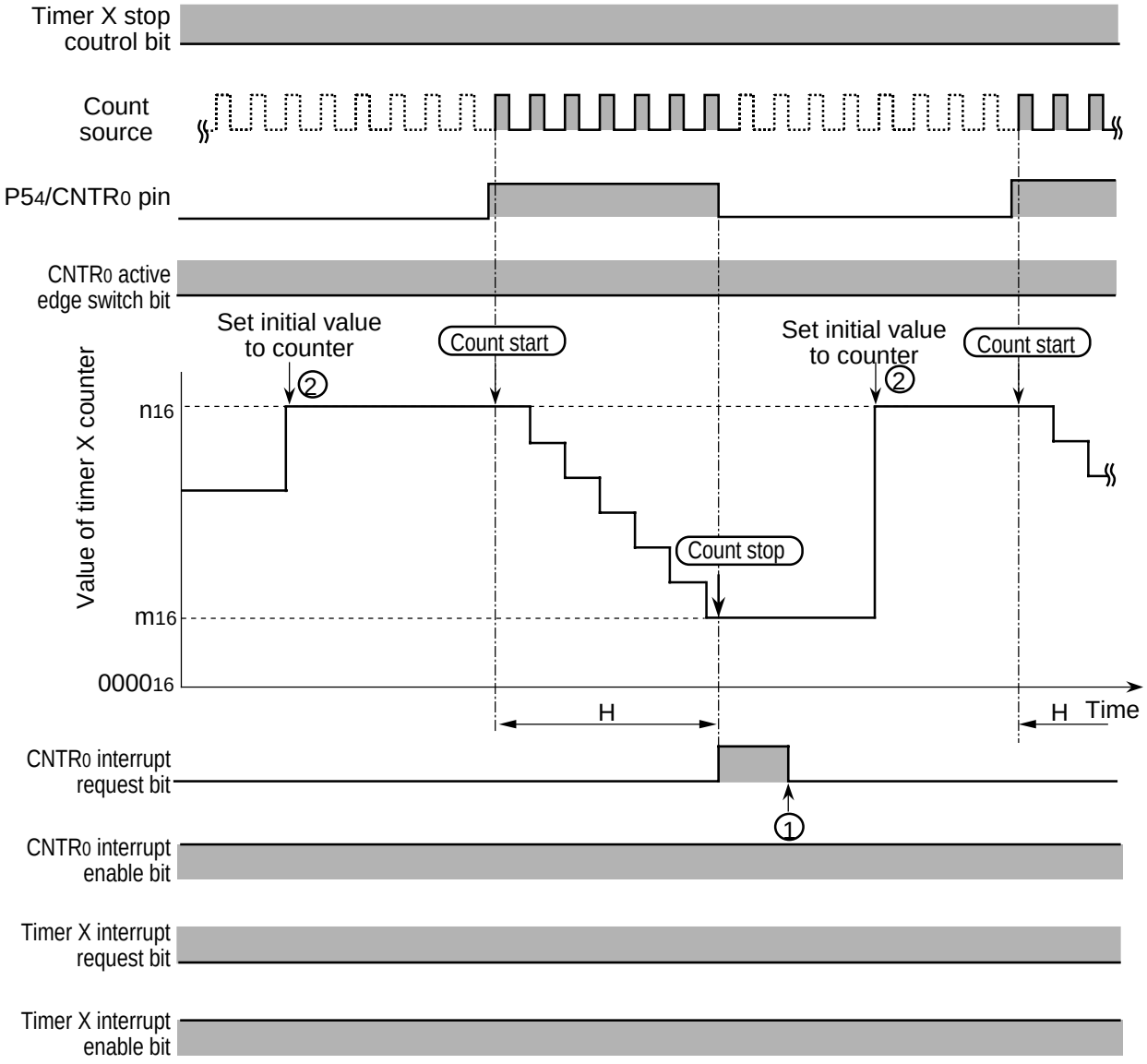
Figure 2.3.4 shows a pulse width measurement mode operation example.

Pulse width

Pulse width H(s) = 1 ÷ count source frequency × (the X counter initial value – the X counter value at count stop)

Pulse width measurment mode operation example

•n : The X counter initial value
•m: The X counter value at count stop



① : •Clearing by writing "0" to the CNTR0 interrupt request bit.
•Clearing by accepting the CNTR0 interrupt request when the CNTR0 interrupt enable bit is "1."

② : Set initial value to the timer X when timer X write control bit is "0."

✻ : When the CNTR0 active edge switch bit is "1" ;
•"L" level width of the input pulse is measured.
•The CNTR0 interrupt request occurs at the rising edge of the input pulse.

Fig. 2.3.4 Pulse width measurement mode operation example

(5) Real time port control

The real time port control is the function which outputs preset data from the real time ports in synchronization with an underflow of the X counter. Table 2.3.1 shows real time ports and data storage bits. This real time port control function is available in every mode.

A data output from the real time port is started at setting the real time port control bit to "1." When the values of the data storage bits are rewritten, the rewritten values are output at the first underflow of the X counter after rewritting. Figure 2.3.5 shows a timer mode operation example with the real time port function.

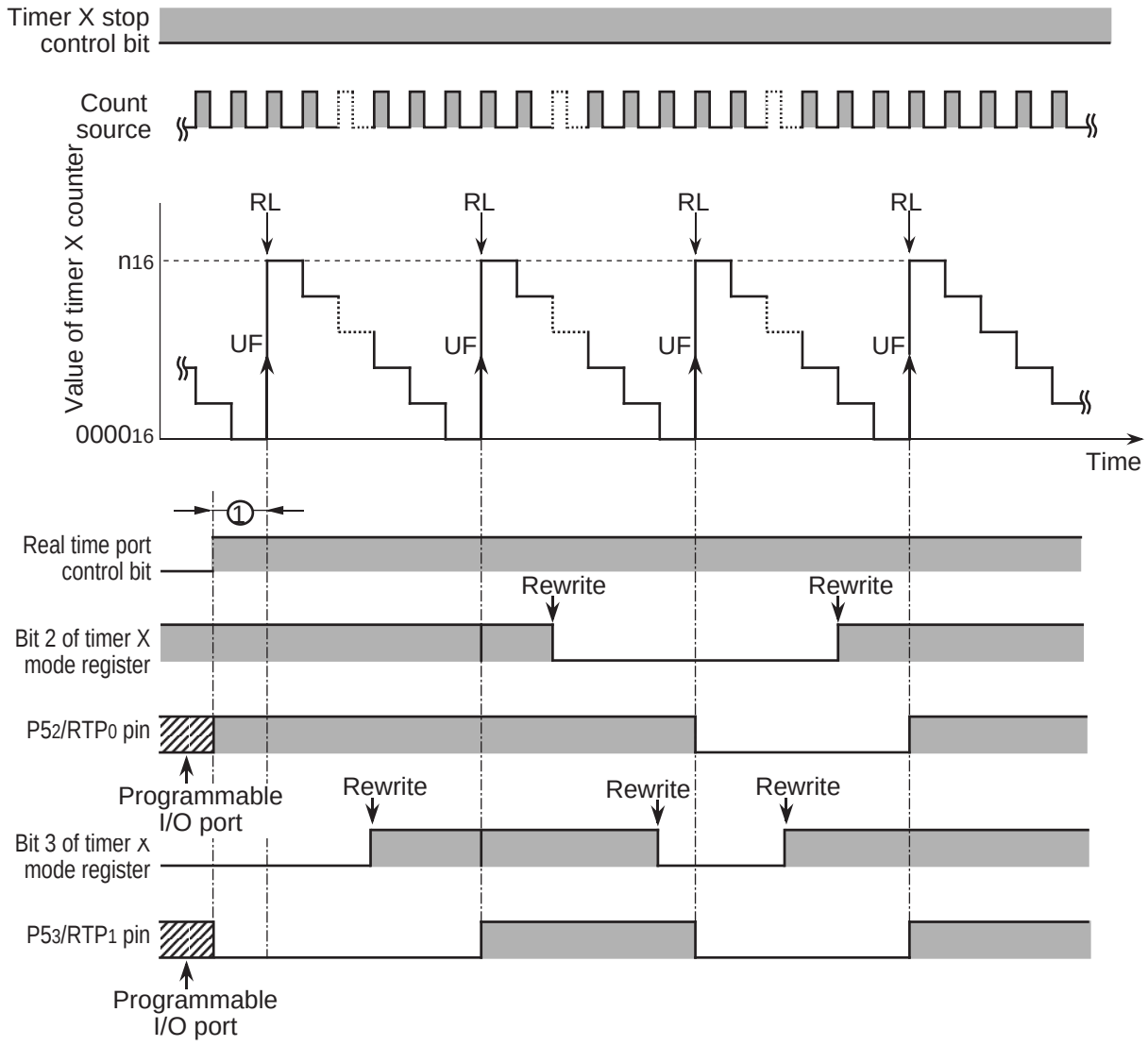
The real time port is also used as port $P5_2$ and $P5_3$. When using the real time port, set the corresponding bit of the port P5 direction register (address $000B_{16}$) to "1" for using as an output port.

Table 2.3.1 Real time ports and data storage bits

Real time port	Data storage bit
RTP_0 ($P5_2$)	Bit 2 of timer X mode register
RTP_1 ($P5_3$)	Bit 3 of timer X mode register

Timer mode operation example with real time port function

•UF : Underflow
•RL : Reload
•n : The X counter initial value



① : When the both settings below are performed immediately after reset, "0" is output from pins RTP0 and RTP1.
•Set ports P52 and P53 for the output mode.
•Set the real time port control bit to "1."

Fig. 2.3.5 Timer mode operation example with real time port function

2.3.2 Explanation of timer Y operations

Timer Y has 4 modes of operation.
Operation in each mode is described below.

(1) Timer Mode

Operation in the timer mode is described below.

①Start of count operation

Immediately after reset release, the timer Y stop control bit is in the "0" state. For this reason, the count operation is automatically started after reset release.

The value of the timer Y counter (referred as "the Y counter") is decremented by 1 each time a count source is input.

The count source is f(XIN)/16 clock (low-speed mode ; f(XCIN)/16 clock).

②Reload operation

The Y counter underflows at the first count pulse after the value of the Y counter reaches "00_{16}." At this time, the value of the timer Y latch (referred as "the Y latch") is transferred (reloaded) to the Y counter.

③Interrupt operation

An interrupt request occurs at the Y counter underflow. At the same time, the timer Y interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the timer Y interrupt enable bit.

An interrupt request occurs each time the counter underflows. In other words, an interrupt request occurs every "the Y counter initial value + 1" count of the rising edge of the count source.

④Stop of count operation

By writing "1" to the timer Y stop control bit by software, the count operation is stopped.

The count operation is continued until "1" is set to the timer Y stop control bit.

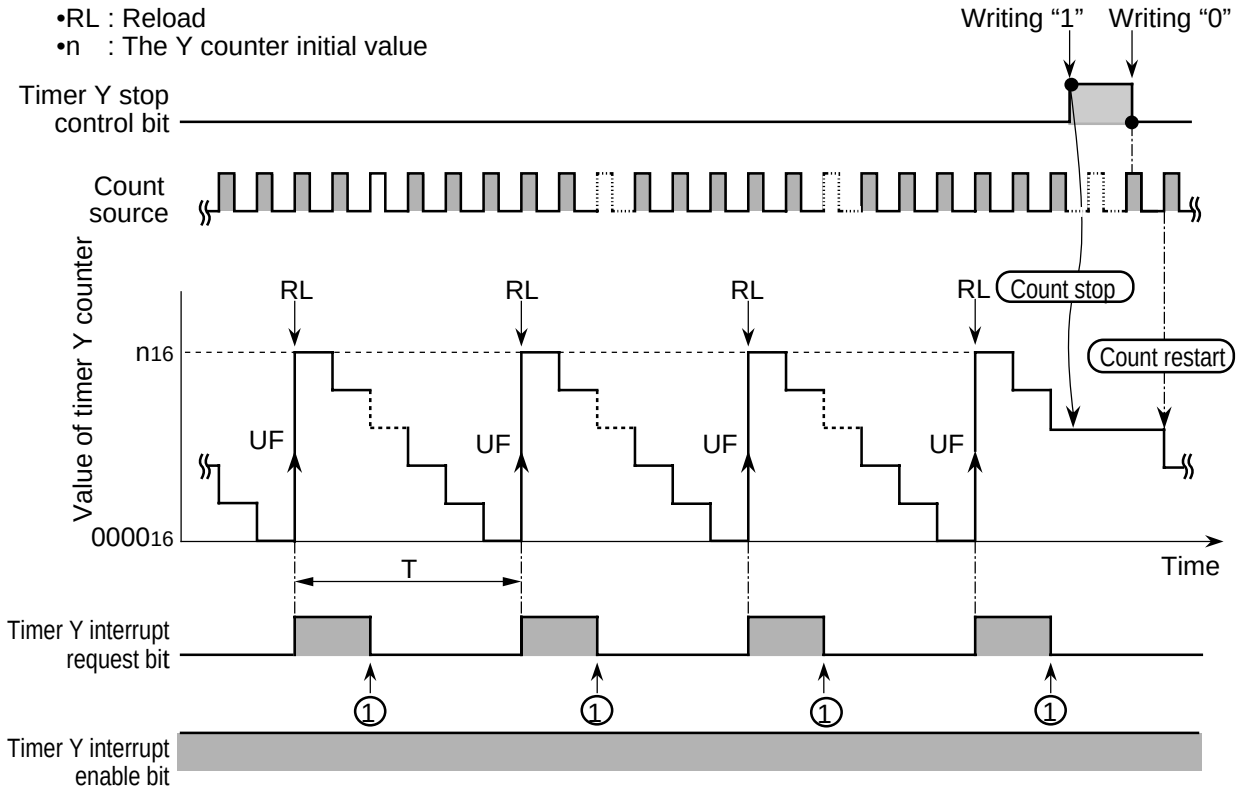
Figure 2.3.6 shows a timer mode operation example.

Count period

Count period T(s) = 1 ÷ count source frequency × (the Y counter initial value + 1)

Timer mode operation example

•UF : Underflow
•RL : Reload
•n : The Y counter initial value



①: •Clearing by writing "0" to the timer Y interrupt request bit.
•Clearing by accepting the timer Y interrupt request when the timer Y interrupt enable bit is "1."

Fig. 2.3.6 Timer mode operation example

(2) Period measurement mode

In the period measurement mode, the period of a pulse input from the P55/CNTR1 pin is measured. Operation in the period measurement mode is described below.

①Start of count operation

Immediately after reset release, the timer Y stop control bit is in the "0" state. For this reason, the count operation is automatically started after reset release.

The value of the Y counter is decremented by 1 each time a count source is input.

The count source is f(XIN)/16 clock (low-speed mode ; f(XCIN)/16 clock).

②Reload operation

At the edge of the pulse input from the P55/CNTR1 pin, the value of the Y latch is transferred (reloaded) to the Y counter. The count value immediately before reload is held until it is read out once after reload.

As an active edge, the falling edge (⤵) or rising edge (⤴) is specified by the CNTR1 active edge switch bit.

The value of the Y latch is also reloaded at the Y counter underflow.

③Period measurement

As a period measurement duration, the following is selected by the CNTR1 active edge switch bit (bit 6) : Duration from the falling edge to the falling edge (bit 6 = "0")
Duration from the rising edge to the rising edge (bit 6 = "1")

The difference between the count value at an active edge input and that immediately before reload is a measured period.

④Interrupt operation

■Edge of input pulse

At the edge of the pulse input from the P55/CNTR1 pin, an interrupt request occurs. At the same time, the CNTR1 interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the CNTR1 interrupt enable bit.

As an active edge, the falling edge (⤵) or rising edge (⤴) is specified by the CNTR1 active edge switch bit.

■Counter underflow

An interrupt request occurs at the Y counter underflow. At the same time, the timer Y interrupt request bit is set to "1."

The occurrence of an interrupt is controlled by the timer Y interrupt enable bit.

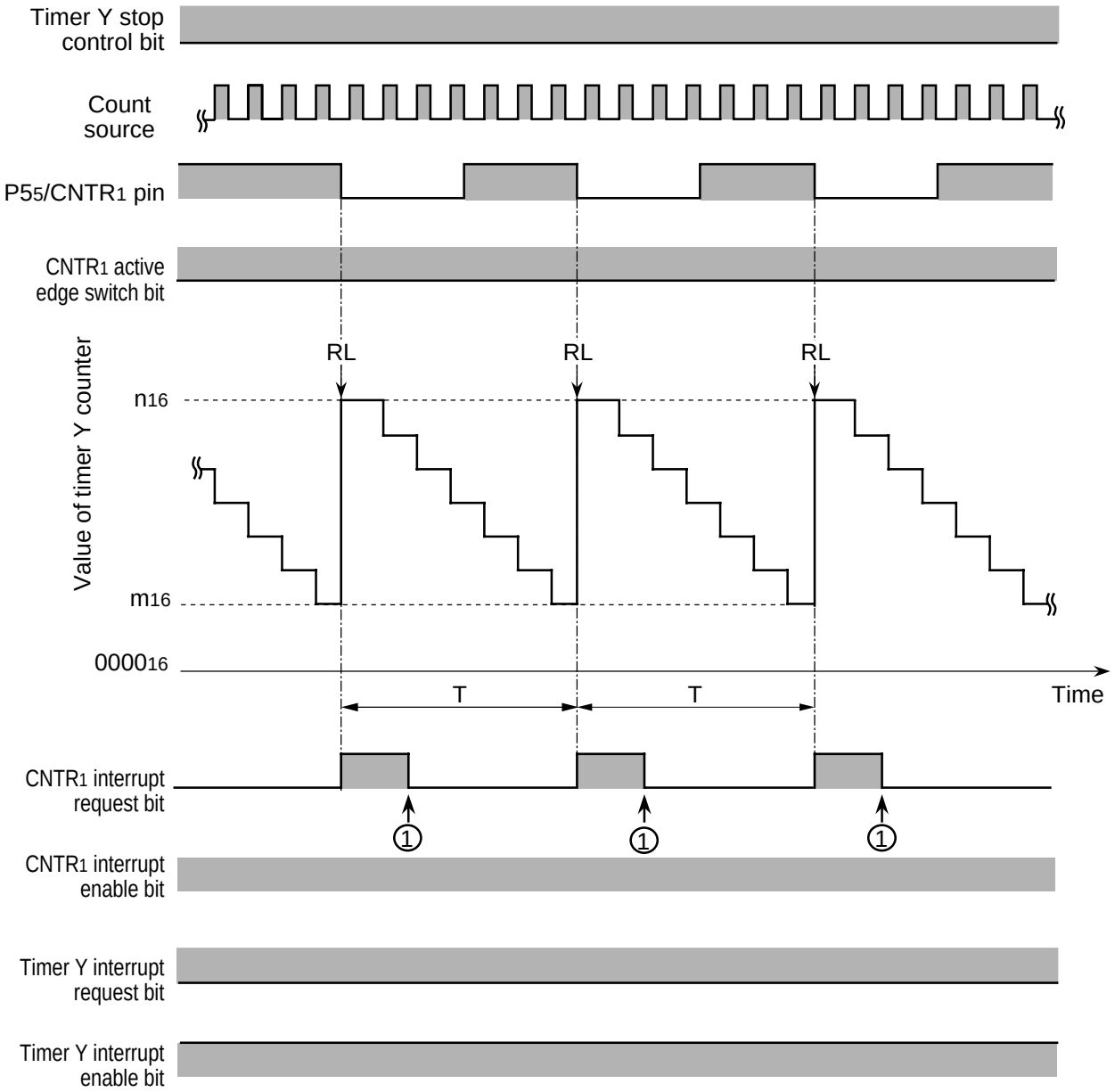
Figure 2.3.7 shows a period measurement mode operation example.

Pulse period

Pulse period T(s) = 1 ÷ count source frequency × (the Y counter initial value – the Y counter value immediately before reload)

Period measurement mode operation example

- •RL : Reload
- •n : The Y counter initial value
- •m : The Y counter value immediately before reload



①: •Clearing by writing "0" to the CNTR1 interrupt request bit.
•Clearing by accepting the CNTR1 interrupt request when the CNTR1 interrupt enable bit is "1."

✻ : When the CNTR1 active edge switch bit is "1" ;
•From the rising edge to the rising edge of the input pulse is measured.
•The CNTR1 interrupt request occurs at the rising edge of the input pulse.

Fig. 2.3.7 Period measurement mode operation example

(3) Event counter mode

The operation in the event counter mode is the same as that in the timer mode except that the input signal to the P55/CNTR1 pin is used as a count source.
Operation in the event counter mode is described below.

①Start of count operation

Immediately after reset release, the timer Y stop control bit is in the "0" state. For this reason, the count operation is automatically started after reset release.
The value of the Y counter is decremented by 1 each time a count source is input.
As an active edge, the falling edge (↓) or rising edge (↑) is specified by the CNTR1 active edge switch bit.

②Reload operation

The Y counter underflows at the first count pulse after the value of the Y counter reaches "0016." At this time, the value of the Y latch is transferred (reloaded) to the Y counter.

③Interrupt operation

■Counter underflow

An interrupt request occurs at the Y counter underflow. At the same time, the timer Y interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the timer Y interrupt enable bit.

■Edge of count source

At the edge of the count source input from the P55/CNTR1 pin, an interrupt request occurs. At the same time, the CNTR1 interrupt request bit is set to "1." The occurrence of an interrupt is controlled by the CNTR1 interrupt enable bit.
As an active edge, the falling edge (↓) or rising edge (↑) is specified by the CNTR1 active edge switch bit.

④Stop of count operation

By writing "1" in the timer Y stop control bit by software, the count operation is stopped.
The count operation is continued until "1" is set in the timer Y stop control bit.

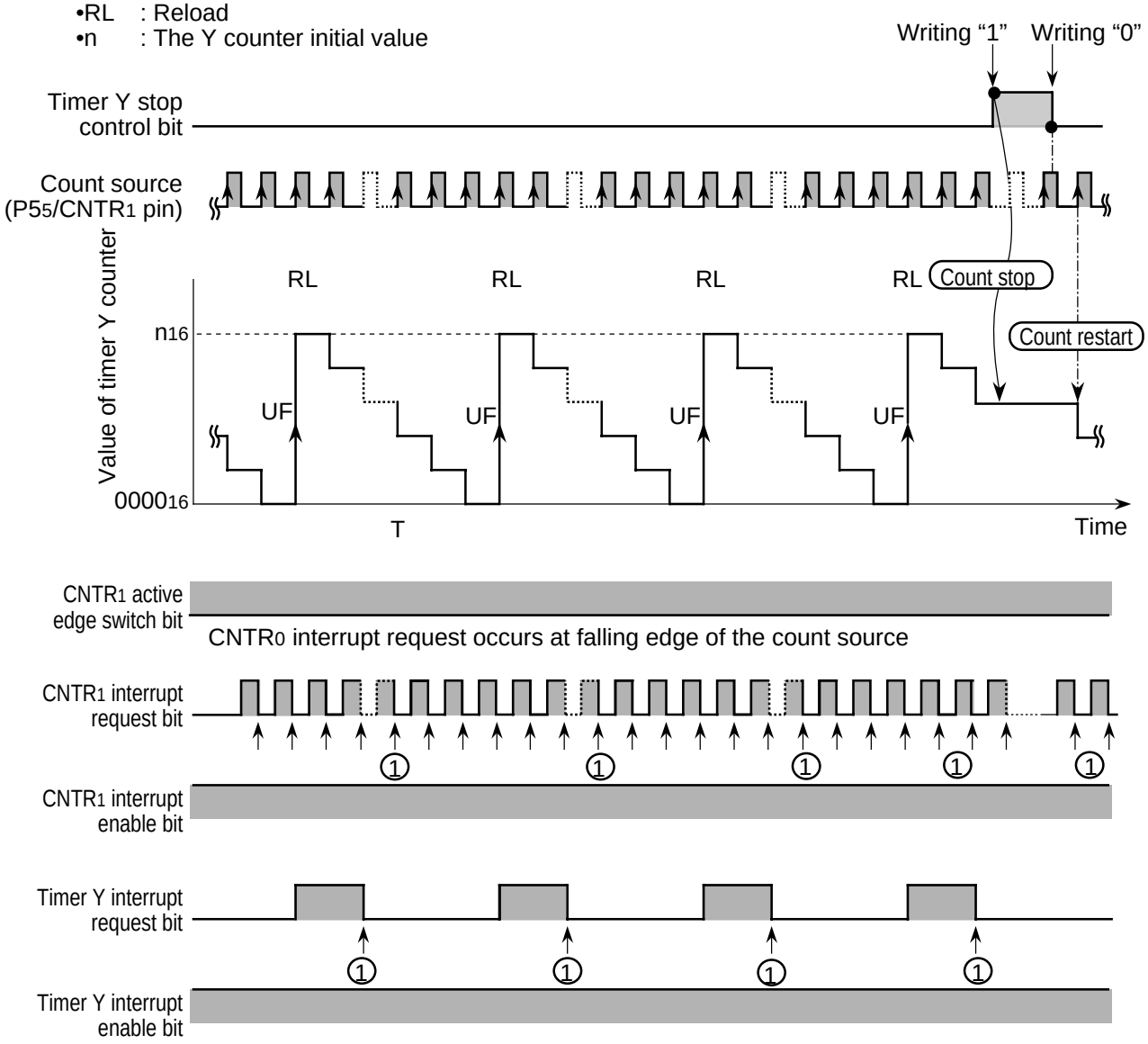
Figure 2.3.8 shows an event counter mode operation example.

Count period

Count period T(s) = 1 ÷ count source frequency × (the Y counter initial value + 1)

Event counter mode operation example

•UF : Underflow
•RL : Reload
•n : The Y counter initial value



① : •Clearing by writing "0" to the timer Y interrupt request bit or the CNTR 1 interrupt request bit.
•Clearing by accepting the timer Y interrupt request and the CNTR 1 interrupt request when the respective interrupt enable bits are "1."

✻ : When the CNTR1 active edge switch bit is "1" ;
•Falling edge of the count source is valid.
•The CNTR1 interrupt request occurs at the rising edge of the output pulse.

Fig. 2.3.8 Event counter mode operation example

(4) Pulse width HL continuously measurement mode

In the pulse width HL continuously measurement mode, the width ("H" and "L" level) of pulses input from the P55/CNTR1 pin are continuously measured.

With the exception that reload and an interrupt request occur at both edges of pulses input from CNTR1, the operation in the pulse width HL continuously measurement mode is the same as that in the period measurement mode.

The pulse width HL continuously measurement mode of operation is described below.

①Start of count operation

Immediately after reset release, the timer Y stop control bit is in the "0" state. For this reason, the count operation is automatically started after reset release.

The value of the Y counter is decremented by 1 each time a count source is input.

The count source is f(XIN)/16 (low-speed mode ; f(XCIN)/16).

②Reload operation

At both edges of the pulse input from the P55/CNTR1 pin, the value of the timer Y is transferred (reloaded) to the Y counter. The count value immediately before reload is held until it is read out once after reload.

The value of the Y latch is also reloaded at the Y counter underflow.

③Pulse width measurement

The difference between the count value at an active edge input and that immediately before reload is a measured pulse width.

④Interrupt operation

■Edge of input pulse

At both edges of pulses input from the P55/CNTR1 pin, an interrupt request occurs. At the same time, the CNTR1 interrupt request bit is set. The occurrence of an interrupt is controlled by the CNTR1 interrupt enable bit.

■Counter underflow

An interrupt request occurs at the Y counter underflow. At the same time, the timer Y interrupt request bit is set to "1."

The occurrence of an interrupt is controlled by the timer Y interrupt enable bit.

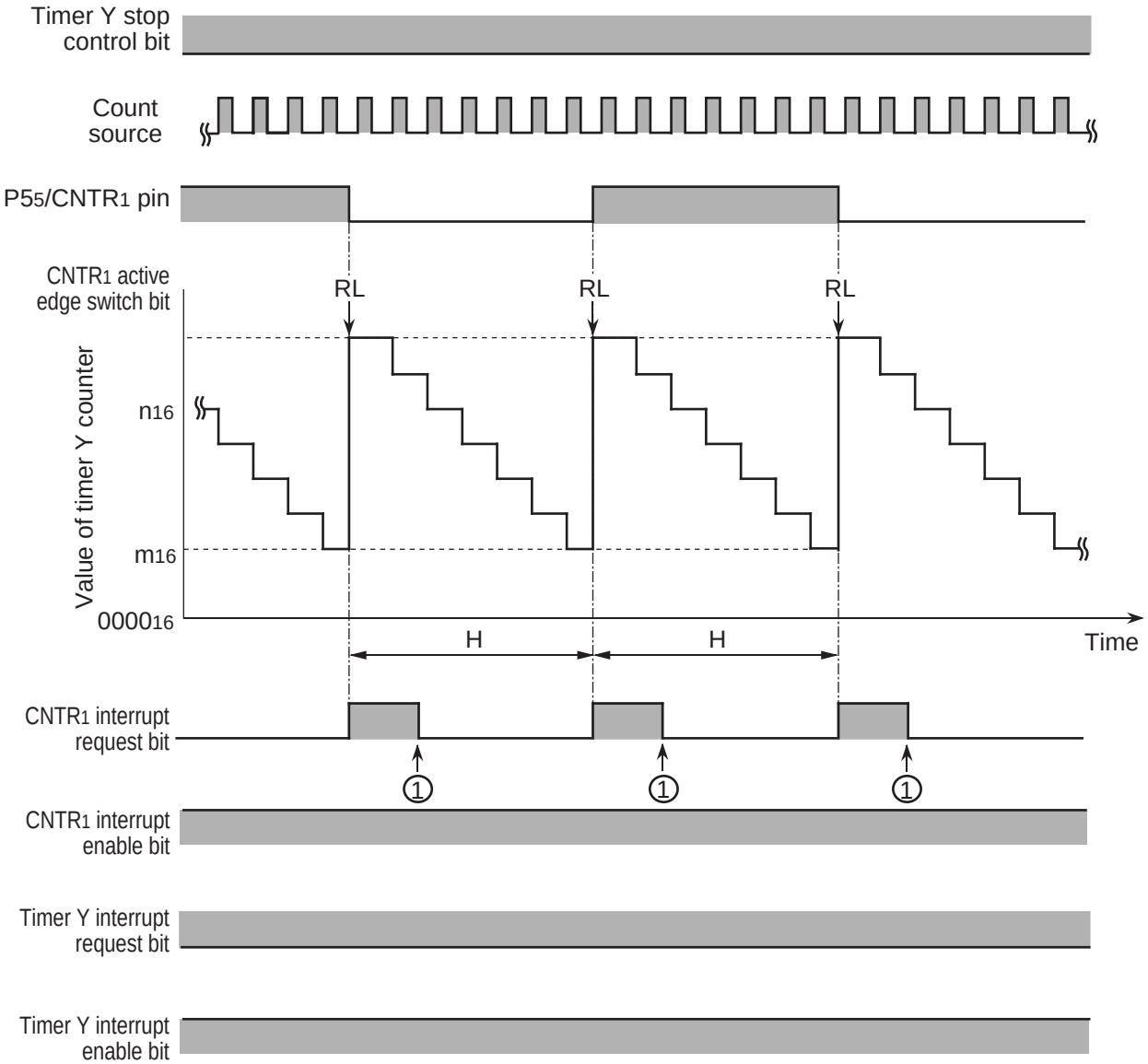
Figure 2.3.9 shows a pulse width HL continuously measurement mode operation example.

Pulse width

Pulse width H(s) = 1 ÷ count source frequency × (the Y counter initial value – the Y counter value immediately before reload)

Operation example in pulse width HL continuously measurement mode

- RL : Reload
- n : The Y counter initial value
- m : The Y counter value immediately before reload



① : •Clearing by writing "0" to the CNTR1 interrupt request bit.
•Clearing by accepting the CNTR1 interrupt request when the CNTR1 interrupt enable bit is "1."

Fig. 2.3.9 Pulse width HL continuously measurement mode operation example

2.3 Timer X and timer Y

2.3.3 Related registers

Figure 2.3.10 shows the memory allocation of the timer X- and timer Y-related registers. Each of these registers is described below.

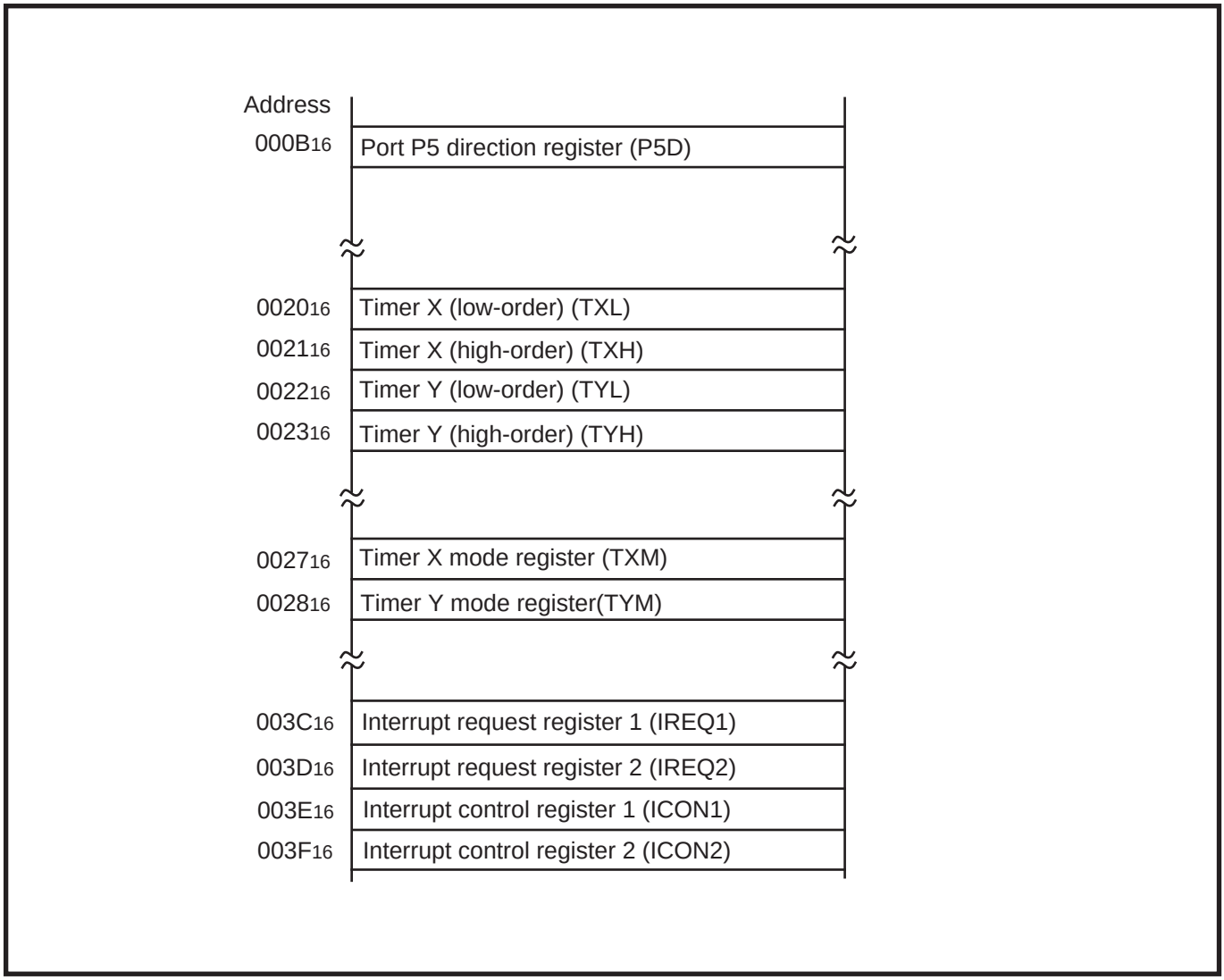


Fig. 2.3.10 Memory allocation of timer X- and timer Y-related registers

(1) Port P5 direction register (P5D)

The port P5 direction register (address 000B16) selects the I/O direction of port P5. Figure 2.3.11 shows the structure of the port P5 direction register.

The $CNTR_0$ pin is also used as $P5_4$, while the $CNTR_1$ pin is also used as $P5_5$.

■Timer X

In the pulse output mode, set bit 4 to "1" for the output mode.

In the event counter mode or the pulse width measurement mode, set bit 4 to "0" for the input mode.

The real time port RTP_0 pin is also used as $P5_2$, while the RTP_1 pin is also used as $P5_3$.

To use as the RTP_0 pin, set bit 2 to "1" for the output mode. To use as the RTP_1 pin, set bit 3 to "1" for the output mode.

■Timer Y

In the period measurement mode or the event counter mode or the pulse width HL continuously measurement mode, set bit 5 to "0" to set it for the input mode.

Port P5 direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port P5 direction register (P5D) [Address B16]

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

Note : Port P5 direction register cannot be read out.

$P5_7$	$P5_6$	$P5_5$	$P5_4$	$P5_3$	$P5_2$	$P5_1$	$P5_0$
ADT	T_{OUT}	$CNTR_1$	$CNTR_0$	RTP_1	RTP_0	INT_3	INT_2

Fig. 2.3.11 Structure of port P5 direction register

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

The timer X latch (referred as "the X latch") and the timer X counter (referred as "the X counter") consist of 16 bits in a combination of high-order and low-order.

The X latch and the X counter are allocated at the same address. To access the X latch and the X counter, access both the timer X (low-order) and the timer X (high-order).

■Read

When the timer X (high-order) and the timer X (low-order) are read out, the value of the X counter (count value) are read out. Read both registers in the order of the timer X (high-order) and the timer X (low-order).

Do not write any value to the timer X (high-order) and the timer X (low-order) before the timer X (low-order) has been read out. In this case, timer X will not operate normally.

■Write

When a value is written to the timer X (low-order) and the timer X (high-order), the value is set in the X latch and the X counter at the same time. Writing to the X latch only can be selected by the timer X write control bit (refer to **"2.3.3 Related registers, (4) Timer X mode register"**). Write the values to both registers in the order of the timer X (low-order) and the timer X (high-order).

Do not read timer X (low-order) and the timer X (high-order) before the timer X (high-order) has been written. In this case, timer X will not operate normally.

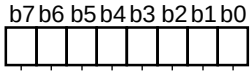
●Timer X latch

The X latch is a register which holds the value to be transferred (reloaded) automatically to the X counter as the initial value of the X counter at the X counter underflow. Figure 2.3.12 shows the structure of the timer X latch.

The contents of the X latch cannot be read out.

●Timer X latch

Timer X (high-order, low-order)



Timer X (high-order, low-order) (TXH, TXL) [Address 21 16, 2016]

B	Functions	At reset	R	W
0 to 7	•Set "0000 16 to FFFF 16" as timer X count value. •Write high-order byte of setting value to TXH, and low-order byte to TXL, respectively. •The values of TXH and TXL are set to the respective X latches and transferred automatically to the respective X counters at the X counter underflow.	1	×	○

Note : Write both registers in the order of TXL and TXH.

Fig. 2.3.12 Structure of timer X latch

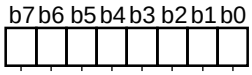
■Timer X counter

The X counter counts the count source. Figure 2.3.13 shows the structure of the timer X counter. The contents of the X counter are decremented by 1 each time a count source is input.
The division ratio of the counter is represented by the following expression.

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

●Timer X counter

Timer X (high-order, low-order)



Timer X (high-order, low-order) (TXH, TXL) [Address 21 16, 2016]

B	Functions	At reset	R	W
0 to 7	•Set "0000$_{16}$ to FFFF$_{16}$" as timer X count value. •The value of the X counter is decremented by 1 each time a count source is input. •When the timer X write control bit is "0," the values of TXH and TXL are set to the respective X latches at the same time. •The values of each X counter are read out by reading the respective timer Xs.	1	○	○

Notes 1 : Write both registers in the order of TXL and TXH.
2 : Read both registers in the order of TXH and TXL.

Fig. 2.3.13 Structure of timer X counter

(3) Timer Y latch and timer Y counter (corresponding to timer Y (low-order) and timer Y (high-order))

The timer Y latch (referred as “the Y latch”) and the timer Y counter (referred as “the Y counter”) consist of 16 bits in a combination of high-order and low-order.

The Y latch and Y counter are allocated at the same address. To access the Y latch and the Y counter, access both the timer Y (low-order) and the timer Y (high-order).

■Read

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

■Write

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

Do not read the timer Y (low-order and high-order) before the timer Y (high-order) has been written. In this case, timer Y will not operate normally.

●Timer Y latch

The Y latch is a register which holds the value to be transferred (reloaded) automatically to the Y latch as the initial value of the Y counter at the Y counter underflow. Figure 2.3.14 shows the structure of the timer Y latch.

Reload is performed at the following :

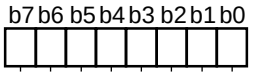
- At the Y counter underflow
- At the edge of the input pulse from the $P5_5$/$CNTR_1$ pin

(period measurement mode/pulse width HL coutinuously measurement mode)

The contents of the Y latch cannot be read out.

●Timer Y latch

Timer Y (high-order, low-order)



Timer Y (high-order, low-order) (TYH, TYL) [Address 23_{16}, 22_{16}]

B	Functions	At reset	R	W
0 to 7	•Set “0000_{16} to $FFFF_{16}$” as timer Y count value. •Write high-order byte of setting value to TYH, and low-order byte to TYL, respectively. •The values of TYH and TYL are set to the respective Y latches and transferred automatically to the respective Y counters at the Y counter underflow.	1	×	○

Note : Write both registers in the order of TYL and TYH.

Fig. 2.3.14 Structure of timer Y latch

■Timer Y counter

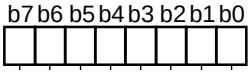
The Y counter counts the count source. Figure 2.3.15 shows the structure of the timer Y counter. The contents of the Y counter are decremented by 1 each time a count source is input.
The division ratio of the counter is represented by the following expression.

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

In the period measurement mode or the pulse width HL coutinuously measurement mode, the value immediately before reload is held until it is read out once after reload. The count operation is coutinued.

●Timer Y counter

Timer Y (high-order, low-order)



Timer Y (high-order, low-order) (TYH, TYL) [Address 23 16, 2216]

B	Functions	At reset	R	W
0 to 7	•Set "0000$_{16}$ to FFFF$_{16}$" as timer Y count value. •The value of the Y counter is decremented by 1 each time a count source is input. •The values of each Y counter are read out by reading the respective timer Ys. •The Y counter value immediately before reload is held until it is read out once after reload. (period measurement mode/pulse width HL continuously measurement mode)	1	○	○

Notes 1 : Write both registers in the order of TYL and TYH.

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

Fig. 2.3.15 Structure of timer Y counter

(4) Timer X mode register (TXM)

The timer X mode register (address 0027$_{16}$) consists of bits which select operation or control counting. Figure 2.3.16 shows a structure of the timer X mode register. Each bit is described below.

Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer X mode register (TXM) [Address 27 $_{16}$]

B	Name	Functions	At reset	R	W
0	Timer X write control bit	0 : Write value in latch and counter 1 : Write value in latch only	0	○	○
1	Real time port control bit	0 : Real time port function invalid 1 : Real time port function valid	0	○	○
2	Data storage bit for real time port (RTP$_0$)	0 : "L" level output 1 : "H" level output	0	○	○
3	Data storage bit for real time port (RTP$_1$)	0 : "L" level output 1 : "H" level output	0	○	○
4	Timer X operating mode bits	b5b4 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
5			0	○	○
6	CNTR$_0$ active edge switch bit	•CNTR$_0$ interrupt 0 : Falling edge active 1 : Rising edge active •Pulse output mode 0 : Start at initial level "H" output 1 : Start at initial level "L" output •Event counter mode 0 : Rising edge active 1 : Falling edge active •Pulse width measurement mode 0 : Measure "H" level width 1 : Measure "L" level width	0	○	○
7	Timer X stop control bit	0 : Count start 1 : Count stop	0	○	○

Fig. 2.3.16 Structure of timer X mode register

■Timer X write control bit (bit 0)

The timer X write control bit controls writing to the timer X (low-order and high-order).

When bit 0 is "0," the value written in the timer X (low-order and high-order) are set into both the X latch and the X counter at the same time.

When bit 0 is "1," the value written in the timer X (low-order and high-order) is set into the X latch only.

When a value is written into the X latch only, this rewritten value is transferred to the X counter at the first X counter underflow after rewriting.

■Real time port control bit (bit 1)

The real time port control bit selects a function to output data from the real time port. When bit 1 is "0," this function is invalid. When the bit is "1," this function is valid.

For an explanation of operations, refer to **"2.3.1 Explanation of timer X operations, (5) Real time port control."**

■Data storage bits for real time port (bit 2 and bit 3)

The data storage bits for real time port set the data to be output from the real time port.

■Timer X operating mode bits (bit 4 and bit 5)

The timer X operating mode bits select a operating mode of the timer X.

Table 2.3.2 shows the relation between the timer X operating mode bits and the operating modes. For an explanation of each mode operation, refer to the section pertaining to the explanation of each operation.

Table 2.3.2 Relation between timer X operating mode bits and operating modes

b5	b4	Operation mode
0	0	Timer mode
0	1	Pulse output mode
1	0	Event counter mode
1	1	Pulse width measurement mode

■CNTR0 active edge switch bit (bit 6)

The CNTR0 active edge switch bit has a function which selects an active edge of the CNTR0 interrupt, and functions for each mode.

●CNTR0 interrupt

When bit 6 is "0," the falling edge (↓) is active.
When bit 6 is "1," the rising edge (↑) is active.

●Pulse output mode

In the pulse output mode, the initial level at the start of pulse output is selected.
When bit 6 is "0," the initial level is "H."
When bit 6 is "1," the initial level is "L."

●Event counter mode

An active edge of the count source is selected.
When bit 6 is "0," the rising edge (↑) is active.
When bit 6 is "1," the falling edge (↓) is active.

●Pulse width measurement mode

A duration of pulse width measured is selected.
When bit 6 is "0," the "H" level width is measured.
When bit 6 is "1," the "L" level width is measured.

■Timer X stop control bit (bit 7)

The timer X stop control bit controls the count operation of the timer X.

By writing "0" to bit 7, a count source is input to the X counter, so that a count operation is started. As bit 7 is in the "0" state immediately after reset release, the count operation is automatically started after reset release.

By writing "1" to bit 7, the input of count source to the X counter is stopped, so that the count operation stops.

In the pulse width measurement mode, however, a count operation is performed only in the period in which the measurement level is input to the P54/CNTR0 pin when bit 7 is in the "0" state.

At read, this bit functions as a status bit to indicate the operating state (counting or stop) of the X counter. When bit 7 is "0," the counter is in the operating state. When bit 7 is "1," the counter is in the stop state.

(5) Timer Y Mode Register (TYM)

The timer Y mode register (address 0028$_{16}$) consists of bits which select operation or control counting. Figure 2.3.17 shows a structure of the timer Y mode register. Each bit is described below.

Timer Y mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer Y mode register (TYM) [Address 28 $_{16}$]

B	Name	Functions	At reset	R	W
0 to 3	Nothing is allocated. These bits are fixed to "0" at reading.		0	0	×
4	Timer Y operating mode bits	b5b4 0 0 : Timer mode 0 1 : Period measurement mode 1 0 : Event counter mode 1 1 : Pulse width HL continuously measurement mode	0	○	○
5			0	○	○
6	CNTR$_1$ active edge switch bit	•CNTR$_1$ interrupt 0 : Falling edge active 1 : Rising edge active •Period measurement mode 0 : Measure falling edge to falling edge 1 : Measure rising edge to rising edge •Event counter mode 0 : Rising edge active 1 : Falling edge active	0	○	○
7	Timer Y stop control bit	0 : Count start 1 : Count stop	0	○	○

Fig. 2.3.17 Structure of timer Y mode register

■Timer Y operating mode bits (bit 4 and bit 5)

The timer Y operating mode bits select a operating mode of the timer Y.

Table 2.3.3 shows the relation between the timer Y operating mode bits and the operating modes. For an explanation of each mode operation, refer to the section pertaining to the explanation of each operation.

Table 2.3.3 Relation between timer Y operating mode bits and operating modes

b5	b4	Operation mode
0	0	Timer mode
0	1	Period measurement mode
1	0	Event counter mode
1	1	Pulse width HL continuously measurement mode

■CNTR1 active edge switch bit (bit 6)

The CNTR1 active edge switch bit has a function which selects an active edge of the CNTR1 interrupt and functions for each mode.

In the pulse width HL continuously measurement mode, this bit is invalid.

●CNTR1 interrupt

When bit 6 is "0," the falling edge (↓) is active.

When bit 6 is "1," the rising edge (↑) is active.

In the pulse width HL continuously measurement mode, an interrupt request occurs at the both edges regardless of the value of this bit.

●Period measurement mode

This bit selects the duration which is measured.

When bit 6 is "0," the falling edge to the falling edge duration is measured.

When bit 6 is "1," the rising edge to the rising edge duration is measured.

●Event counter mode

An active edge of the count source is selected.

When bit 6 is "0," the rising edge (↑) is active.

When bit 6 is "1," the falling edge (↓) is active.

■Timer Y stop control bit (bit 7)

The timer Y stop control bit controls the count operation of the timer Y.

By writing "0" to bit 7, a count source is input to the Y counter, so that a count operation is started. As bit 7 is in the "0" state immediately after reset release, the count operation is automatically started after reset release.

By writing "1" to bit 7, the input of count source to the Y counter is stopped, so that the count operation stops.

At read, this bit functions as a status bit to indicate the operating state (counting or stop) of the counter. When bit 7 is "0," the counter is in the operating state. When bit 7 is "1," the counter is in the stop state.

(6) Interrupt request register 1 (IREQ1) and interrupt request register 2 (IREQ2)

The interrupt request register 1 (address 003C16) and the interrupt request register 2 (address 003D16) indicate whether an interrupt request has occured or not.

Figure 2.3.18 shows the structure of the interrupt request register 1 and Figure 2.3.19 shows the structure of the interrupt request register 2.

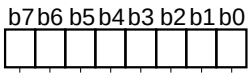
The occurrence of an interrupt request (timer X, timer Y, CNTR0, and CNTR1 interrupt requests) causes the corresponding bit to be set to "1." This interrupt request bit is automatically cleared to "0" by the acceptance of the interrupt request.

The interrupt request bits can be set to "0" by software, but it cannot be set to "1" by software.

The occurrence of each interrupt is controlled by the corresponding interrupt enable bit (refer to the next item).

For details of interrupts, refer to **"2.2 Interrupts."**

Interrupt request register 1



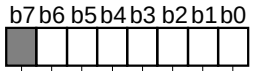
Interrupt request register 1 (IREQ1) [Address 3C 16]

B	Name	Functions	At reset	R	W
0	INT0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

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

Fig. 2.3.18 Structure of interrupt request register 1

Interrupt request register 2



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

B	Name	Functions	At reset	R	W
0	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key input interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	ADT/A-D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated. This bit cannot be written to and is fixed to “0” at reading.		0	0	×

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

Fig. 2.3.19 Structure of interrupt request register 2

(7) Interrupt control register 1 (ICON1) and interrupt control register 2 (ICON2)

The interrupt control register 1 (address 003E16) and the interrupt control register 2 (address 003F16) control each interrupt request source.

Figure 2.3.20 shows the structure of the interrupt control register 1 and Figure 2.3.21 shows the structure of the interrupt control register 2.

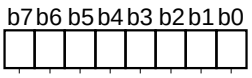
When an interrupt enable bit (timer X, timer Y, CNTR0, and CNTR1 interrupt enable bits) is "0," the corresponding interrupt request is disabled. If an interrupt request occurs when this bit is "0," the corresponding interrupt request bit only is set to "1," and the interrupt request is not accepted.

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

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

For details of interrupts, refer to **"2.2 Interrupts."**

Interrupt control register 1



Interrupt control register 1 (ICON1) [Address 3E 16]

B	Name	Functions	At reset	R	W
0	INT0 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	INT1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	Timer 2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Timer 3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○

Fig. 2.3.20 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	Functions	At reset	R	W
0	CNTR0 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	CNTR1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Timer 1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	INT2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	INT3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Key input interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	ADT/A-D conversion interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Fixed this bit to “0.”		0	0	0

Fig. 2.3.21 Structure of interrupt control register 2

2.3.4 Register setting example

In the following, an example of setting registers for using each mode of the timer X and timer Y is described.

(1) Timer X

■Timer mode

Figure 2.3.22 shows an example of setting registers for using the timer mode.

[Notes on use]

Notes 1: For using interrupt processing, set the following :
- •Before setting ① below, clear the timer X interrupt enable bit and the timer X interrupt request bit to "0."
- •After setting ④ below, set the timer X interrupt enable bit to "1" (interrupts enabled) .

2: Write values in the order of the timer X (low-order) and the timer X (high-order).

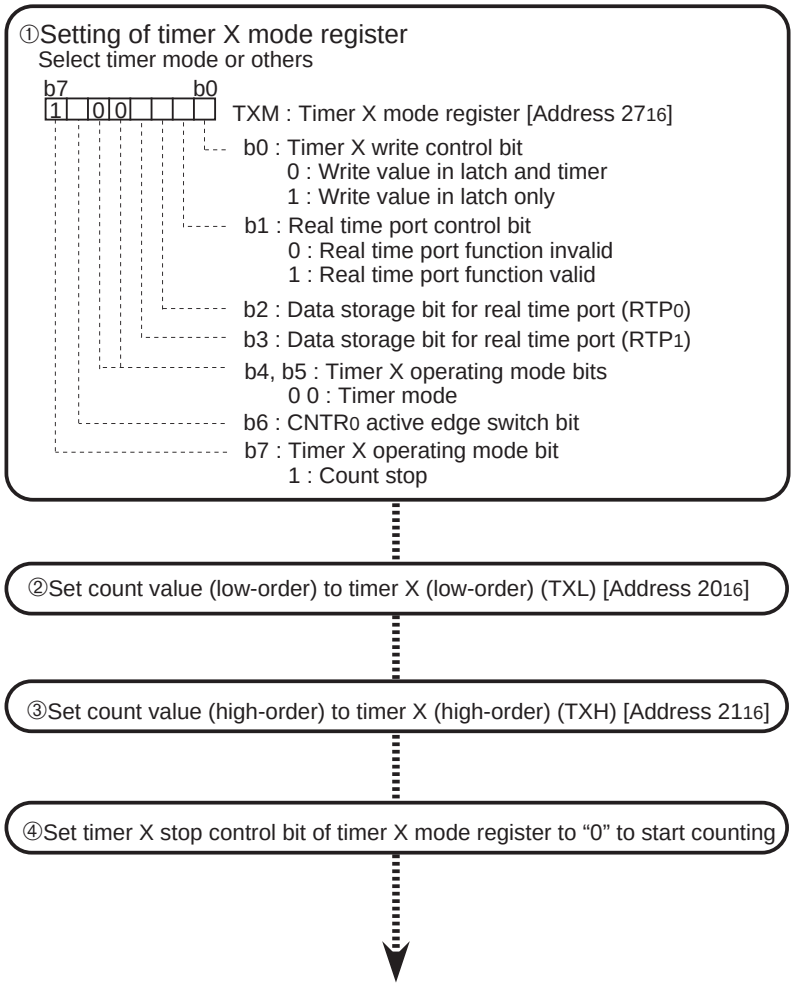


Fig. 2.3.22 Example of setting registers for using timer mode

■**Pulse output mode**

Figure 2.3.23 shows an example of setting registers for using the pulse output mode.

[Notes on use]

Notes 1: For using interrupt processing, set the following :
- Before setting ① below, clear the interrupt enable bits (timer X or CNTR0) and the interrupt request bits (timer X or CNTR0) to "0."
- After setting ⑤ below, set the interrupt enable bits (timer X or CNTR0) to "1" (interrupts enabled) .

2: Write values in the order of the timer X (low-order) and the timer X (high-order).

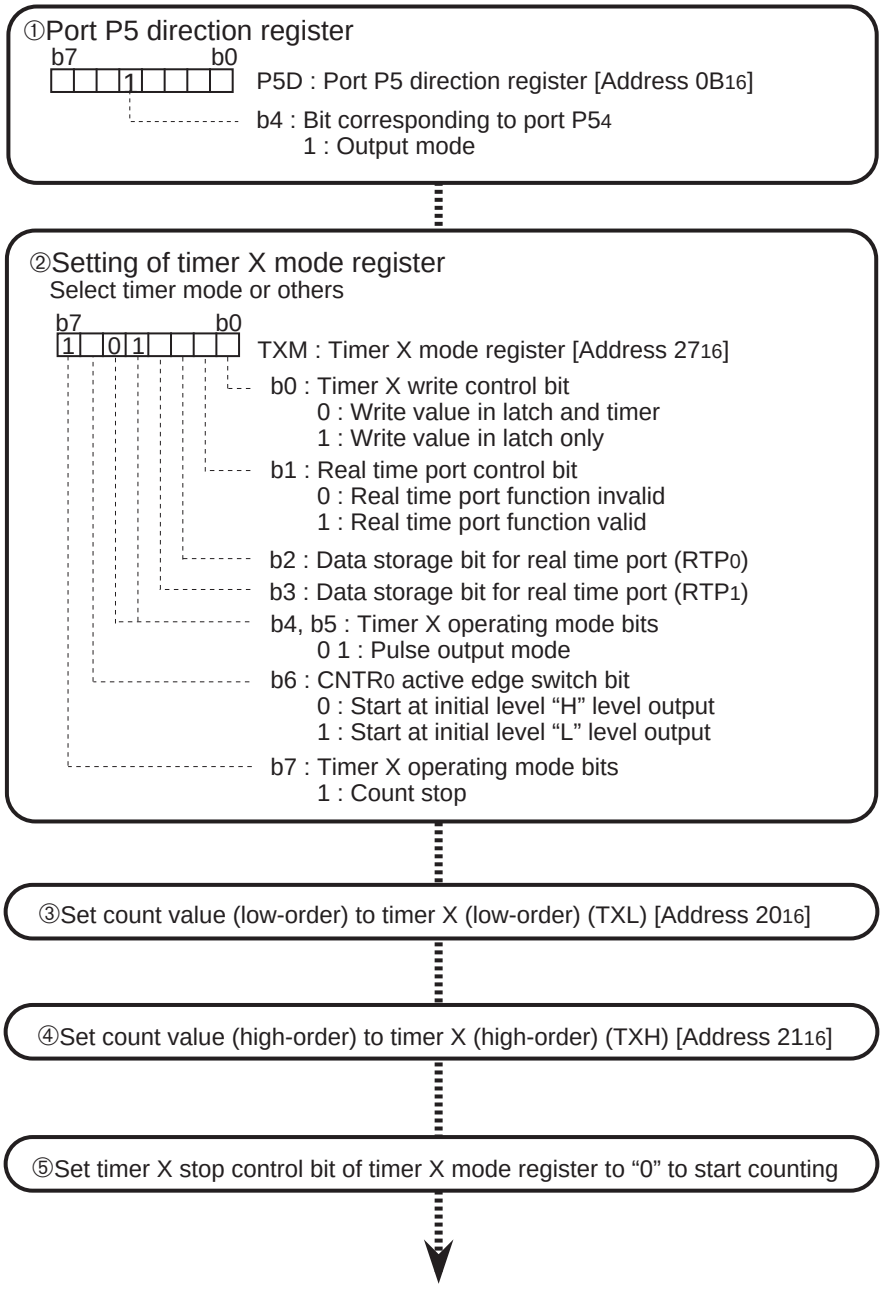


Fig. 2.3.23 Example of setting registers for using pulse output mode

■**Event counter output mode**

Figure 2.3.24 shows an example of setting registers for using the event counter mode.

[Notes on use]

Notes 1: For using interrupt processing, set the following :
- •Before setting ① below, clear the interrupt enable bits (timer X or CNTR0) and the interrupt request bits (timer X or CNTR0) to "0."
- •After setting ⑤ below, set the interrupt enable bits (timer X or CNTR0) to "1" (interrupts enabled) .

2: Write values in the order of the timer X (low-order) and the timer X (high-order).

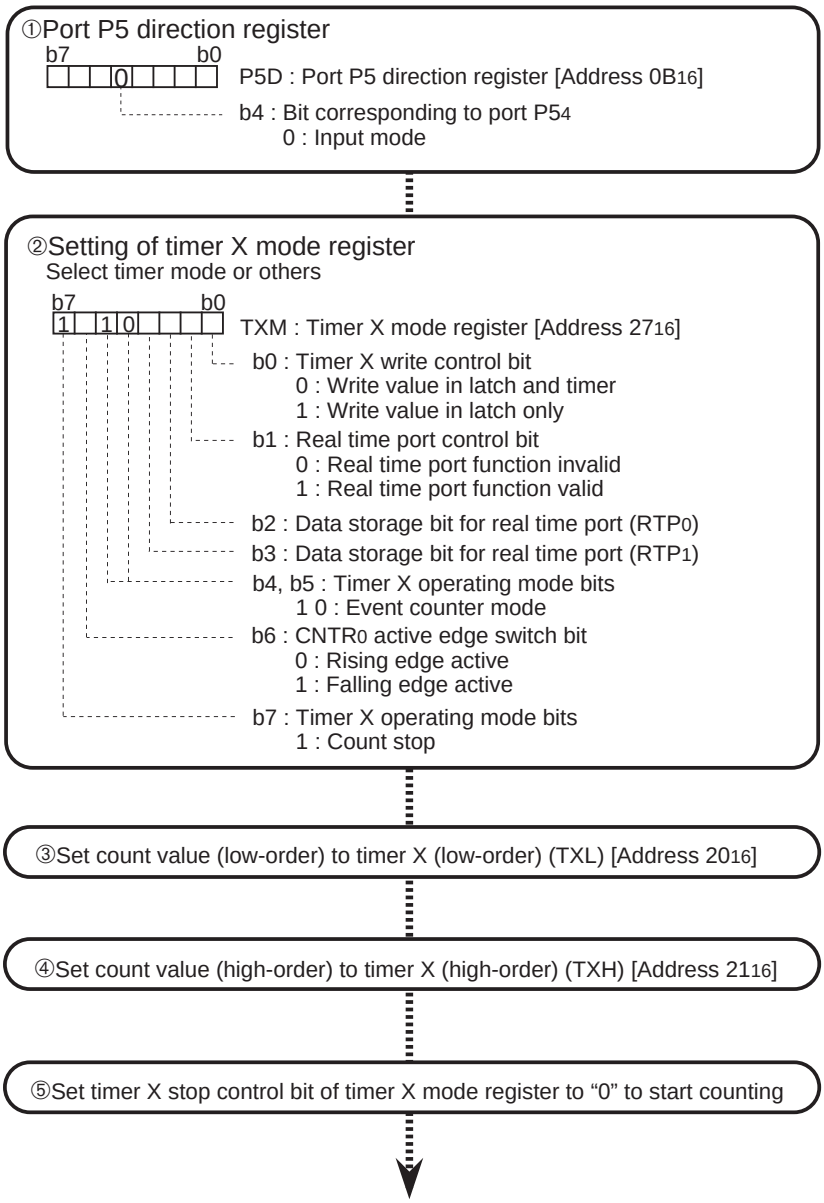


Fig. 2.3.24 Example of setting registers for using event counter mode

■**Pulse width measurement mode**

Figure 2.3.25 shows an example of setting registers for using the pulse width measurement mode.

[Notes on use]

Notes 1: For using interrupt processing, set the following :
- Before setting ① below, clear the interrupt enable bits (timer X or CNTR0) and the interrupt request bits (timer X or CNTR0) to "0."
- After setting ⑤ below, set the interrupt enable bits (timer X or CNTR0) to "1" (interrupts enabled) .

2: Write values in the order of the timer X (low-order) and the timer X (high-order).

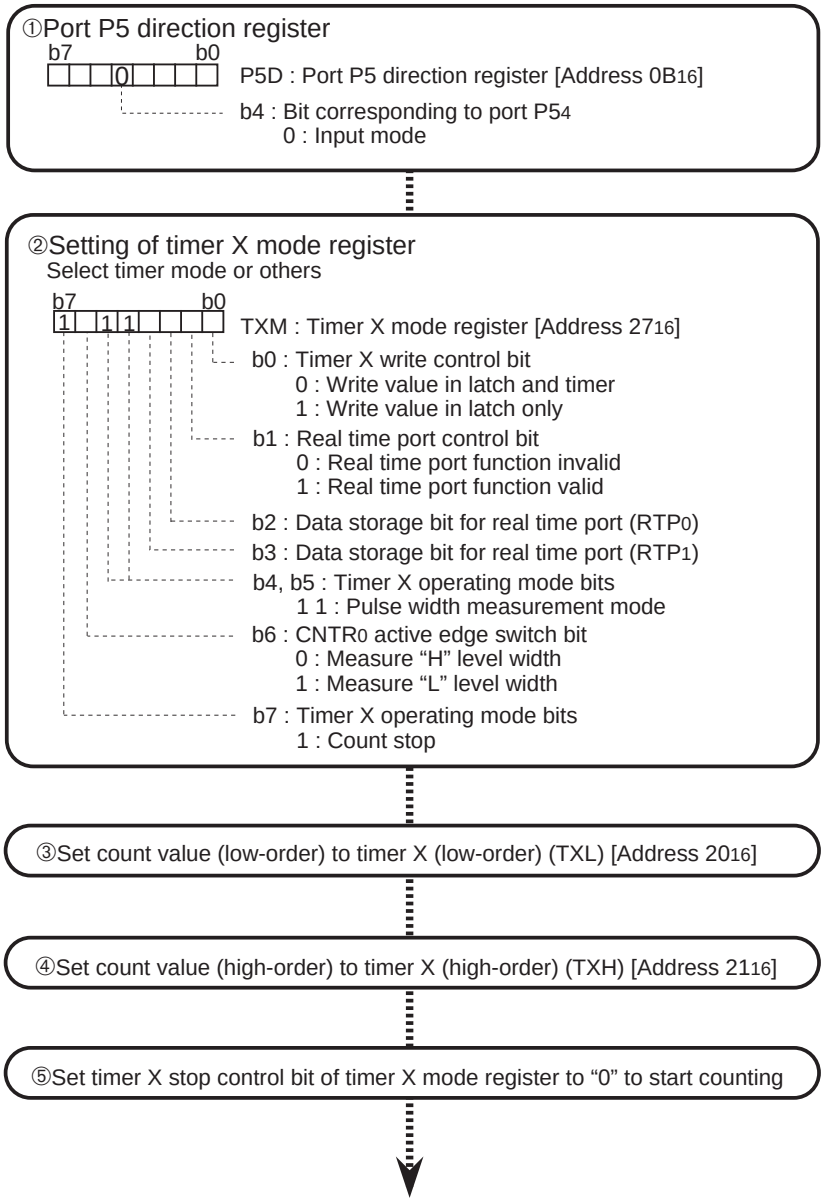


Fig. 2.3.25 Example of setting registers for using pulse width measurement mode

■**Real time port function**

Figure 2.3.26 shows an example of setting registers for using the real time port (referred as RTP) function.

[Notes on use]

Notes 1: After reset release, port P5 direction register is set for the input mode, so pins P52/RTP0 and P53/RTP1 operate as ordinary input ports. For using as real time ports, be sure to set the corresponding bits of the port P5 direction register for the output mode.

2: Change RTP output data as required, for example, by using an interrupt.

3: Do not change ports P52 and P53 selected as RTP into input pins during RTP operation.

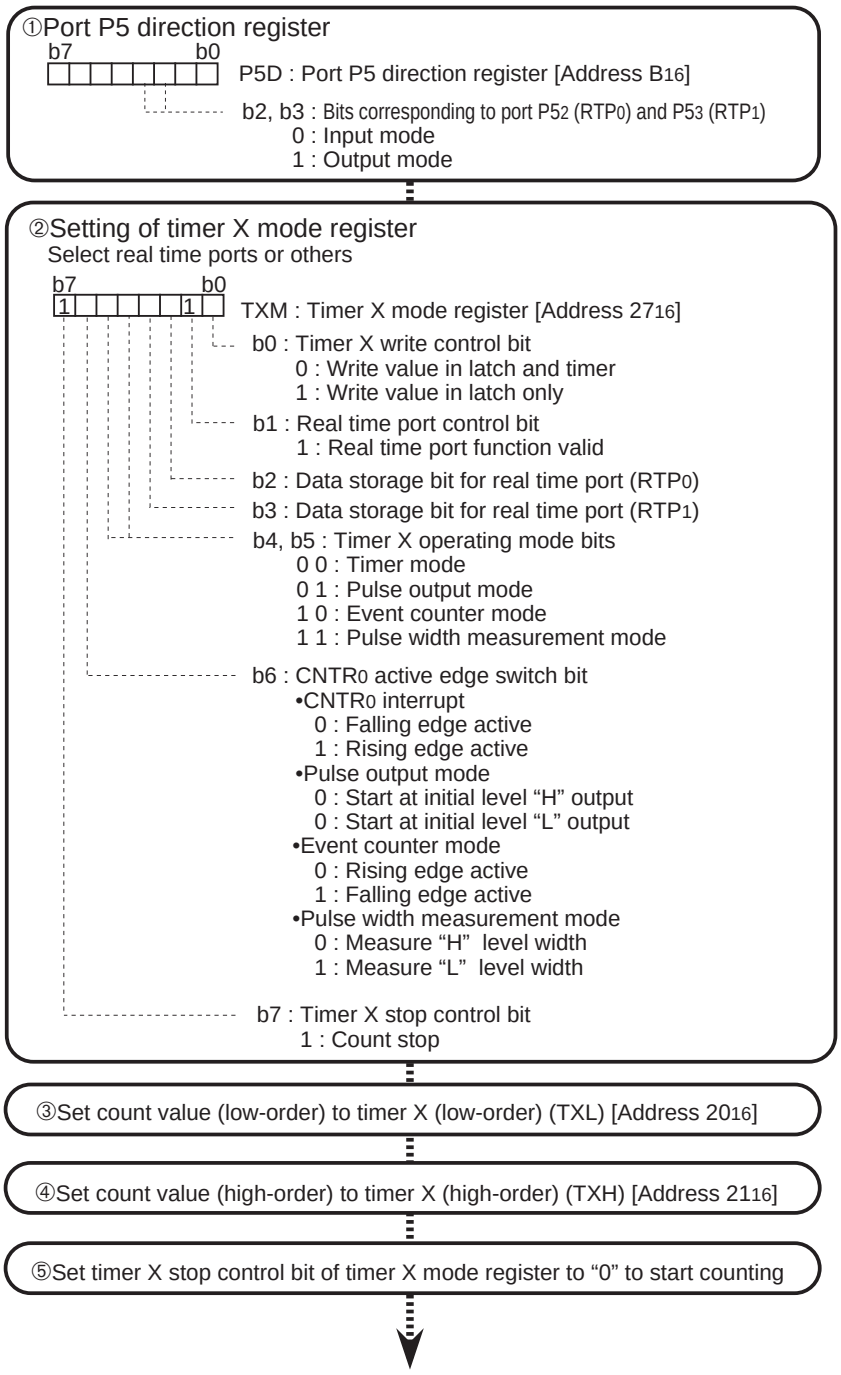


Fig. 2.3.26 Example of setting registers for using real time port

(2) Timer Y

■Timer mode

Figure 2.3.27 shows an example of setting registers for using the timer mode.

[Notes on use]

Notes 1: For using interrupt processing, set the following :

•Before setting ① below, clear the timer Y interrupt enable bit and the timer Y interrupt request bit to "0."

•After setting ④ below, set the timer Y interrupt enable bit to "1" (interrupts enabled) .

2: Write values in the order of the timer Y (low-order) and the timer Y (high-order).

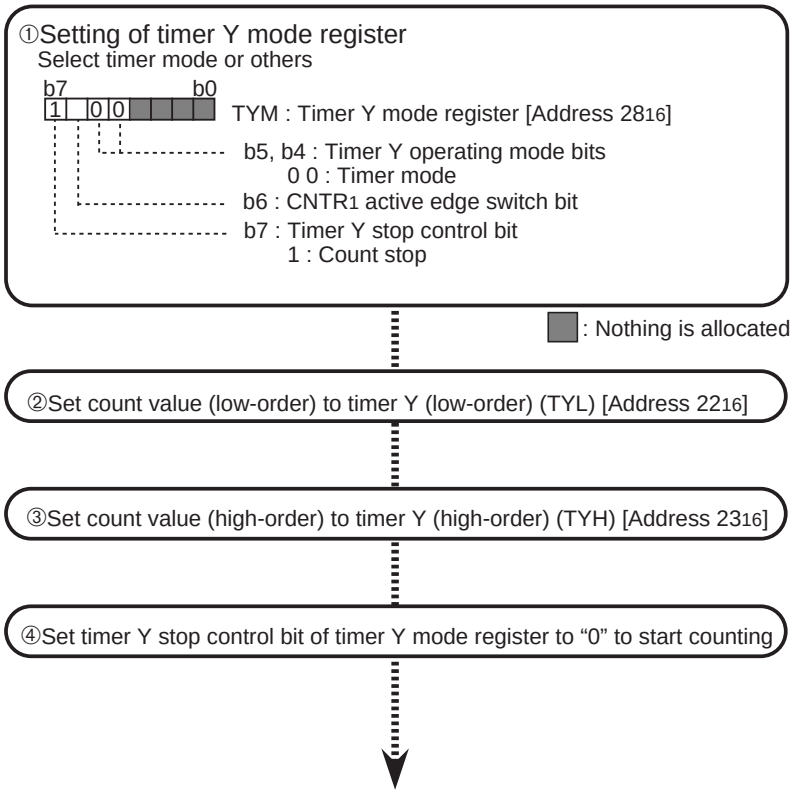


Fig. 2.3.27 Example of setting registers for using timer mode

■**Period measurement mode**

Figure 2.3.28 shows an example of setting registers for using the period measurement mode.

[Notes on use]

Notes 1: For using interrupt processing, set the following :
- •Before setting ① below, clear the interrupt enable bits (timer Y or CNTR 1) and the interrupt request bits (timer Y or CNTR 1) to "0."
- •After setting ⑤ below, set the interrupt enable bits (timer Y or CNTR 1) to "1" (interrupts enabled) .

2: Write values in the order of the timer Y (low-order) ant the timer Y (high-order).

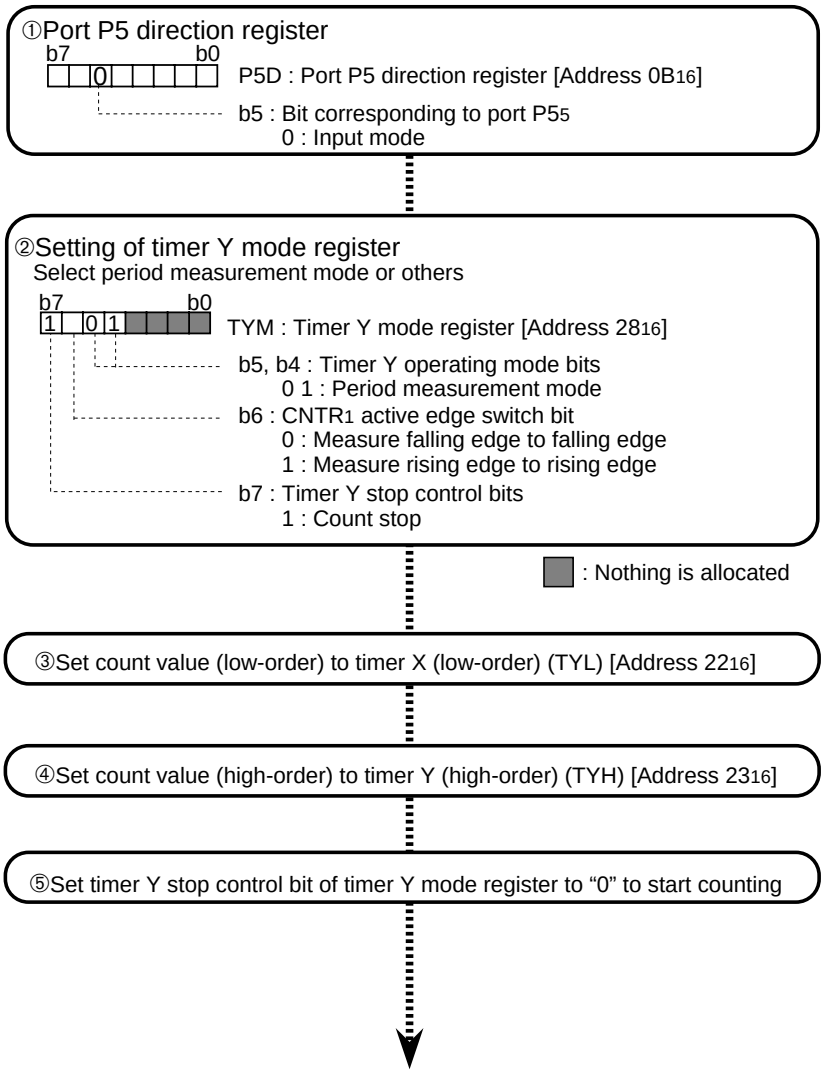


Fig. 2.3.28 Example of setting registers for using period measurement mode

■**Event counter mode**

Figure 2.3.29 shows an example of setting registers for using the event counter mode.

[Notes on use]

Notes 1: For using interrupt processing, set the following :
- Before setting ① below, clear the interrupt enable bits (timer Y or CNTR1) and the interrupt request bits (timer Y or CNTR1) to "0."
- After setting ⑤ below, set the interrupt enable bits (timer Y or CNTR1) to "1" (interrupts enabled) .

2: Write values in the order of the timer Y (low-order) ant the timer Y (high-order).

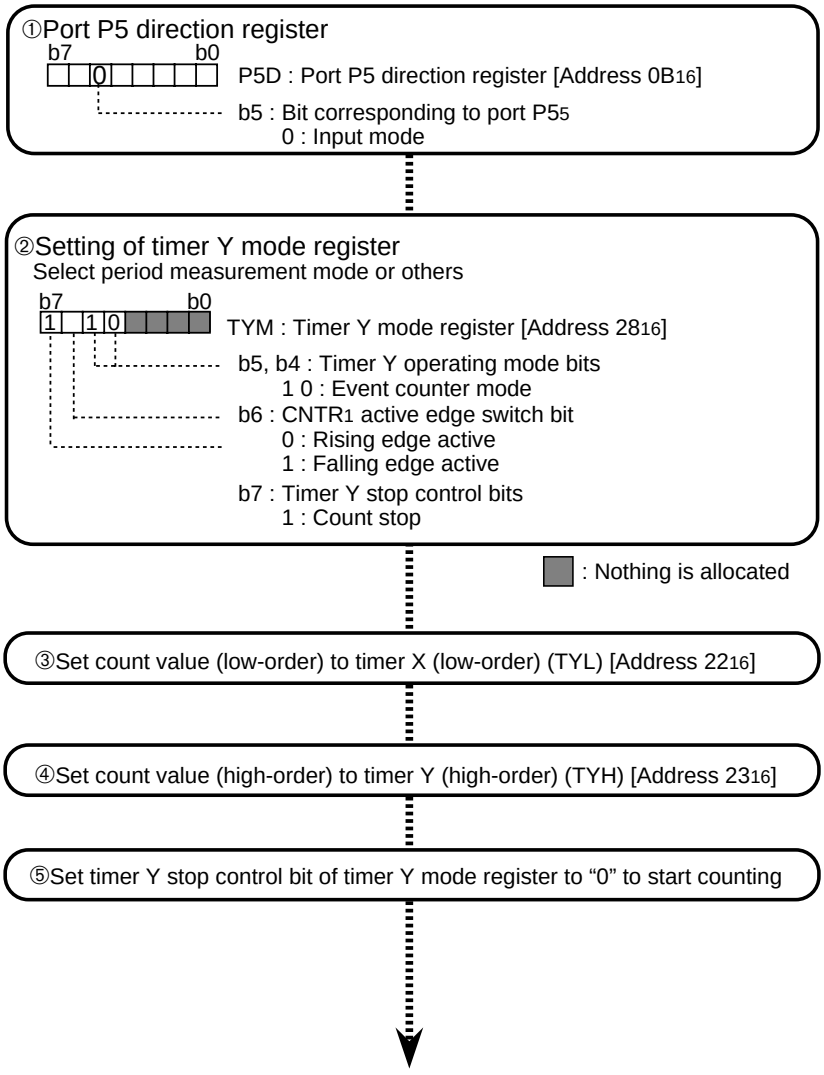


Fig. 2.3.29 Example of setting registers for using event counter mode

■**Pulse width HL countinuously measurement mode**

Figure 2.3.30 shows an example of setting registers for using the pulse width HL countinuously measurement mode.

[Notes on use]

Notes 1: For using interrupt processing, set the following :
- •Before setting ① below, clear the interrupt enable bits (timer Y or CNTR 1) and the interrupt request bits (timer Y or CNTR 1) to "0."
- •After setting ⑤ below, set the interrupt enable bits (timer Y or CNTR 1) to "1" (interrupts enabled) .

2: Write values in the order of the timer Y (low-order) ant the timer Y (high-order).

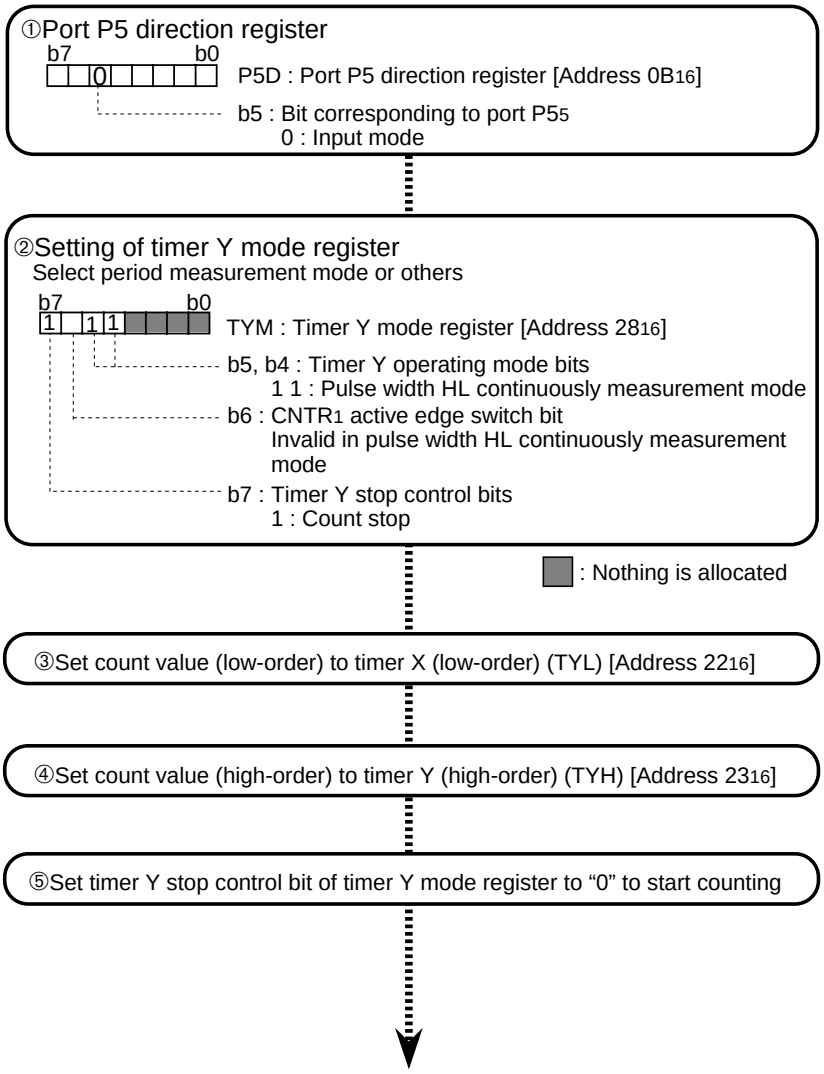


Fig. 2.3.30 Example of setting registers for using pulse width HL continuously measurement mode

2.3.5 Application examples

(1) Pulse output mode : Piezoelectric buzzer output

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

Specifications : •The rectangular waveform which is divided clock f(XIN) = 8 MHz up to about 2 kHz is output from the P54/CNTR0 pin.

•The level of the P54/CNTR0 pin fixes to "H" while a piezoelectric buzzer output is stopped.

Figure 2.3.31 shows an example of a peripheral circuit, Figure 2.3.32, a connection of the timer and a setting of the division ratio, Figure 2.3.33, setting of the related registers, and Figure 2.3.34, the control procedure.

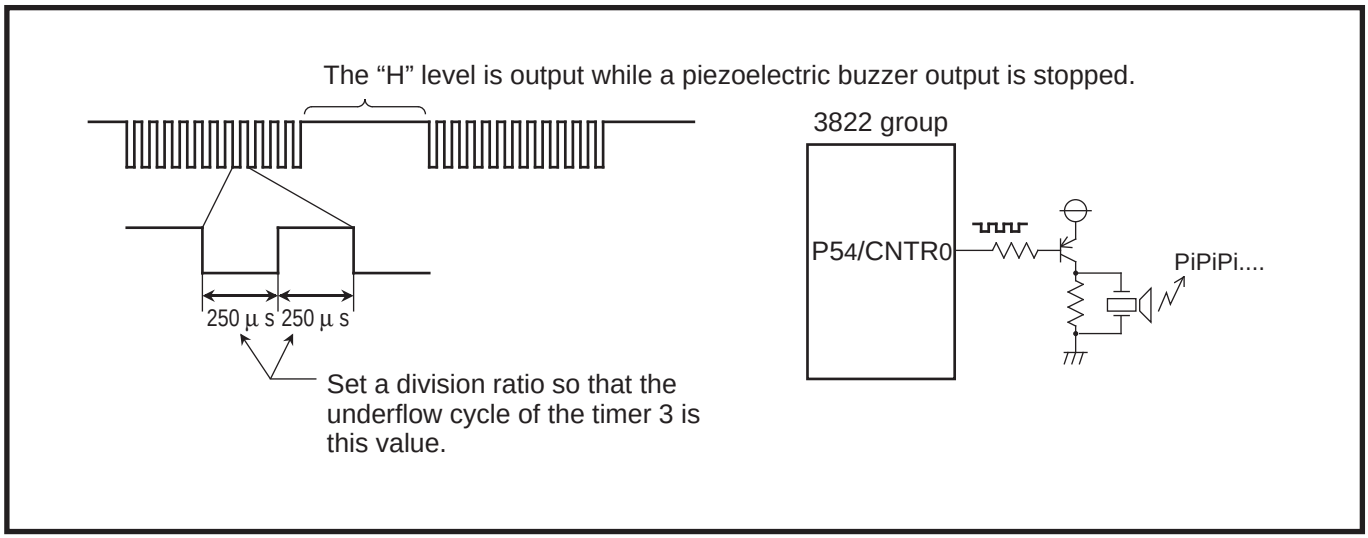


Fig. 2.3.31 Example of peripheral circuit

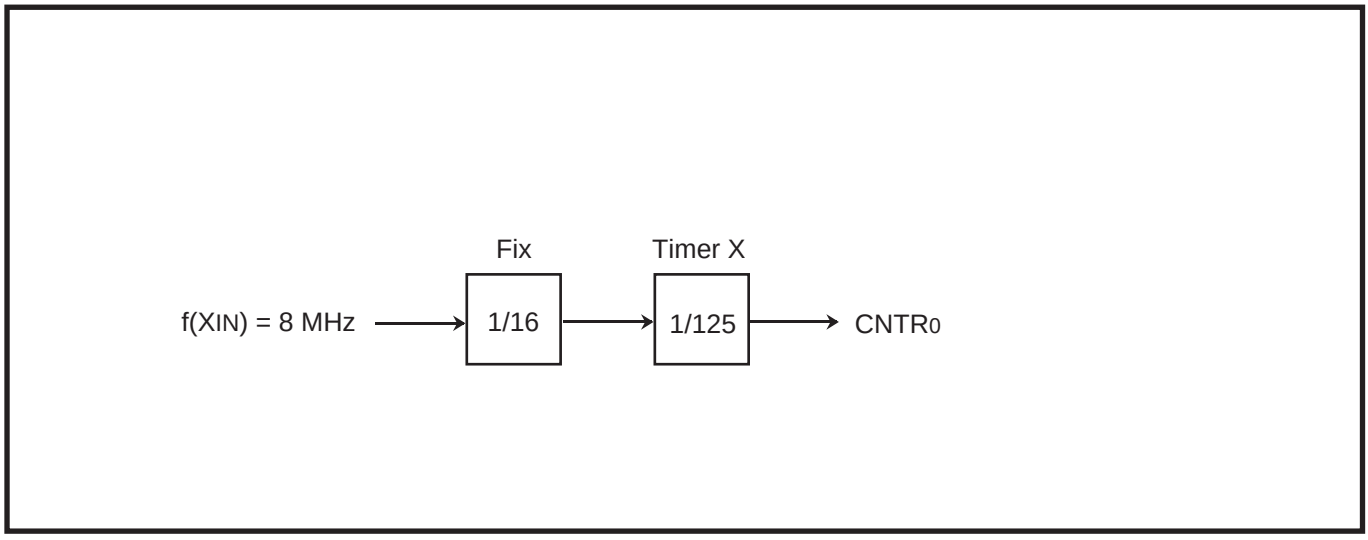


Fig. 2.3.32 Connection of timer and setting of division ratio

b7 b0
X X X 1 X X X X P5D : Port P5 direction register [Address 0B16]

b4 : Bit corresponding to port P54
1 : Output mode

b7 b0
1 X 0 1 X X X X TXM : Timer X mode register [Address 2716]

b4, b5 : Timer X operating mode bits
0 1 : Pulse output mode

b7 : Timer X stop control bit
1 : Count stop

7C16 TXL : Timer X (low-order) [Address 2016]

0016 TXH : Timer X (high-order) [Address 2116]

Note : Write values in the order of the low-order byte and the high-order byte.

Set "division ratio – 1 (124 : 007C16)" in the timer X register

b7 b0
X X X 1 X X X X ICON1 : Interrupt control register [Address 3E16]

b4 : Timer X interrupt enable bit
1 : Interrupt enabled

Fig. 2.3.33 Setting of related registers

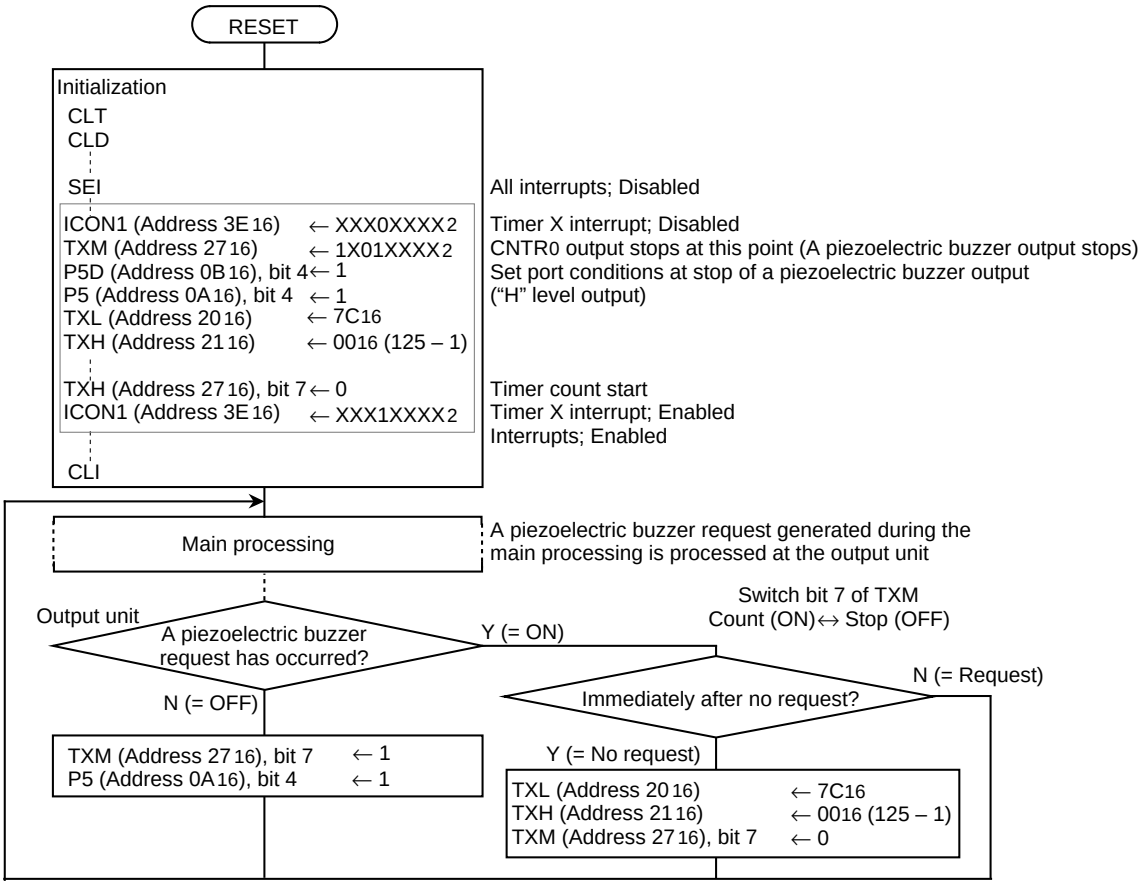


Fig.2.3.34 Control procedure

(2) Pulse width measurement mode: Ringer signal detection

Outline : A telephone ringing pulse*1 is detected by applying the timer X interrupt and the pulse width measurement mode.

Specifications : •Whether a telephone call exists or not is judged by measuring a pulse width output from the "H" active ringing pulse detection circuit.
•f(X_{IN}) = 8 MHz is used as the count source.
•When the following condition is satisfied, it is regard as normal.
200 ms ≤ pulse width of a ringing pulse < 1.2 s

Figure 2.3.35 shows an example of a peripheral circuit, Figure 2.3.36, setting of the related registers, Figure 2.3.37, a ringing pulse waveform, Figure 2.3.38, an operation timing when a ringing pulse is input, and Figure 2.3.39, the control procedure.

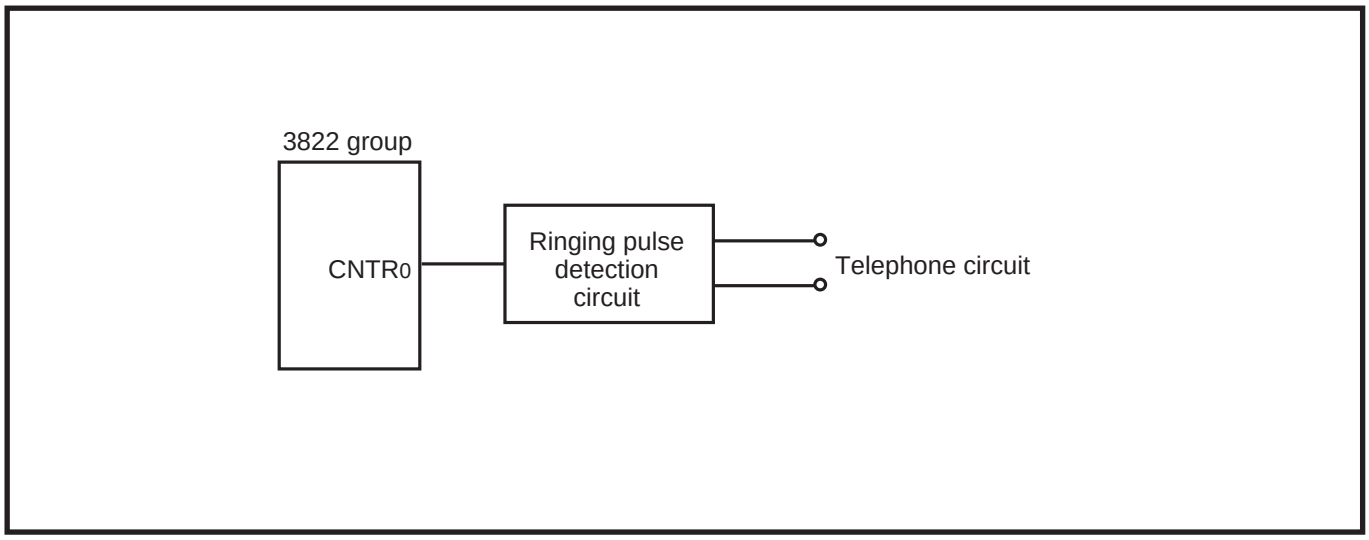


Fig. 2.3.35 Example of peripheral circuit

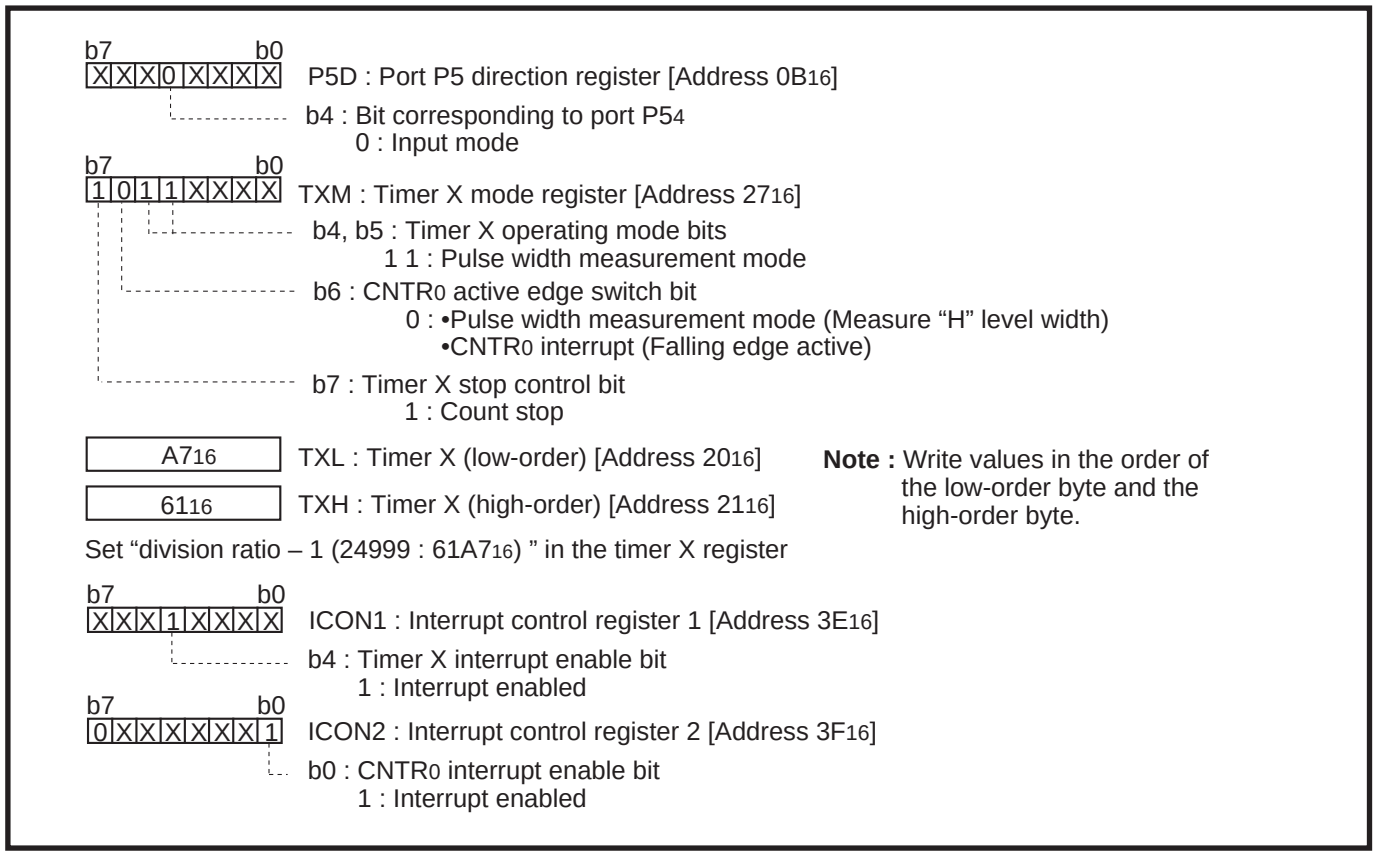
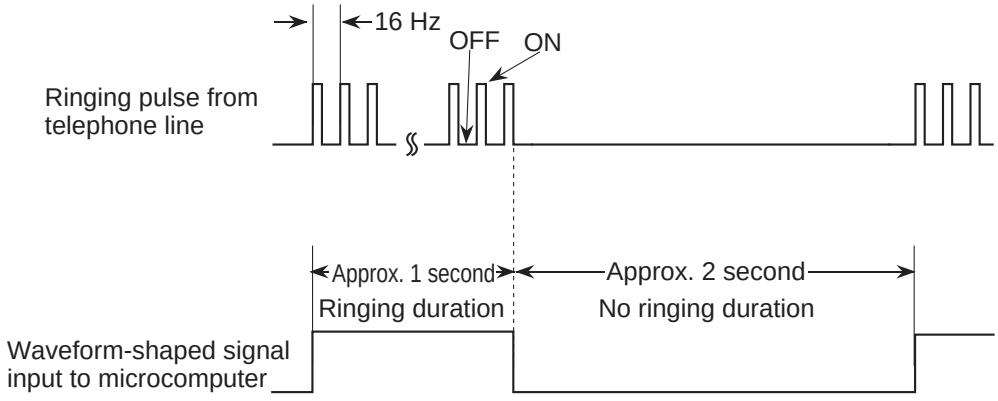


Fig. 2.3.36 Setting of related registers



✻ Ringing pulse : Signal which is sent by turning on/off (make/break) the telephone line. Each country has a different standard. In this case, Japanese domestic standard is adopted as an example.

Fig. 2.3.37 Ringer signal waveform

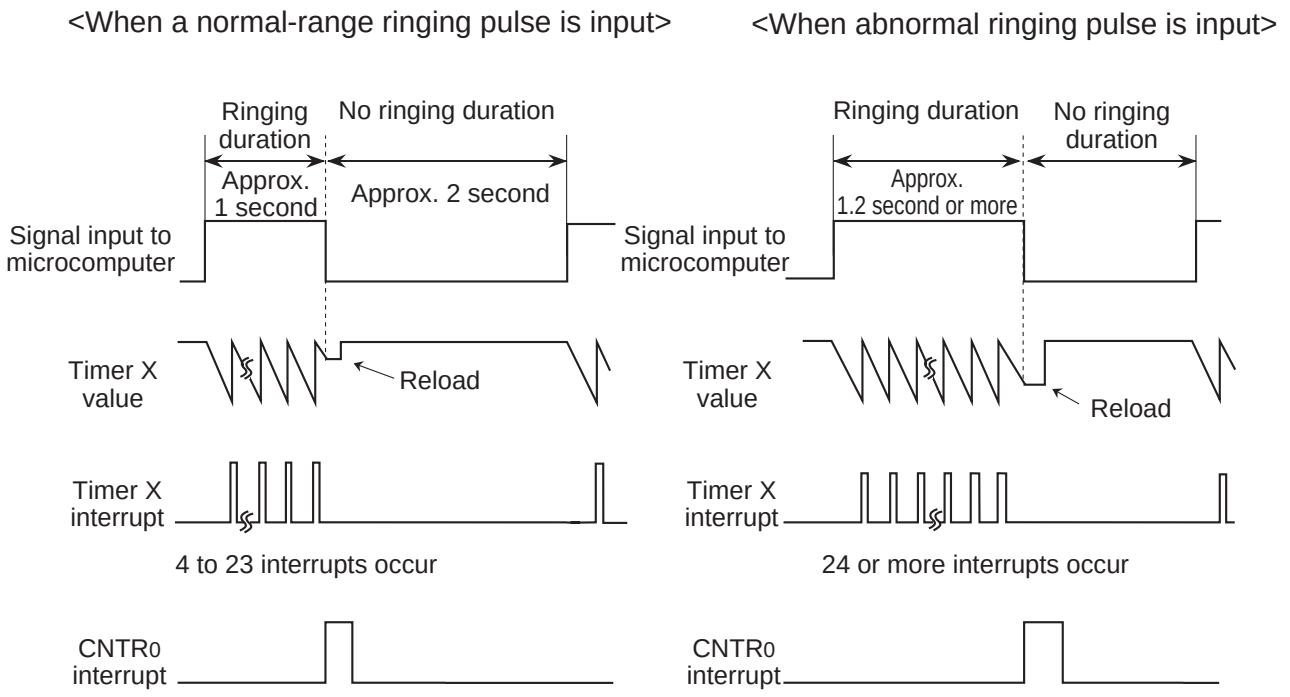


Fig. 2.3.38 Operation timing when ringing pulse is input

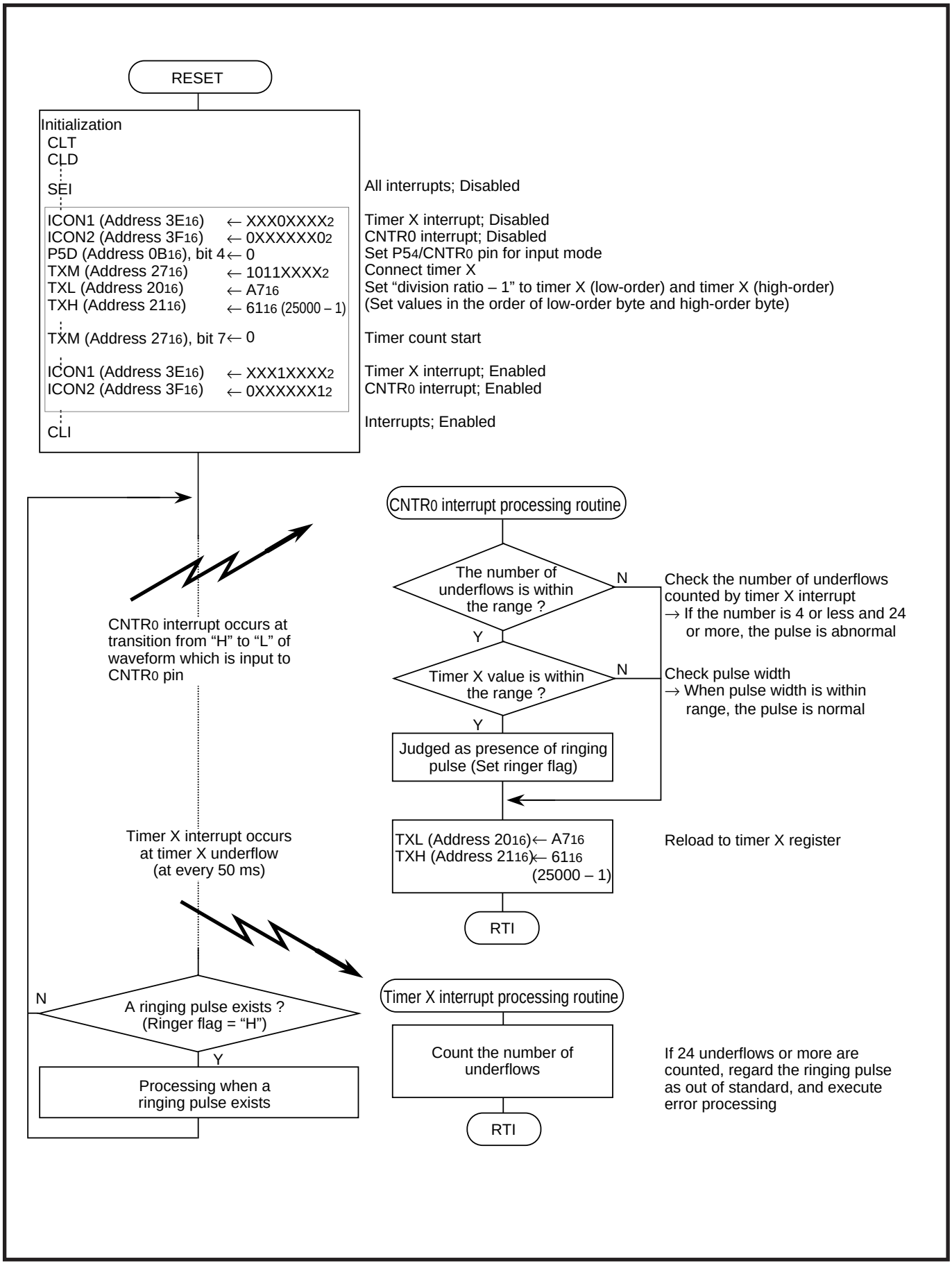


Fig. 2.3.39 Control procedure

(3) Real time port function

Figure 2.3.40 shows a timer X interrupt processing procedure example when the real time port (referred as "RTP") is used. Figure 2.3.41 shows an application connection example when the RTP is used. Figure 2.3.42 shows an RTP output example. Table 2.3.4 and Table 2.3.5 show table examples for it.

[Notes on use]

Notes 1 : When there is no necessity for changing the timer X underflow time in ②, omit it.

2 : When writing to the latch only is selected as the timer X write control, the timer X value (TXL, TXH) is rewritten at the first underflow after ②.

3 : Execute another timer X interrupt processing in ① to ④.

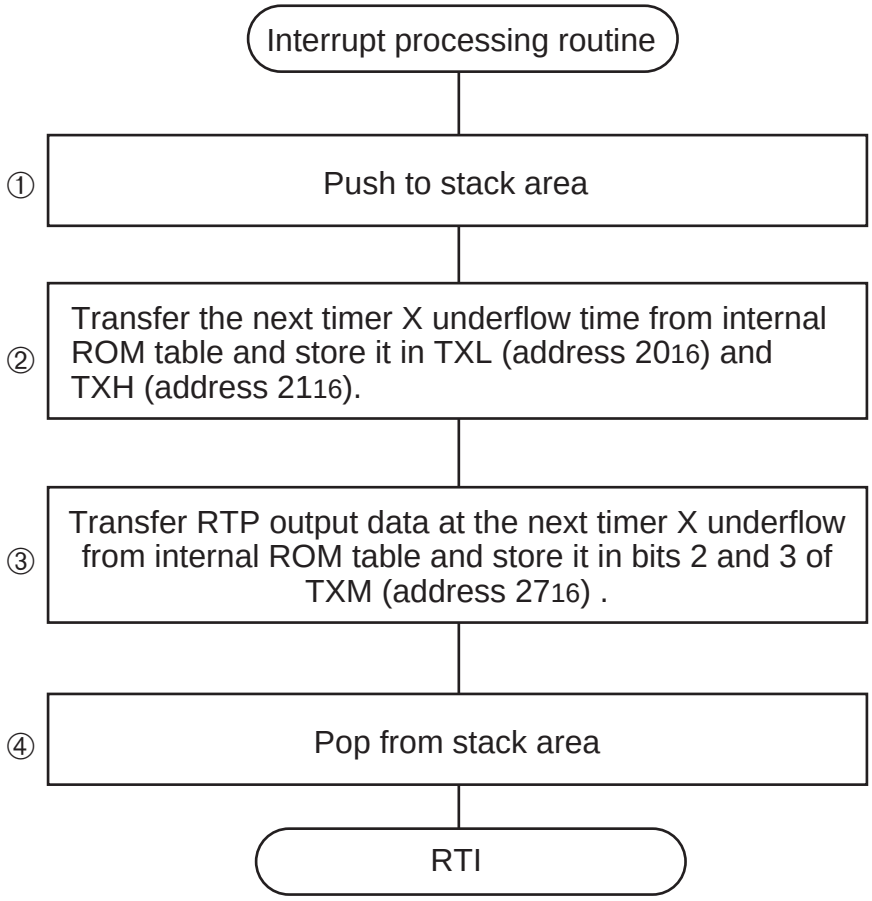


Fig. 2.3.40 Timer X interrupt processing procedure example when real time port is used

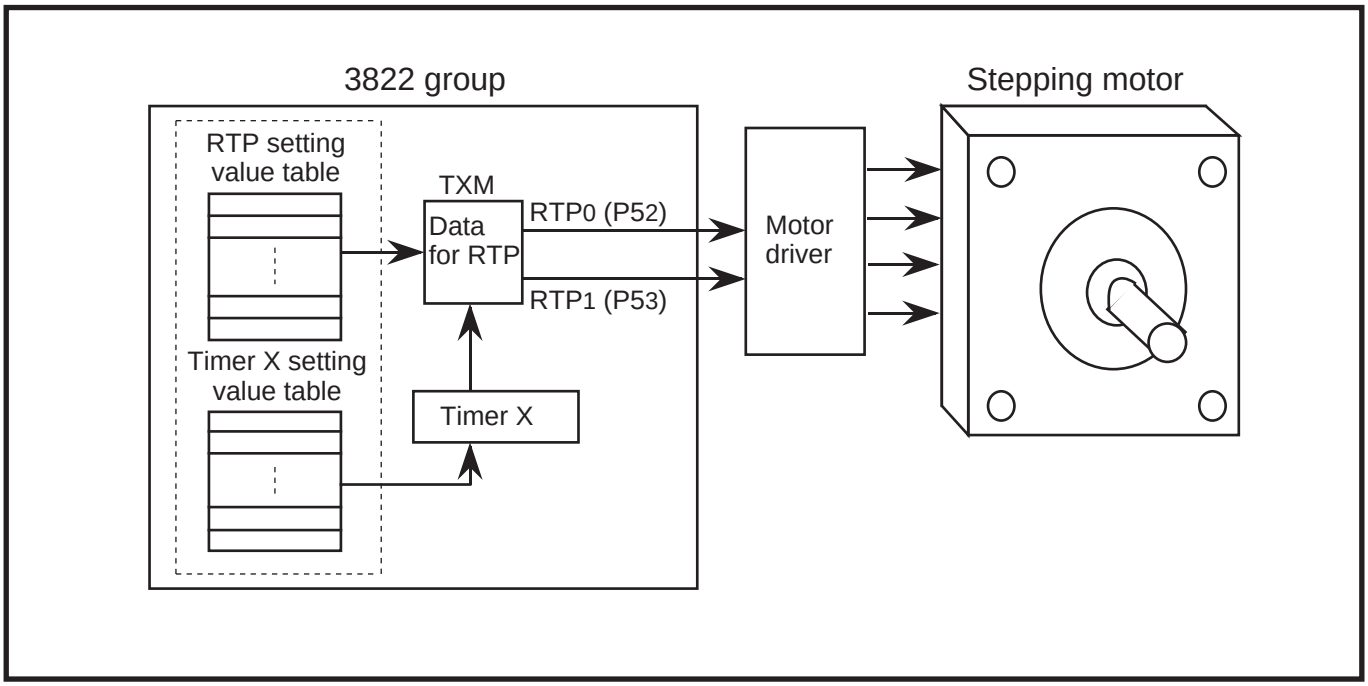


Fig. 2.3.41 Application connection example when RTP is used

Table 2.3.4 Table example for timer X setting value

RTP output time	Timer X setting value
T_1	$2FD0_{16}$
T_2	$2B71_{16}$
T_3	2081_{16}
T_4	1869_{16}
T_5	$13C9_{16}$
T_6	$13A9_{16}$
T_7	1221_{16}
T_8	$11C1_{16}$

Table 2.3.5 Table example for RTP setting value

RTP output pattern	RTP setting values	
	TXM, b2	TXM, b3
①	0	0
②	0	1
③	1	0
④	1	1

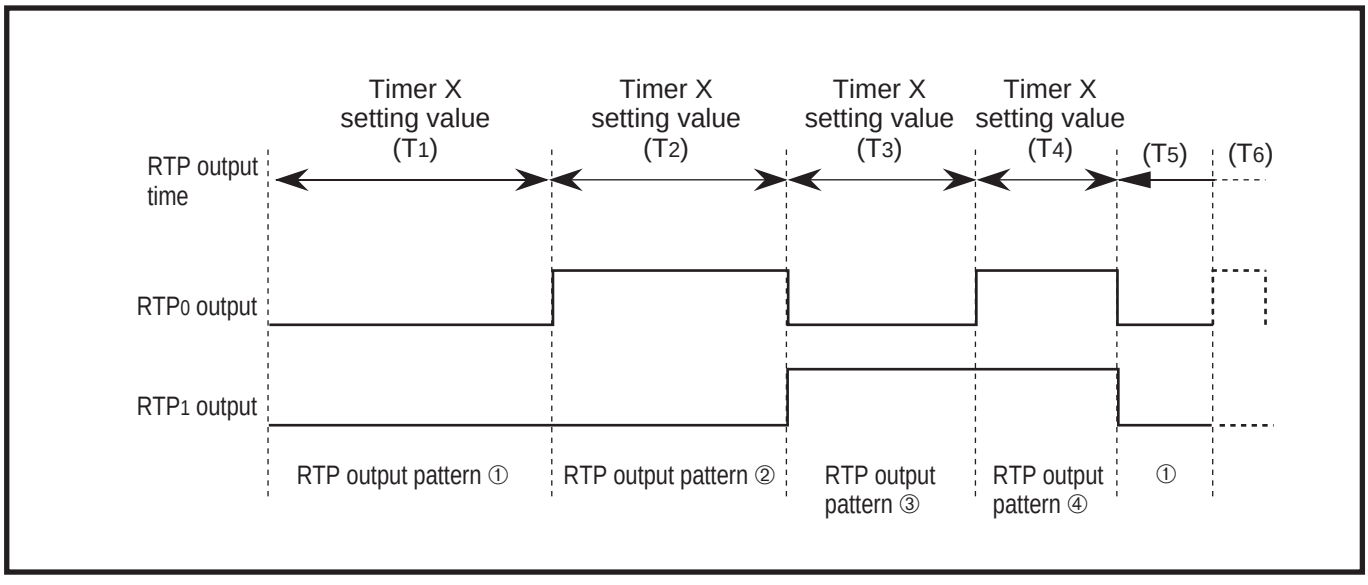


Fig. 2.3.42 RTP output example

2.3.6 Notes on use

Notes on using each mode of the timer X and timer Y are described below.

(1) Timer X

■Common to all modes

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

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

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

•In writing for the latch only (timer X write control bit = "1"), if writing timing for the high-order latch is almost same as the underflow timing, a normal value may not be set in the high-order counter.

■Pulse output mode

●In the pulse output mode, set the bit 4 (corresponding to the $P5_4/CNTR_0$) of the port P5 direction register (address $000B_{16}$) to "1" (output mode).

●When the bit 4 (corresponding to the $P5_4/CNTR_0$) of the port P5 register (address $000A_{16}$) in the pulse output mode is read, the value of the port register are not read out but the output value of the pin is read out.

■Event counter mode

●When using the event counter mode, set the bit 4 (corresponding to the $P5_4/CNTR_0$) of the port P5 direction register (address $000B_{16}$) to "0" (input mode).

●The maximum input frequency in the event counter mode is:

4 MHz (250 ns) ..... at V_{CC} = 4.0 V to 5.5 V

(2 × V_{CC}) – 4 MHz ($\frac{500}{V_{CC} - 2}$ ns) ..... at V_{CC} = 2.5 V to 4.0 V

The minimum "H" pulse width is:

105 ns ..... at V_{CC} = 4.0 V to 5.5 V

($\frac{250}{V_{CC} - 2}$ – 20 ns) ..... at V_{CC} = 2.5 V to 4.0 V

The minimum "L" pulse is:

105 ns ..... at V_{CC} = 4.0 to 5.5 V

($\frac{250}{V_{CC} - 2}$ – 20 ns) ..... at V_{CC} = 2.5 V to 4.0 V

■Pulse width measurement mode

●In the pulse width measurement mode, set the bit 4 (corresponding to $P5_4/CNTR_0$) of the port P5 direction register (address $000B_{16}$) to "0" (input mode).

●In reading the value of the $P5_4/CNTR_0$ pin as an input pin, the value is "1" at "H" level input or "0" at "L" level input regardless of the value of the $CNTR_0$ active edge switch bit.

●Setting the $CNTR_0$ active edge switch bit effects on the active edge of an interrupt. Consequently, a $CNTR_0$ interrupt request may be caused by setting the $CNTR_0$ active edge switch bit.
As a countermeasure against the above, switch the active edge after disabling the $CNTR_0$ interrupt, then set the $CNTR_0$ interrupt request bit to "0."

●The minimum "H" pulse width in the pulse width measurement mode is:

105 ns ..... at V_{CC} = 4.0 V to 5.5 V

($\frac{250}{V_{CC} - 2}$ – 20 ns) ..... at V_{CC} = 2.5 V to 4.0 V

The minimum "L" pulse is:

105 ns ..... at V_{CC} = 4.0 to 5.5 V

($\frac{250}{V_{CC} - 2}$ – 20 ns) ..... at V_{CC} = 2.5 V to 4.0 V

■Real time port function

●After reset release, the port P5 direction register is set for the input mode, so the pins $P5_0$ to $P5_7$ function as ordinary I/O ports. For the pin to be used as RTP, be sure to set the corresponding bits of the port P5 direction register for the output mode.

●For a pin used as RTP, do not change this port for the input mode during real time port operation.

●Change RTP output data as required, for example, by using an interrupt.

(2) Timer Y

■Common to all modes

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

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

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

■Period measurement mode

●In the period measurement mode, set the bit 5 (corresponding to the $P5_5$/$CNTR_1$) of the port P5 direction register (address $000B_{16}$) to "0" (input mode).

●Setting the $CNTR_1$ active edge switch bit effects on the active edge of an interrupt. Consequently, the $CNTR_1$ interrupt request may be caused by setting the $CNTR_1$ active edge switch bit.
As a countermeasure, switch the active edge after disabling the $CNTR_1$ interrupt, then set the $CNTR_1$ interrupt request bit to "0."

●The maximum input frequency in the period measurement mode is:

4 MHz (250 ns) ..... at V_{CC} = 4.0 V to 5.5 V

(2 × V_{CC}) – 4 MHz ($\frac{500}{V_{CC} - 2}$ ns) ..... at V_{CC} = 2.5 V to 4.0 V

The minimum "H" pulse width is:

105 ns ..... at V_{CC} = 4.0 V to 5.5 V

($\frac{250}{V_{CC} - 2}$ – 20 ns) ..... at V_{CC} = 2.5 V to 4.0 V

The minimum "L" pulse is:

105 ns ..... at V_{CC} = 4.0 to 5.5 V

($\frac{250}{V_{CC} - 2}$ – 20 ns) ..... at V_{CC} = 2.5 V to 4.0 V

■Event counter mode

●In the event counter mode, set the bit 5 (corresponding to the $P5_5$/$CNTR_1$) of the port P5 direction register (address $000B_{16}$) to "0" (input mode).

●Setting the $CNTR_1$ active edge switch bit, the active edge of an interrupt is also affected. Consequently, a $CNTR_1$ interrupt request may be caused by setting the $CNTR_1$ active edge switch bit.

●The maximum input frequency in the event counter mode is:

4 MHz (250 ns) ..... at V_{CC} = 4.0 V to 5.5 V

(2 × V_{CC}) – 4 MHz ($\frac{500}{V_{CC} - 2}$ ns) ..... at V_{CC} = 2.5 V to 4.0 V

The minimum "H" pulse width is:

105 ns ..... at V_{CC} = 4.0 V to 5.5 V

($\frac{250}{V_{CC} - 2}$ – 20 ns) ..... at V_{CC} = 2.5 V to 4.0 V

The minimum “L” pulse is:

105 ns ..... at V_{CC} = 4.0 to 5.5 V

$\left(\frac{250}{V_{CC} - 2} - 20 \text{ ns}\right)$ ..... at V_{CC} = 2.5 V to 4.0 V

■Pulse width HL continuously measurement mode

●In the pulse width HL continuously measurement mode, set the bit 5 (corresponding to $P5_5$/$CNTR_1$) of the port P5 direction register (address $000B_{16}$) to “0” (input mode).

●The $CNTR_1$ interrupt request occurs at both edges of input pulses regardless of the value of the $CNTR_1$ active edge switch bit.

●The minimum “H” pulse width in the pulse width HL continuously measurement mode is:

105 ns ..... at V_{CC} = 4.0 V to 5.5 V

$\left(\frac{250}{V_{CC} - 2} - 20 \text{ ns}\right)$ ..... at V_{CC} = 2.5 V to 4.0 V

The minimum “L” pulse is:

105 ns ..... at V_{CC} = 4.0 to 5.5 V

$\left(\frac{250}{V_{CC} - 2} - 20 \text{ ns}\right)$ ..... at V_{CC} = 2.5 V to 4.0 V

2.4 Timer 1, timer 2, and timer 3

2.4.1 Explanation of operations

Timer 1 to timer 3 are 8-bit timers that operate in the timer mode. The timer mode is a count-down system, so the value of the counter is decremented each time a count source is input. When the counter underflows, an interrupt request occurs.
The timer 2 can also output a pulse whose polarity is reversed at each underflow.

(1) Timer mode

Operation of the timers 1 to 3 in the timer mode are described below.

①Start of count operation
A count operation is automatically started after reset release.
The value of the counter is decremented by 1 each time a count source is input.

②Reload operation
The counter underflows at the first count pulse after the value of the counter reaches "00$_{16}$." At this time, the value of the corresponding timer latch is transferred (reloaded) to the counter.

③Interrupt operation
An interrupt request occurs at the counter underflow. At the same time, the corresponding interrupt request bit is set to "1." The occurrence of each interrupt is controlled by the interrupt enable bit. The acceptance of the interrupt request causes the interrupt request bit which has been set to "1" to be automatically cleared to "0." It can also be cleared to "0" by software. An interrupt request occurs each time the counter underflows. In other words, an interrupt request occurs every "the counter initial value + 1" count of the rising edge of the count source.

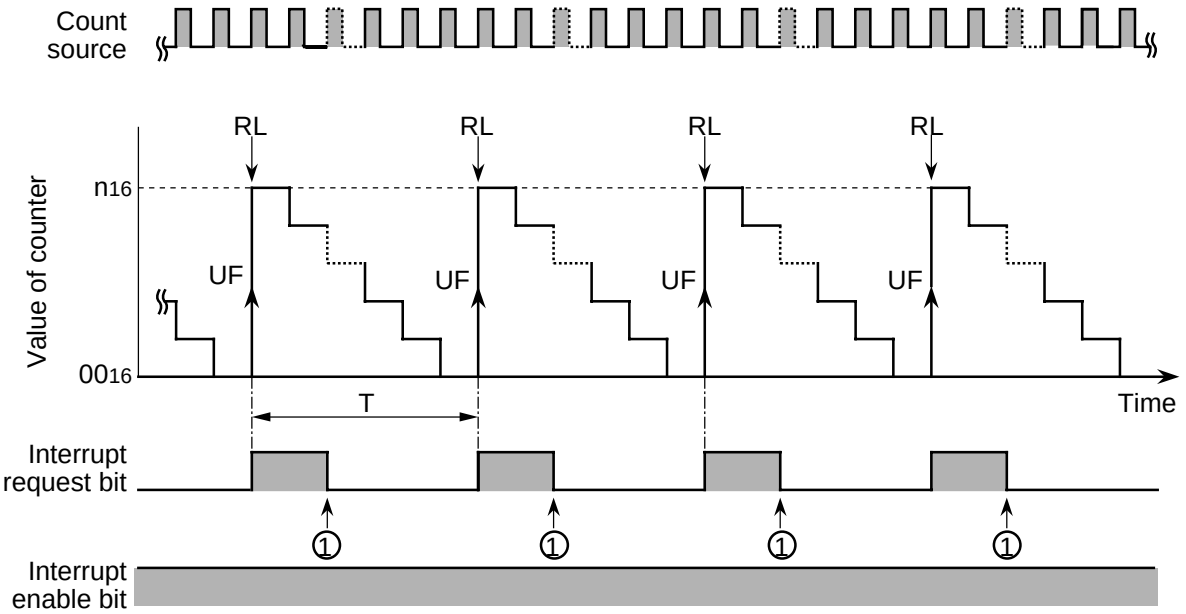
Figure 2.4.1 shows a timer mode operation example.

Count period

Count period T (s) = 1 + count source frequency × (the counter initial value + 1)

Operation example in timer mode

- UF : Underflow
- RL : Reload
- n : The counter initial value



①: •Clearing by writing "0" to the corresponding interrupt request bit of the timers 1 to 3.
•Clearing by accepting the corresponding interrupt request of the timers 1 to 3 when the corresponding interrupt enable bit is "1."

Fig. 2.4.1 Timer mode operation example

(2) Rewriting the value of the counter and the latch

When data is written to the timer, the values of the counter and the latch are rewritten. For rewriting the values of the counters and the latches corresponding to each timer is described below.

■Timer 1 and timer 3

By writing a value to the timer, the value is set simultaneously in both the counter and the latch. Accordingly, the counter period, when a value is written to the timer during counting, becomes inaccurate. Figure 2.4.2 shows an rewriting example of the counter and the latch corresponding to the timers 1 or 3.

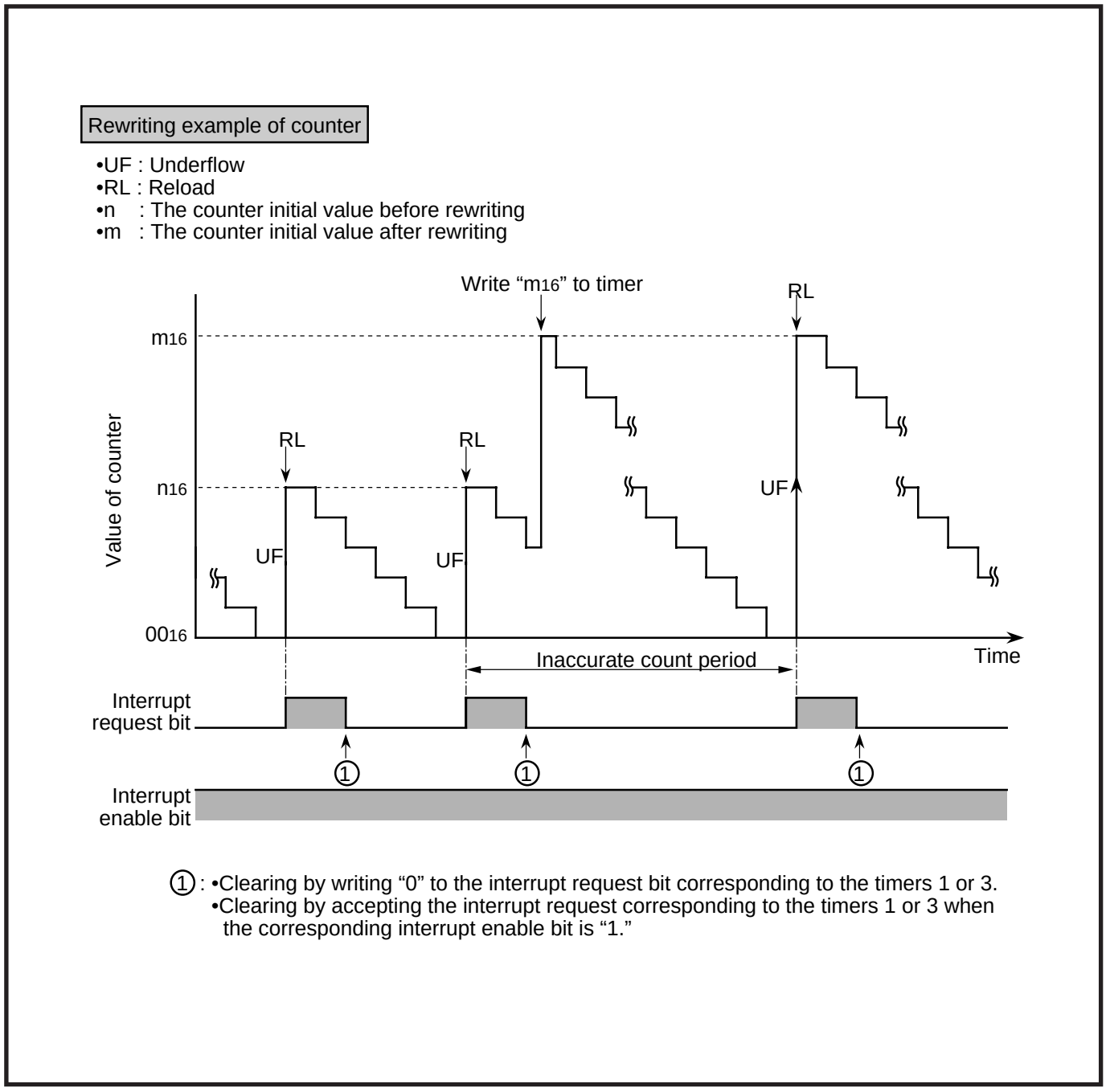


Fig. 2.4.2 Rewriting example of counter and latch corresponding to timers 1 or 3

■Timer 2

The write operation to the timer 2 counter is controlled by the timer 2 write control bit (bit 2 at address 0029_{16}).

(bit 2 = "0")

As the write operation is the same as that to the timer 1 and the timer 3, refer to the previous section, **"■Timer 1 and timer 3."**

(bit 2 = "1")

When a value is written to the timer 2, the value is set in the timer 2 latch only. The rewritten value is reloaded onto the timer 2 counter at the first underflow after rewriting.

Figure 2.4.3 shows an rewriting example of the timer 2 counter and the timer 2 latch.

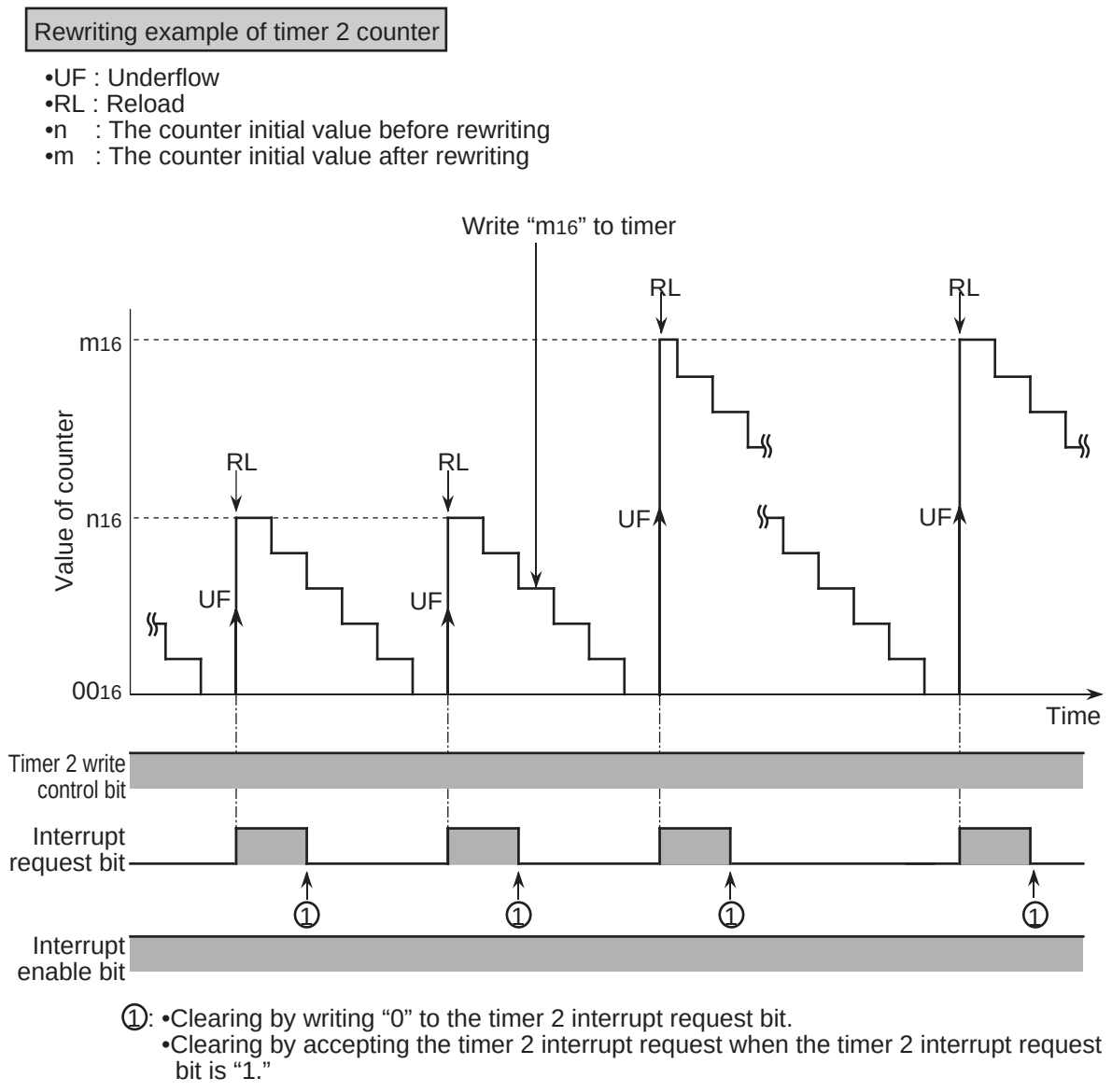


Fig. 2.4.3 Rewriting example of timer 2 counter and timer 2 latch (Writing in timer 2 latch only)

(3) Pulse output by timer 2

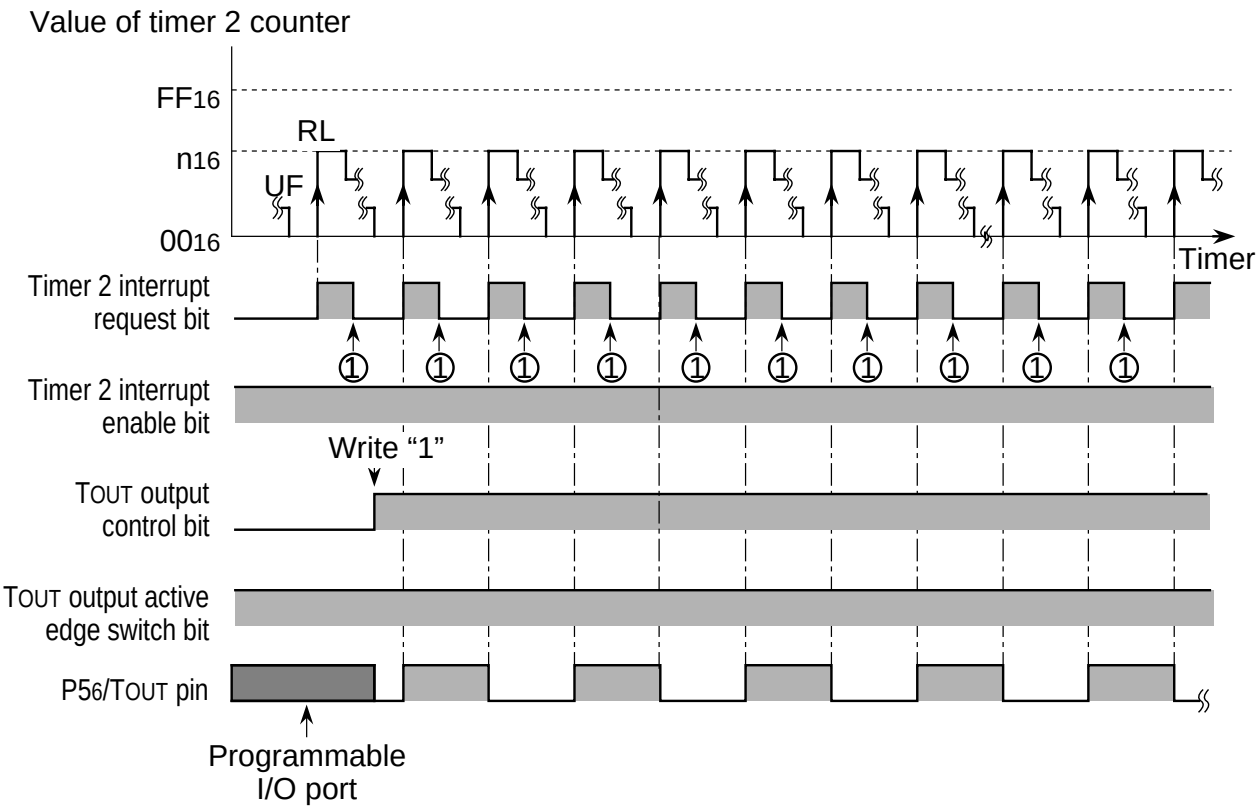
The timer 2 can output a pulse whose polarity is reversed at each the timer 2 counter underflow. Figure 2.4.4 shows a pulse output example.

From the moment that the TOUT output control bit is set to "1," pulses are output from the P56/TOUT output pin. The polarity is reversed every the timer 2 counter underflow.

To output pulses, set bit 6 of the port P5 direction register for the output mode by setting it to "1."

Pulse output example by timer 2

- UF : Underflow
- RL : Reload
- n : The timer 2 counter initial value



①: •Clearing by writing "0" to the timer 2 interrupt request bit.
•Clearing by accepting the timer 2 interrupt request when the timer 2 interrupt enable bit is "1."

Fig. 2.4.4 Pulse output example

2.4.2 Related registers

Figure 2.4.5 shows memory allocation of timer-related registers. Each of these registers is described below.

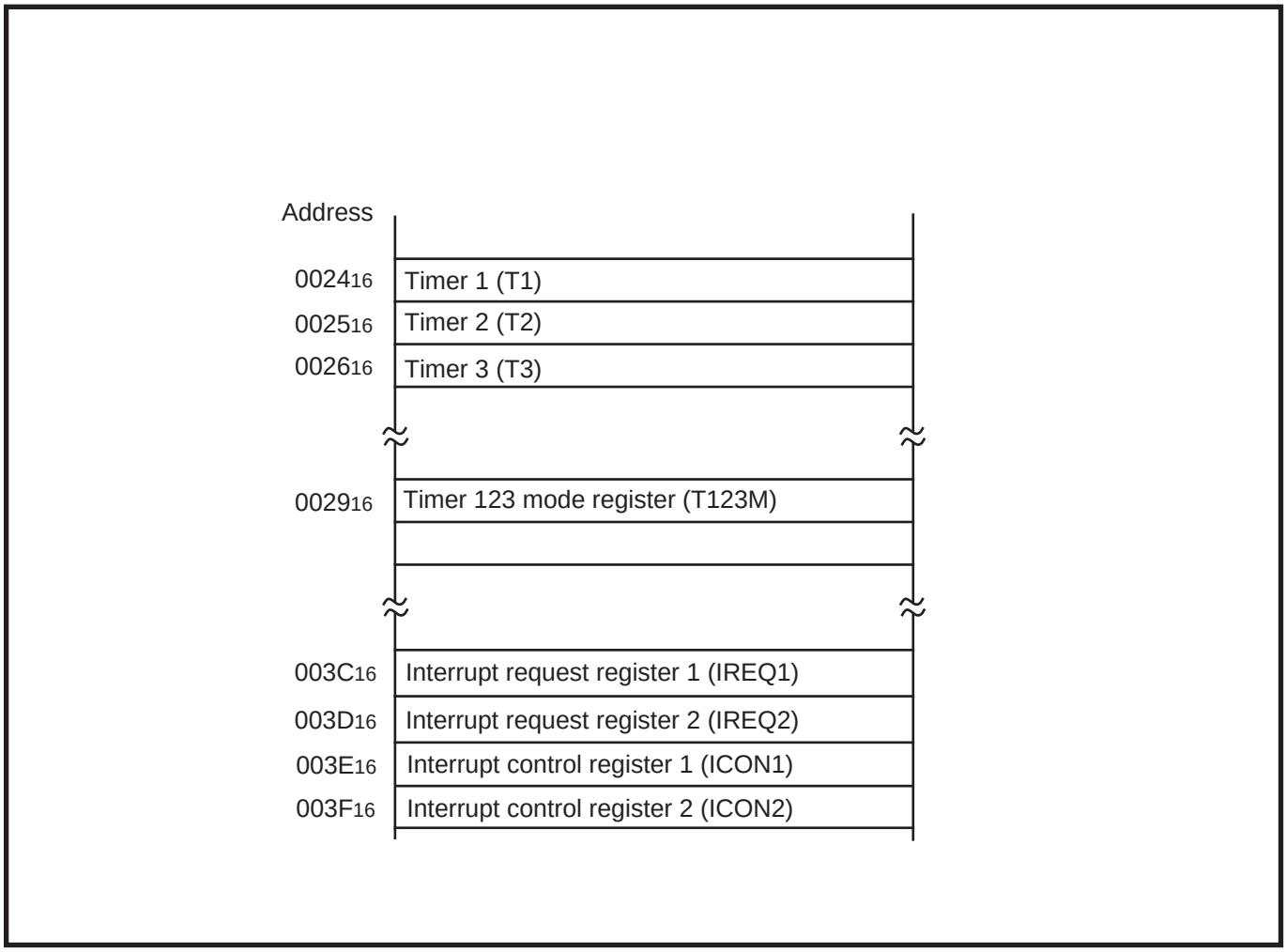


Fig. 2.4.5 Memory allocation of timer-related registers

2.4 Timer 1, timer 2, and timer 3

(1) Timer latches and timer counters (corresponding to timers 1 to 3)

The latches and the counters each consist of 8 bits and are allocated at the same address for each timer.

To access a latch and a timer, access the corresponding timer. When the timer is read out, the value of the counter (count value) is read out.

■Latch

The latch is a register which holds the value to be transferred (reloaded) automatically to the counter as the initial value of the counter at the counter underflow. It is impossible to read out the value of the latch. Figure 2.4.6 the structure of the latches.

For the rewrite operation of the value of the latch, refer to **"2.4.1 Explanation of operations, (2) Rewriting the value of the counter and the latch."**

●Timer 1 latch and timer 3 latch

Timer 1 and timer 3

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1 and timer 3 (T1,T3) [Address 24_{16}, 26_{16}]

B	Functions	At reset	R	W
0 to 7	•Set "00_{16} to FF_{16}" as timers 1 or 3 count value. •The values of each timer are set to the respective latches and transferred automatically to the respective counters at the counter underflow.	1	×	○

●Timer 2 latch

Timer 2

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2 (T2) [Address 25_{16}]

B	Functions	At reset	R	W
0	•Set "00_{16} to FF_{16}" as timer 2 count value. •The value of timer 2 is set to the timer 2 latch and transferred automatically to the timer 2 counter at the timer 2 underflow.	1	×	○
1 to 7		0	×	○

Fig. 2.4.6 Structure of latches

■Counters

The counters count the count source*1. Figure 2.4.7 shows the structure of the timer counters.
The value of the counter is decremented by 1 each time a count source is input.
The division ratio of the counters is represented by the following expression.

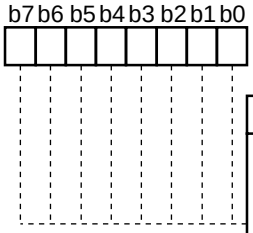
$$\text{Division ratio of the counter} = \frac{1}{\text{the counter initial value} + 1}$$

When the timer is read out, the value of the counter (count value) is read out.
For the rewriting operation for the value of the counter, refer to **"2.4.1 Explanation of operations, (2) Rewriting the value of the counter and the latch."**

*1: For count source selection, refer to **"2.4.2 Related registers, (2) Timer 123 mode register."**

●Timer 1 counter and timer 3 counter

Timer 1 and timer 3

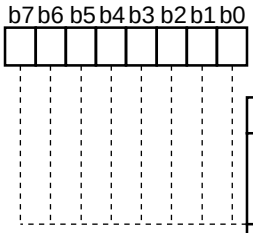


Timer 1 and timer 3 (T1, T3) [Address 24_{16}, 26_{16}]

B	Functions	At reset	R	W
0 to 7	•Set "00_{16} to FF_{16}" as timers 1 or 3 count value. •The value of the counter is decremented by 1 each time a count source is input. •The values of each timer are set to the respective counters. •The respective count values are read out by reading each timer.	1	○	○

●Timer 2 counter

Timer 2



Timer 2 (T2) [Address 25_{16}]

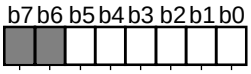
B	Functions	At reset	R	W
0	•Set "00_{16} to FF_{16}" as timer 2 count value. •The value of the counter is decremented by 1 each time a count source is input.	1	○	○
1 to 7	•When timer 2 write control bit is "0," the value of the timer 2 is set to the timer 2 counter. •The timer 2 count value is read out by reading the timer 2.	0	○	○

Fig. 2.4.7 Structure of timer counters

(2) Timer 123 Mode Register (T123M)

The timer 123 mode register (address 002916) consists of TOUT output control bit, the count source selection bits, and others. Figure 2.4.8 shows the structure of the timer 123 mode register. Each bit is described below.

Timer 123 mode register



Timer 123 mode register (T123M) [Address 29 16]

B	Name	Functions	At reset	R	W
0	Timer 2 TOUT output active edge switch bit	0 : Start at "H" output 1 : Start at "L" output	0	○	○
1	Timer 2 TOUT output control bit	0 : TOUT output disabled 1 : TOUT output enabled	0	○	○
2	Timer 2 write control bit	0 : Write data in latch and counter 1 : Write data in latch only	0	○	○
3	Timer 2 count source selection bit	0 : Timer 1 underflow 1 : f(XIN)/16 (Middle-/high-speed mode) f(XCIN)/16 (Low-speed mode) (Note)	0	○	○
4	Timer 3 count source selection bit	0 : Timer 1 underflow 1 : f(XIN)/16 (Middle-/high-speed mode) f(XCIN)/16 (Low-speed mode) (Note)	0	○	○
5	Timer 1 count source selection bit	0 : f(XIN)/16 (Middle-/high-speed mode) f(XCIN)/16 (Low-speed mode) (Note) 1 : f(XCIN)	0	○	○
6, 7	Nothing is allocated. These bits cannot be written to and are fixed to "0" at reading.		0	0	×

Note: Internal clock ϕ is f(XCIN)/2 in the low-speed mode.

Fig. 2.4.8 Structure of timer 123 mode register

■Timer 2 TOUT output active edge switch bit (bit 0)
The TOUT output active edge switch bit selects an initial level of the TOUT output.
When bit 0 is "0," the output pulse from the P56/TOUT pin is started at the "H" level.
When bit 0 is "1," the output pulse from the P56/TOUT pin is started at the "L" level.

■Timer 2 TOUT output control bit (bit 1)
The TOUT output control bit controls the TOUT output.
When bit 1 is "0," the TOUT output is disabled.
When bit 1 is "1," the TOUT output is enabled.

■Timer 2 write control bit (bit 2)
The timer 2 write control bit controls writing to the timer 2.
When bit 2 is "0," a simultaneous write operation to both the timer 2 latch and the timer 2 counter is set.
When a value is written to the timer 2, the value is set into both the timer 2 latch and the timer 2 counter at the same time.
When bit 2 is "1," a write operation to the latch only is set.
When a value is written into the timer 2, the value is set into the timer 2 latch only.
When a value is written into the timer 2 latch only, this rewritten value is transferred to the timer 2 counter at the first timer 2 counter underflow after rewriting.

■Timer 2 count source selection bit (bit 3)
The timer 2 count source selection bit selects a count source of the timer 2. Table 2.4.1 shows the relation between the timer 2 count source selection bit and count sources.

Table 2.4.1 Relation between timer 2 count source selection bit and count sources

bit 3	Timer 2 count source
0	Timer 1 underflow
1	f(XIN)/16 (In low speed mode; f(XCIN)/16)

■Timer 3 count source selection bit (bit 4)
The timer 3 count source selection bit selects a count source of the timer 3. Table 2.4.2 shows the relation between the timer 3 count source selection bit and count sources.

Table 2.4.2 Relation between timer 3 count source selection bit and count sources

bit 4	Timer 3 count source
0	Timer 1 underflow
1	f(XIN)/16 (In low speed mode; f(XCIN)/16)

■Timer 1 count source selection bit (bit 5)
The timer 1 count source selection bit selects a count source of the timer 1. Table 2.4.3 shows the relation between the timer 1 count source selection bit and count sources.

Table 2.4.3 Relation between timer 1 count source selection bit and count sources

bit 5	Timer 1 count source	Count source examples	
		f(XIN) = 8 MHz	f(XCIN) = 32.768 kHz
0	f(XIN)/16 (In low speed mode; f(XCIN)/16)	500 kHz	2.048 kHz
1	f(XCIN)	—	32.768 kHz

(3) Interrupt request register 1 (IREQ1) and interrupt request register 2 (IREQ2)

The interrupt request register 1 (address 003C16) and the interrupt request register 2 (address 003D16) indicate whether an interrupt request has occured or not.

Figure 2.4.9 shows the structure of the interrupt request register 1 and Figure 2.4.10 shows the structure of the interrupt request register 2.

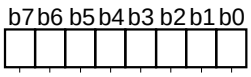
The occurrence of an interrupt request causes the corresponding bit to be set to "1." This interrupt request bit is automatically cleared to "0" by the acceptance of the interrupt request.

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

The occurrence of each interrupt is controlled by the interrupt enable bit (refer to the next item).

For details of interrupts, refer to **"2.2 Interrupts."**

Interrupt request register 1



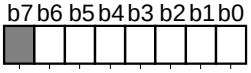
Interrupt request register 1 (IREQ1) [Address 3C 16]

B	Name	Functions	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT_1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

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

Fig. 2.4.9 Structure of interrupt request register 1

Interrupt request register 2



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

B	Name	Functions	At reset	R	W
0	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key input interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	ADT/A-D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated. This bit cannot be written to and is fixed to “0” at reading.		0	0	×

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

Fig. 2.4.10 Structure of interrupt request register 2

(4) Interrupt control register 1 (ICON1) and interrupt control register 2 (ICON2)

The interrupt control register 1 (address 003E16) and the interruot contorol register 2 (address 003F16) control each interrupt request source.

Figure 2.4.11 shows the structure of the interrupt control register 1 and Figure 2.4.12 shows the structure of the interrupt control register 2.

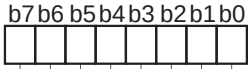
When an interrupt enable bit is "0," the corresponding interrupt request is disabled. If an interrupt request occurs when this bit is "0," the corresponding interrupt request bit only is set to "1," and the interrupt request is not accepted.

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

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

For details of interrupts, refer to **"2.2 Interrupts."**

Interrupt control register 1

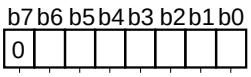


Interrupt control register 1 (ICON1) [Address 3E16]

B	Name	Functions	At reset	R	W
0	INT0 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	INT1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	Timer 2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Timer 3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○

Fig. 2.4.11 Structure of interrupt control register 1

Interrupt control register 2



Interrupt control register 2 (ICON2) [Address 3F 16]

B	Name	Functions	At reset	R	W
0	$CNTR_0$ interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	$CNTR_1$ interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Timer 1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	INT_2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Key input interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	ADT/A-D conversion interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Fix this bit to “0.”		0	0	0

Fig. 2.4.12 Structure of interrupt control register 2

2.4.3 Register setting example

Figure 2.4.13 shows an example of setting registers for timers 1, 2, and 3.

[Notes on use]

Notes 1: For using interrupt processing, set the following :
- •Before setting ① below, clear the respective timer interrupt enable bits and the timer respective interrupt request bits to "0."
- •After setting ④ below, set the respective timer interrupt enable bits to "1" (interrupts enabled)

2: The values written in the timers 1 and 3 are set into both the respective latches and the respective counters at the same time.

3: To enable the TOUT output when the timer 2 is used, set port P56 (this port is also used the TOUT pin) for the output mode.

4: Write values in the order of the timer 1, timer 2, and timer 3.

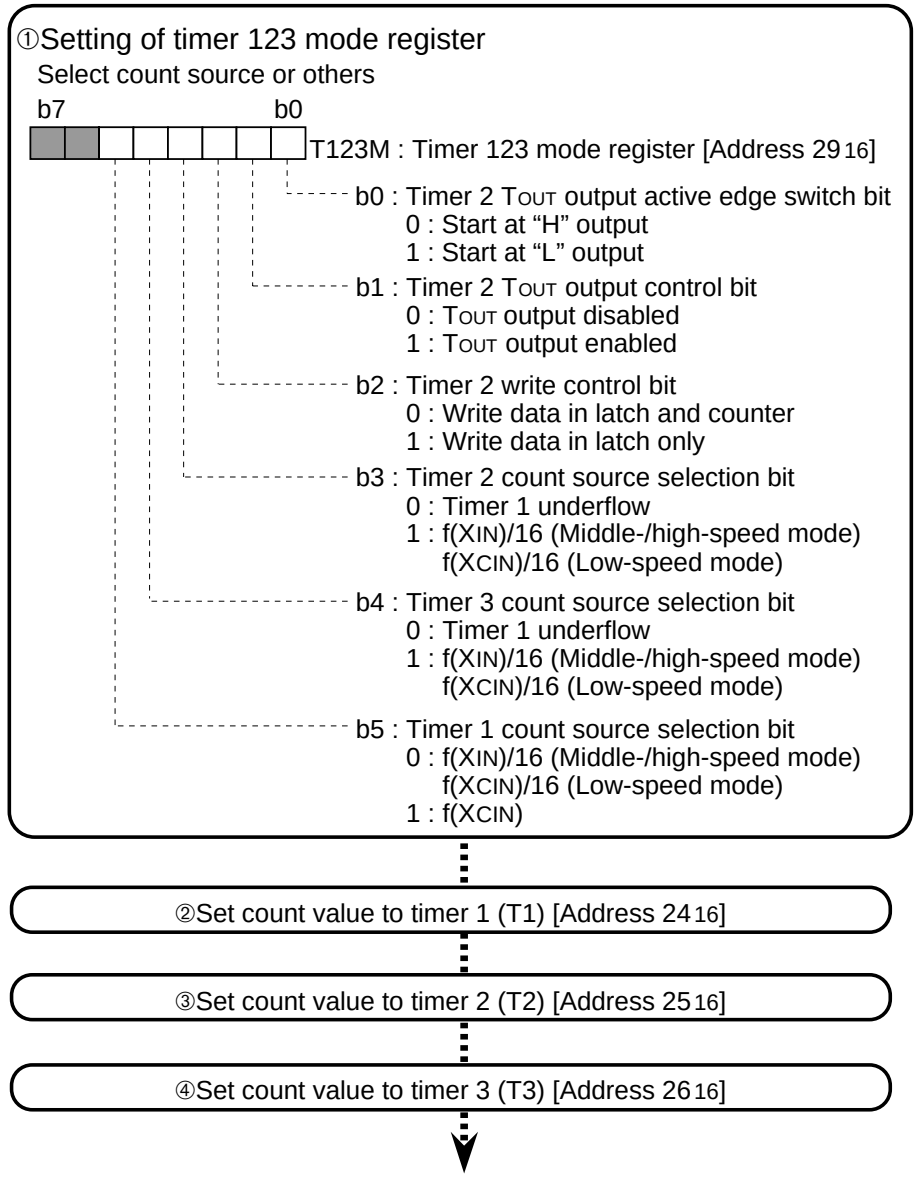


Fig. 2.4.13 Example of setting registers for timers 1, 2, and 3

2.4.4 Application example

Timer mode: Clock function (measurement of one second)

Outline: The input clock is divided by timer, with a timer 1 interrupt caused every 0.4 ms, 1 second is counted. Thus, the clock is counted up every second.

Specification: •Division of f(XCIN) = 32 kHz by timer 1 causes an interrupt.
•The counter value counted by the timer 1 interrupt is checked in the main routine. If 1 second has elapsed, the clock counts up.

Figure 2.4.14 shows setting of the related registers and Figure 2.4.15 shows the control procedure.

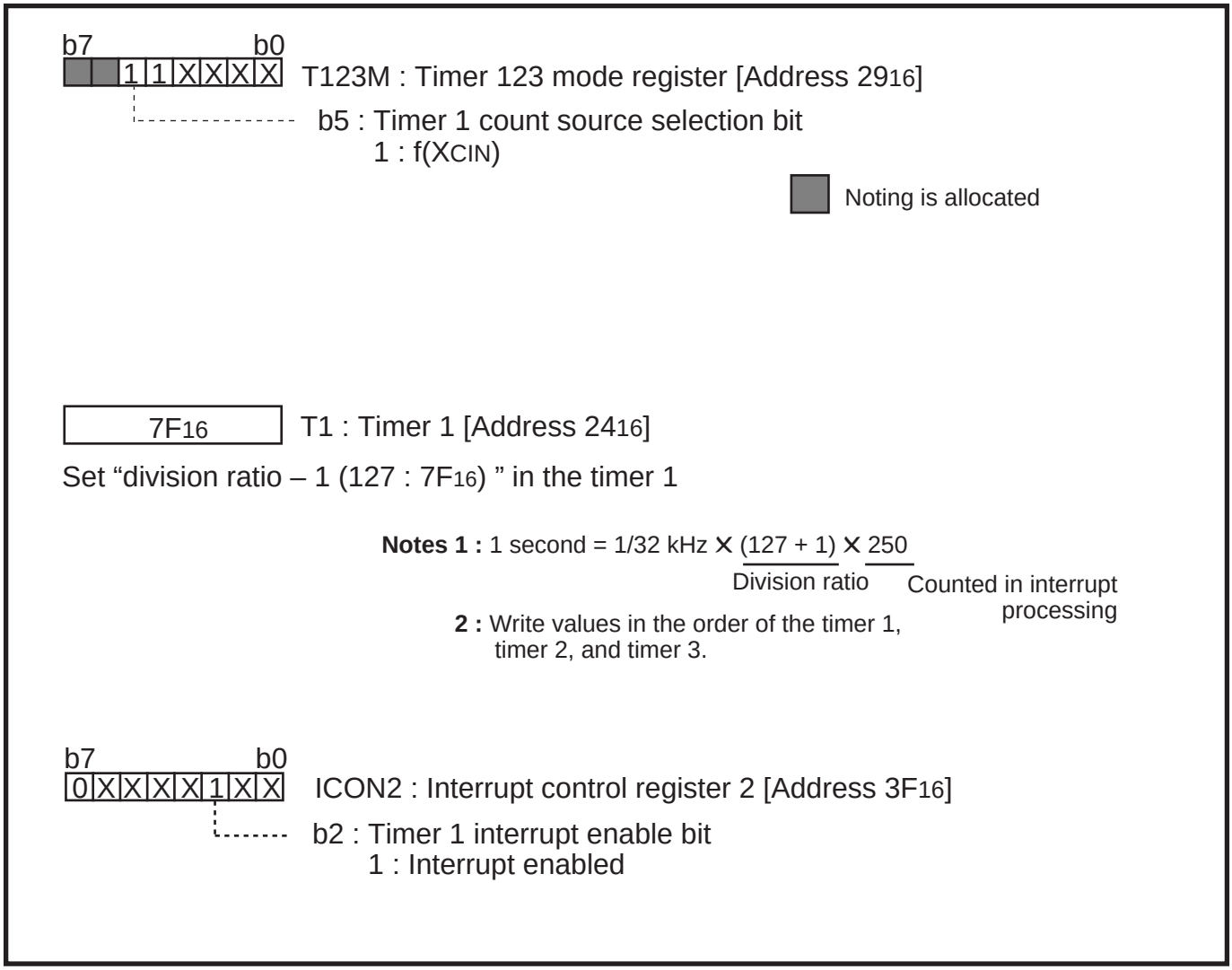


Fig. 2.4.14 Setting of related registers

2.4 Timer 1, timer 2, and timer 3

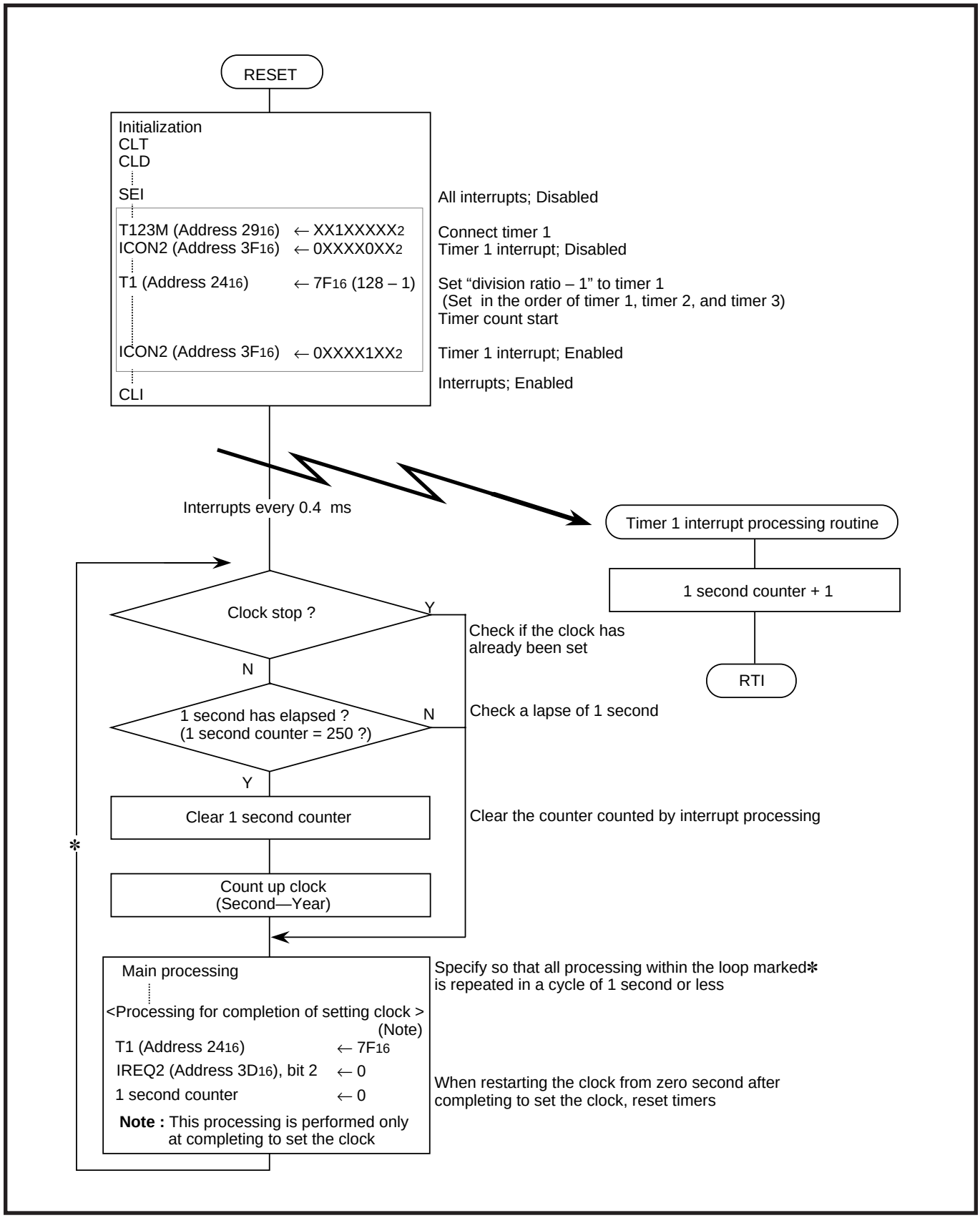


Fig. 2.4.15 Control procedure

2.4.5 Notes on use

(1) Notes on using timer 1 to timer 3

- When the count sources of timers 1 to 3 are switched, a short pulse occurs in counted input signals, so the timer count value may change greatly.
- When the timer 1 output is selected as a count source of timer 2 or timer 3, a short pulse occurs in the output signal at writing value into the timer 1, so the count value of the timer 2 or timer 3 may change greatly.
- For the above reasons, set values in the order of timer 1, timer 2, and timer 3 after setting their count sources.

(2) Timer 2 write control

When writing to the latch only is selected, the value written into the timer 2 T2 (address 0025_{16}) is written only in the latch for reloading. This rewritten value is transferred to the timer 2 counter at the first underflow after rewriting.

Usually, a value is written in both the latch and the counter at the same time. That is, when a value is written to timer, it is set in both the latch and the counter.

(3) Timer 2 output control

In the timer 2 (TOUT) output enable state, a signal whose polarity is reversed each time the timer 2 counter underflows is output from the TOUT pin. In this case, set the port $P5_6$ (this is used as the TOUT pin) for the output mode.

2.5 Serial I/O

2.5.1 Explanation of operations

As a serial I/O, it is possible to select either the clock synchronous serial I/O mode or the clock asynchronous serial I/O (UART) mode. This section describes operations in both the clock synchronous mode and the clock asynchronous (UART) mode. When serial I/O is actually used, refer to **"2.5.4 Register setting example."**

(1) Clock synchronous serial I/O mode

In the clock synchronous mode, 8 shift clocks generated in the clock control circuit are used as synchronizing clocks for transfer. In synchronization with these shift clocks, the transmit operation on the transmitter and the receive operation on the receiver are simultaneously executed.

The transmitter transmits each 1-bit data from the P45/TxD pin in synchronization with the falling of the shift clocks.

The receiver receives each 1-bit data from the P44/RxD pin in synchronization with the rising of the shift clocks.

Figure 2.5.1 shows an external connection example in the clock synchronous mode.

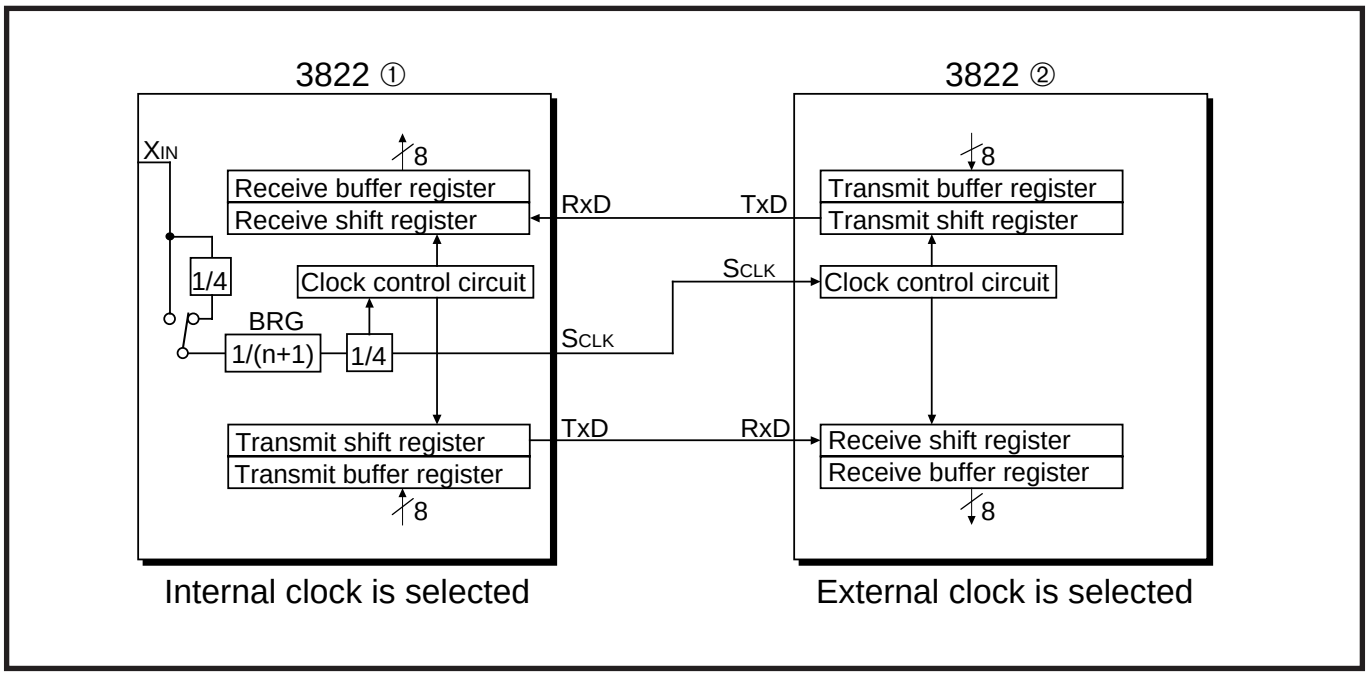


Fig. 2.5.1 External connection example in clock synchronous mode

■Shift clock

Ordinarily, when clock synchronous transfer is performed between microcomputers, an internal clock is selected for one of them, and it outputs 8 shift clocks generated by a start of transmit operation from the P46/SCLK pin. An external clock is selected for the other microcomputer, and it uses the clock input from the P46/SCLK pin as a shift clock. Figure 2.5.2 shows a shift clock.

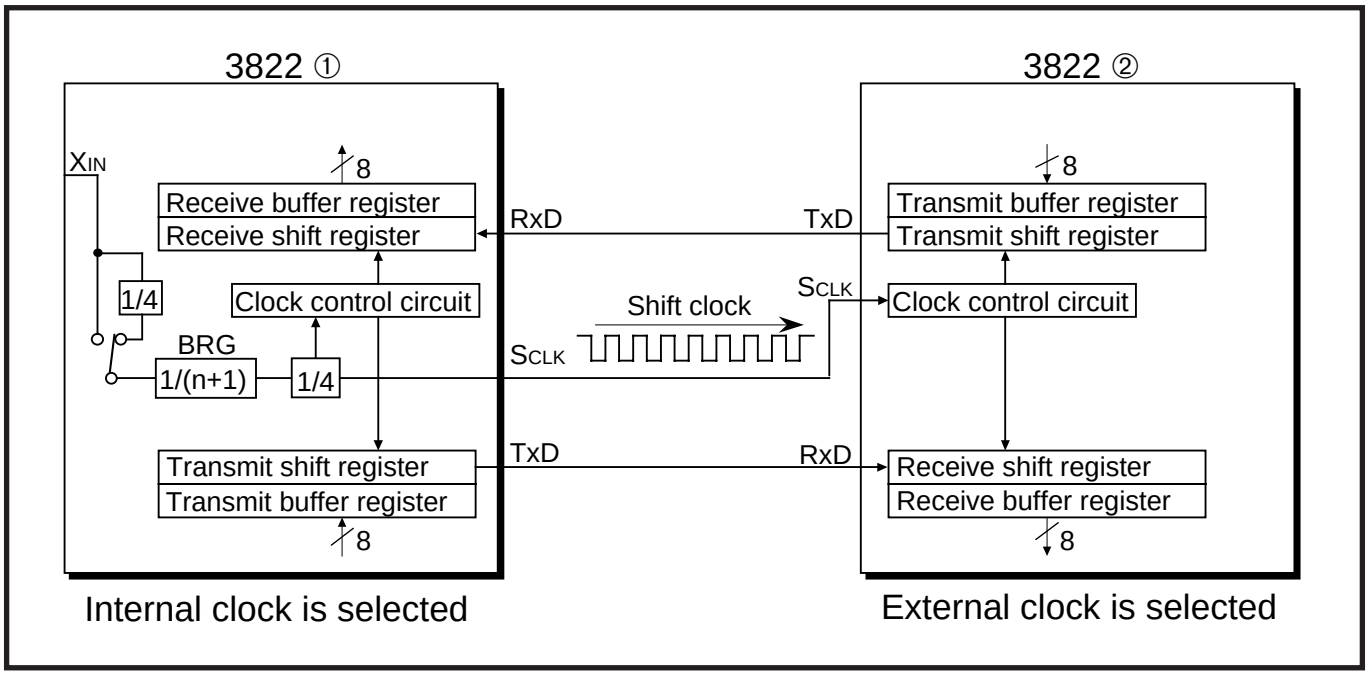


Fig. 2.5.2 Shift clock

■Data transfer rate (baud rate)

When an internal clock is used, the data transfer rate (baud rate), which is a shift clock frequency in the clock synchronous mode, is determined by baud rate generator (BRG). When the BRG count source selection bit (bit 0) of the serial I/O control register (address $001A_{16}$) is "0," X_{IN} pin input clock is input to the BRG, when this bit is "1," X_{IN} pin input clock divided by 4 is input to the BRG. The expression for baud rate is shown below.

●When selecting an internal clock (Using BRG)

$$\text{Baud rate [bps]} = \frac{X_{IN} \text{ pin input}}{\text{Division ratio } *1 \times (\text{BRG setting value } *2 + 1) \times 4}$$

✻1 Division ratio; Select "1," or "4"

✻2 BRG setting value; 0 to 255 (00_{16} to FF_{16})

●When selecting an external clock

$$\text{Baud rate [bps]} = \text{Frequency of input clock to } P4_6/S_{CLK} \text{ pin}$$

■Transmit operation in the clock synchronous mode

Transmit operation in the clock synchronous mode is described below.

●Start of transmit operation

A transmit operation is started by writing transmit data into the transmit buffer register (address 0018_{16}) in the transmit enable state.*1

●Transmit operation

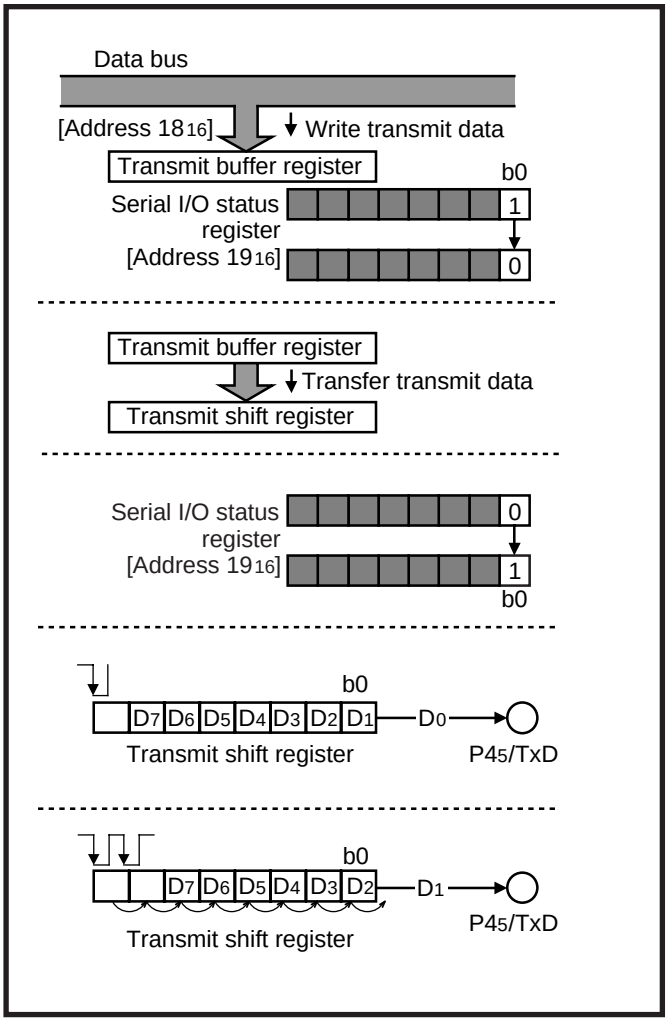
①By writing transmit data into the transmit buffer register, the transmit buffer empty flag (bit 0) of the serial I/O status register (address 0019_{16}) is cleared to "0."

②The transmit data written in the transmit buffer register is transferred to the transmit shift register.*2

③When a data transfer from the transmit buffer register to the transmit shift register is completed, the transmit buffer empty flag is set to "1."*3

④The transmit data transferred to the transmit shift register is output from the P45/TxD pin in synchronization with the falling of the shift clocks.

⑤The data is output from the least significant bit of the transmit shift register. Each time 1-bit data is output, the data of the transmit shift register is shifted by 1 bit toward the least significant bit.



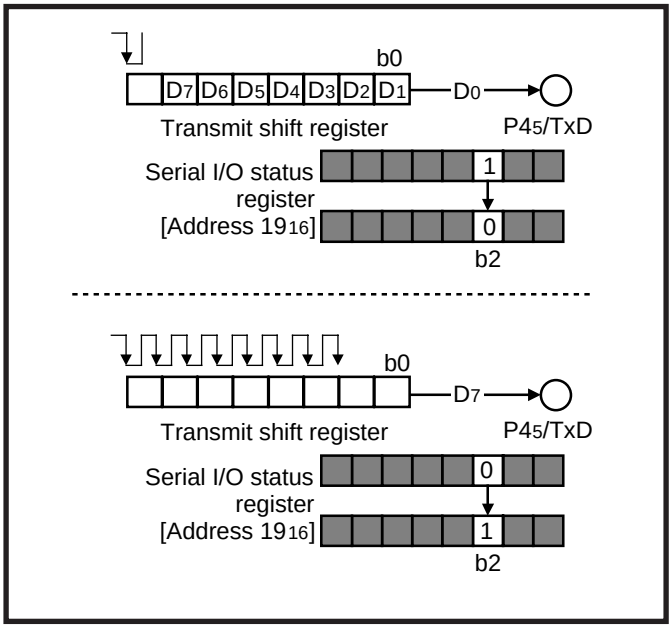
*1: Initialization of register or others for a transmit operation. Refer to **"2.5.4 Register setting example."**

*2: When the transmit interrupt source selection bit (bit 3) of the serial I/O control register (address $001A_{16}$) is set to "0," a serial I/O transmit interrupt request occurs immediately after transfer in ②. When this bit is set to "1," a transmit interrupt request occurs at the time of ⑦.

*3: While the transmit buffer empty flag is "1," it is possible to write the next transmit data into the transmit/receive buffer register.

⑥At the time when a transmit shift operation starts, the transmit shift register shift completion flag (bit 2) of the serial I/O status register is cleared to “0.” ✻4

⑦At the time when the transmit shift operation completes, the transmit shift register shift completion flag is set to “1.” ✻2 ✻4



✻4: When an internal clock is used as a synchronizing clock, supplying the shift clock to the transmit shift register stops automatically at the completion of 8-bit transmission. However, when the next transmit data is written to the transmit buffer register while the transmit shift register shift completion flag is “0,” supplying the shift clock is continued.

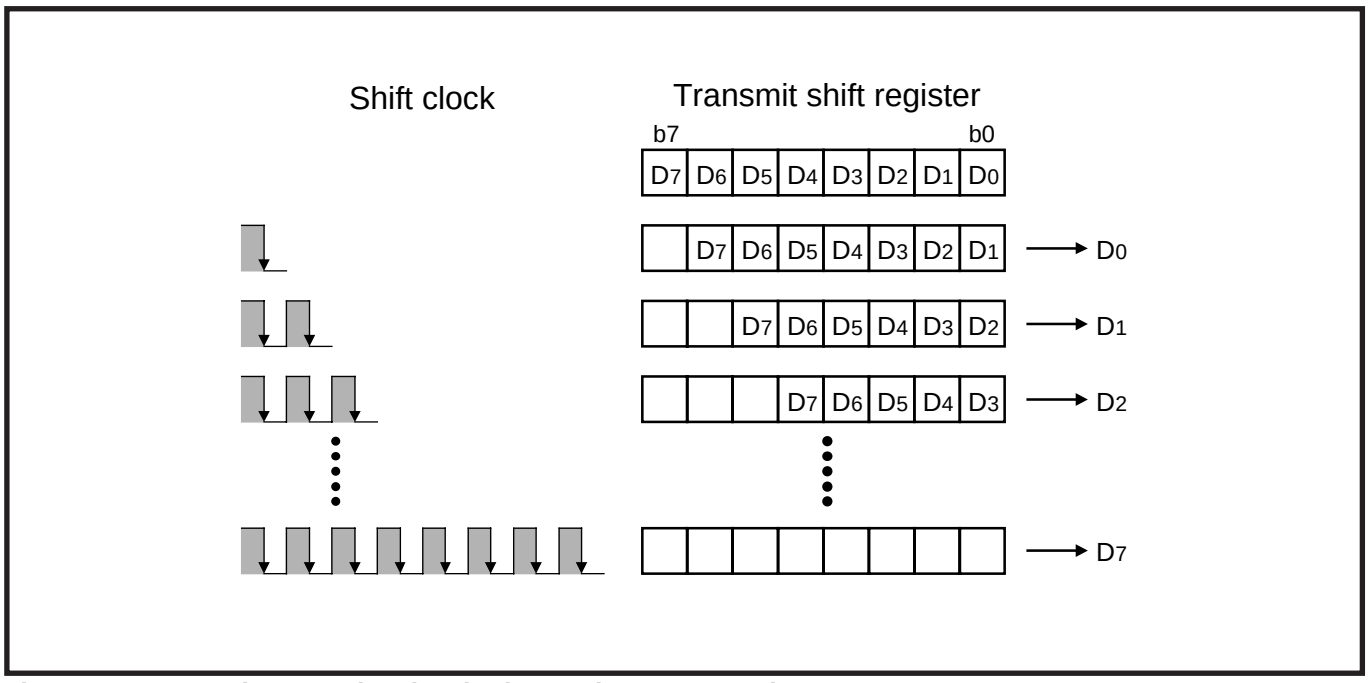


Fig. 2.5.3 Transmit operation in clock synchronous mode

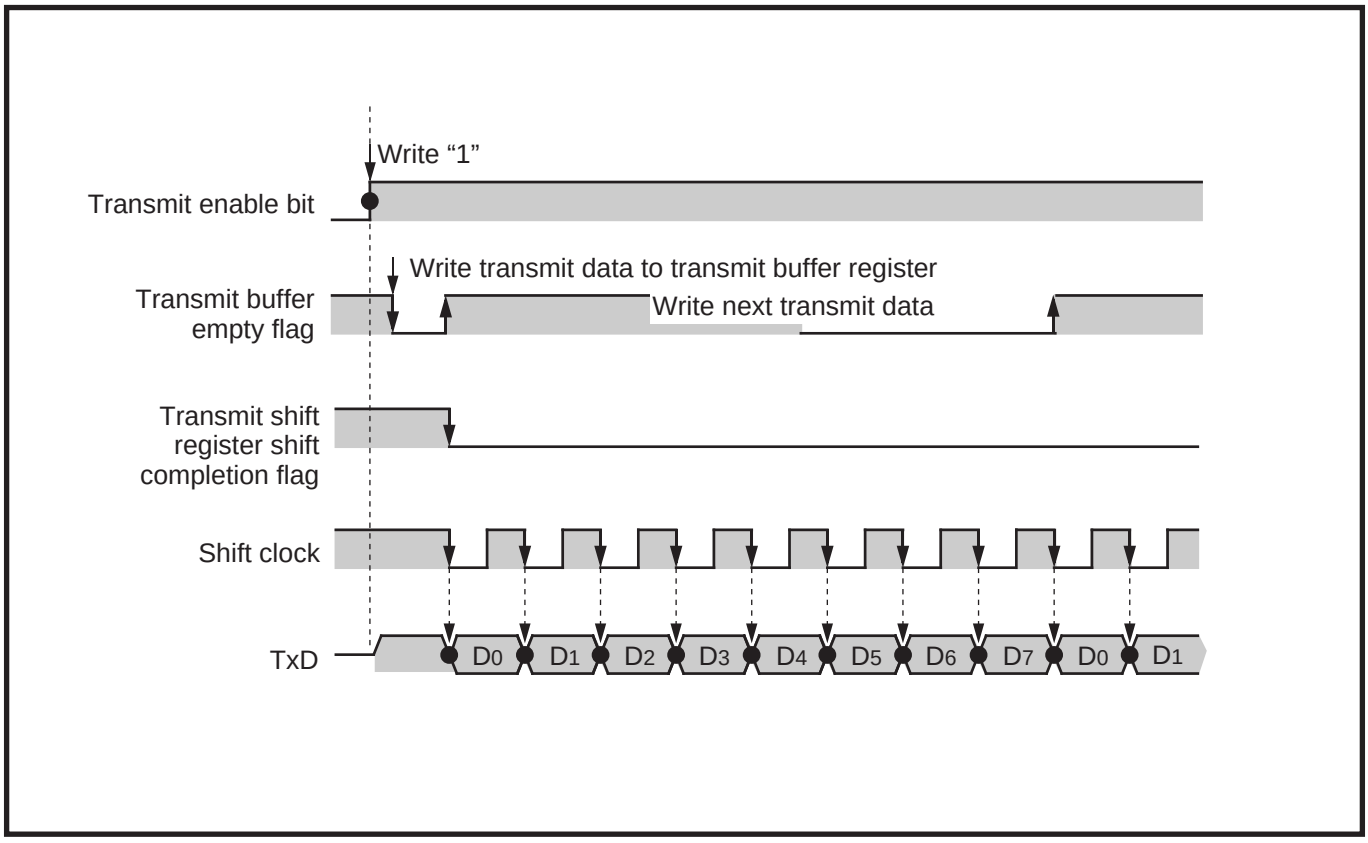


Fig. 2.5.4 Transmit timing example in clock synchronous mode

2.5 Serial I/O

■Receive operation in the clock synchronous mode

Receive operation in the clock synchronous mode is described below.

●Start of receive operation

A receive operation is started by writing data into the receive buffer register (address 0018_{16}) in the receive enable state.*1

- Transmit data in the full duplex data transfer mode
- Arbitrary dummy data in the half duplex data transfer mode

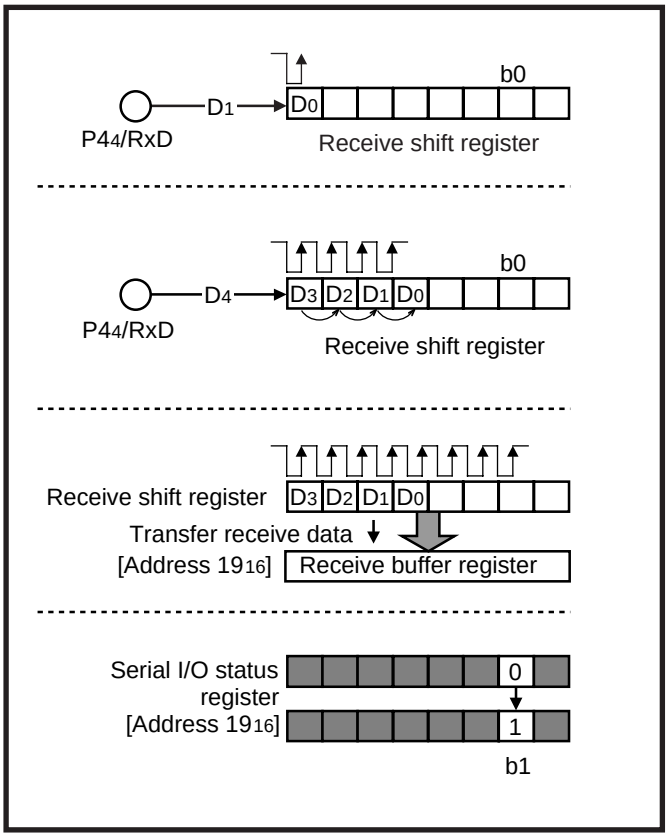
●Receive operation

①Each 1-bit data is read into the receive shift register from the P44/RxD pin in synchronization with the rising of the shift clocks.

②The data enters first into the most significant bit of the receive shift register. Each time 1-bit data is received, the data of the receive shift register is shifted by 1 bit toward the least significant bit.

③When 1-byte data has been input into the receive shift register, the data of the receive shift register is transferred to the receive buffer register (address 0018_{16}).*2

④When a data transfer to the receive buffer register is completed, the receive buffer full flag (bit 1) of the serial I/O status register (address 0019_{16}) is set to "1,"*3 a serial I/O interrupt request occurs.



✻1: Initialization of register or others for a receive operation. Refer to **"2.5.4 Register setting example."**

✻2: When data remains without reading out the data of the receive buffer register (the receive buffer full flag is "1") and yet all the receive data has been input to the receive shift register, the overrun error flag of the serial I/O status register is set to "1." At this time, the data of the receive shift register is not transferred to the receive buffer register, but the former data of the receive buffer register is held.

✻3: The receive buffer full flag is cleared to "0" by reading out the receive buffer register.

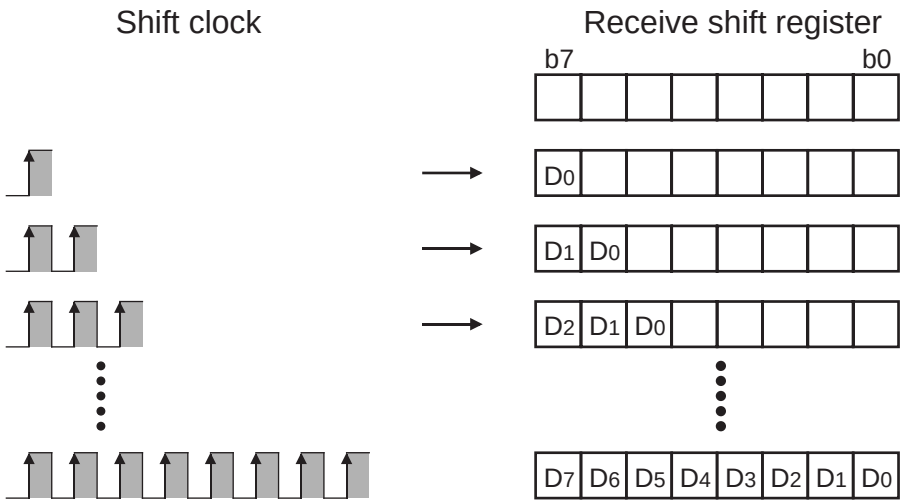


Fig. 2.5.5 Receive operation in clock synchronous mode

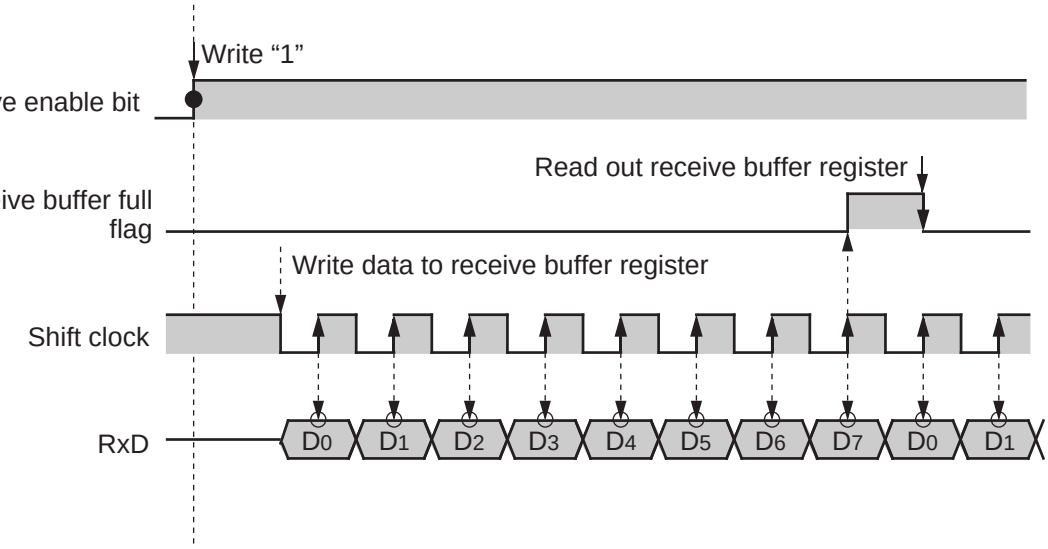


Fig. 2.5.6 Receive timing example in clock synchronous mode

■Transmit/receive timing example in the clock synchronous mode

Figure 2.5.7 shows a data transmit/receive timing example in the clock synchronous mode.

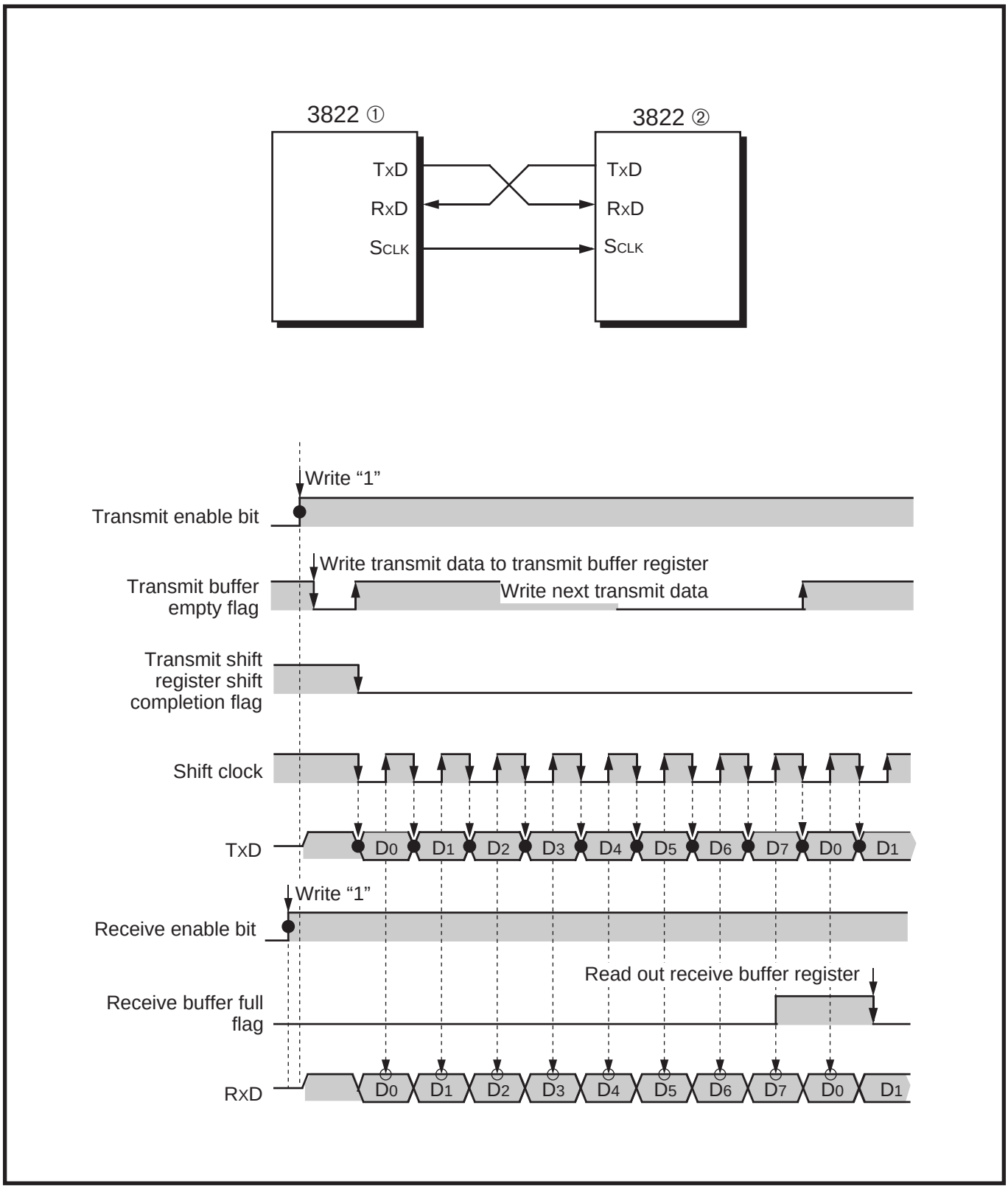


Fig. 2.5.7 Transmit/receive timing example in clock synchronous mode

(2) Clock asynchronous serial I/O (UART) mode

As the clock asynchronous mode (UART mode), data is transmitted and received in asynchronous form unifying the data transfer rate and the transfer data format between the transmitter and the receiver.

Figure 2.5.8 shows an external connection example in the UART mode.

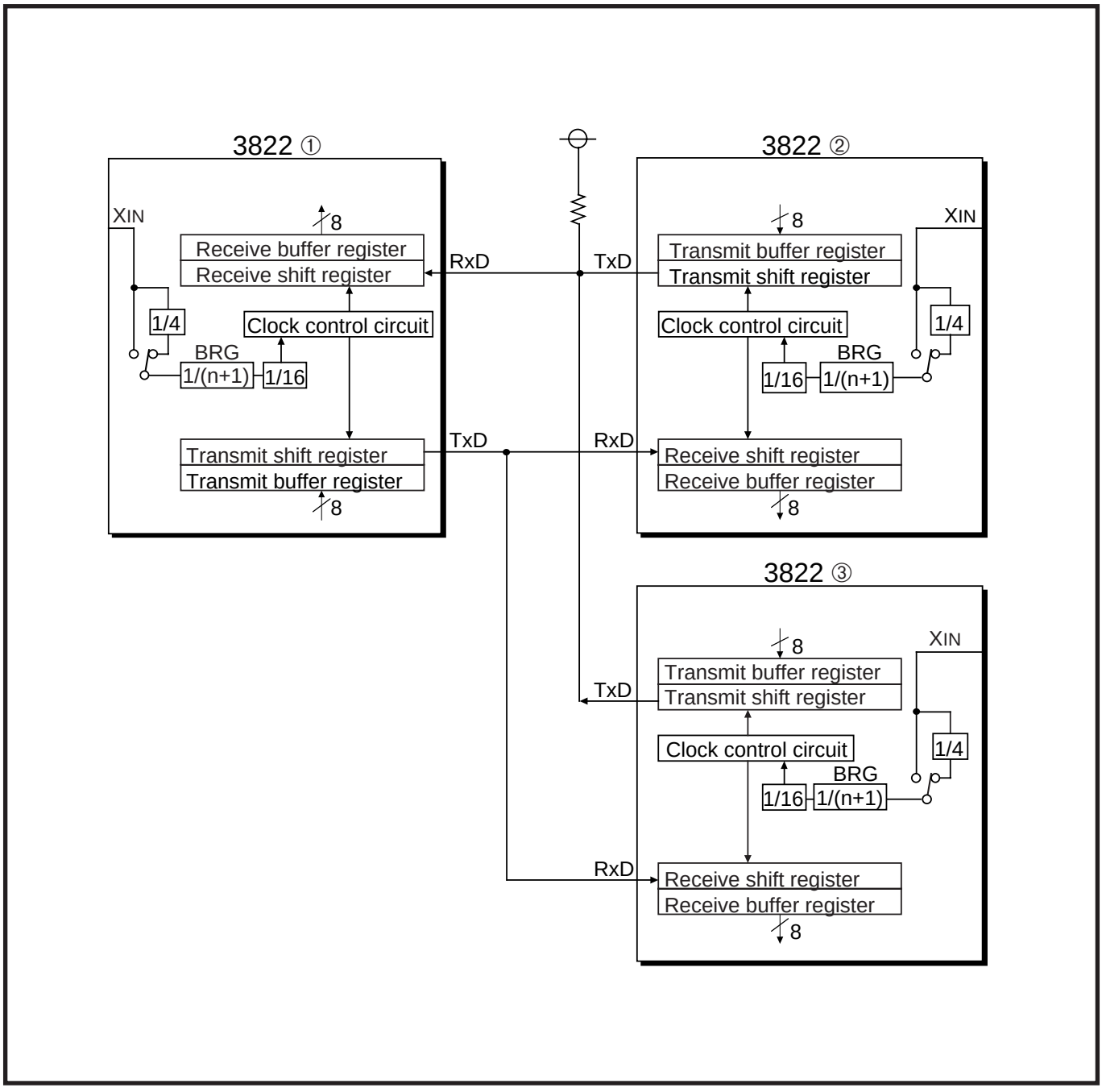


Fig. 2.5.8 External connection example in UART mode

2.5 Serial I/O

■Data transfer rate (baud rate)

When an internal clock is used, the data transfer rate (baud rate), which is a shift clock frequency in the UART mode, is determined by baud rate generator (BRG). When the BRG count source selection bit (bit 0) of the serial I/O control register (address $001A_{16}$) is "0," X_{IN} pin input clock is input to the BRG, when this bit is "0," X_{IN} pin input clock divided by 4 is input to the BRG. The expression for baud rate is shown below.

●When selecting an internal clock (Using BRG)

$$\text{Baud rate [bps]} = \frac{X_{IN} \text{ pin input}}{\text{Division ratio } *1 \times (\text{BRG setting value } *2 + 1) \times 16}$$

✻1 Division ratio; Select "1," or "4"

✻2 BRG setting value; 0 to 255 (00_{16} to FF_{16})

●When selecting an external clock

$$\text{Baud rate [bps]} = \frac{\text{Frequency of input clock to } P4_6/S_{CLK} \text{ pin}}{16}$$

Table 2.5.1 Baud rate selection table (reference values)

Baud rates [bps]		BRG count source	BRG setting value
At X_{IN} input = 4.9152 MHz	At X_{IN} input = 8 MHz		
300	488.28125	X_{IN} input/4	255 (FF_{16})
600	976.5625	X_{IN} input/4	127 ($7F_{16}$)
1200	1953.125	X_{IN} input/4	63 ($3F_{16}$)
2400	3906.25	X_{IN} input/4	31 ($1F_{16}$)
4800	7812.5	X_{IN} input/4	15 ($0F_{16}$)
9600	15625	X_{IN} input/4	7 (07_{16})
19200	31250	X_{IN} input/4	3 (03_{16})
38400	62500	X_{IN} input/4	1 (01_{16})
76800	125000	X_{IN} input/4	0 (00_{16})
153600	250000	X_{IN} input	1 (01_{16})
307200	500000	X_{IN} input	0 (00_{16})

■Transfer data format

Data transfer format is set by the UART control register (address $001B_{16}$). Figure 2.5.9 shows a transfer data format in the UART mode, Table 2.5.2, the each bit function of UART transmit data, Figure 2.5.10, all transfer data formats in the UART mode.

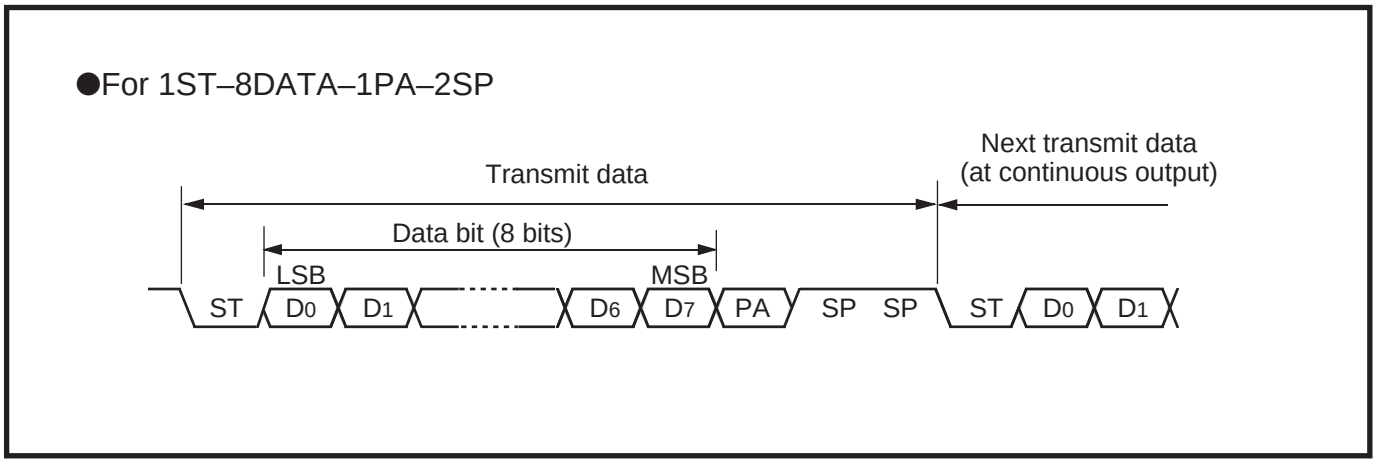


Fig. 2.5.9 Transfer data format in UART mode

Table 2.5.2 Each bit function of UART transmit data

Bit	Functions
ST (Start bit)	Indicates a start of data transmission. A "L" signal for one bit is added just before transmit data.
DATA (Data bit)	Indicates the transmit data written in the transmit buffer register, "0_2" data is a "L" signal and "1_2" data is a "H" signal. These bits are called as character bits.
PA (Parity bit)	To improve the reliability of data, this bit is added just after the last data bit. The value of this bit changes in accordance with the value of the parity selection bit so that the number of "1" in the transmit/receive data (including the parity bit) can always be an even or an odd number.
SP (Stop bit)	Indicates an completion of data transmission. This bit is added just after the last data bit (or just after a parity bit in the parity checking enabled). As a stop bit, a "H" signal for 1 bit or 2 bits is output.

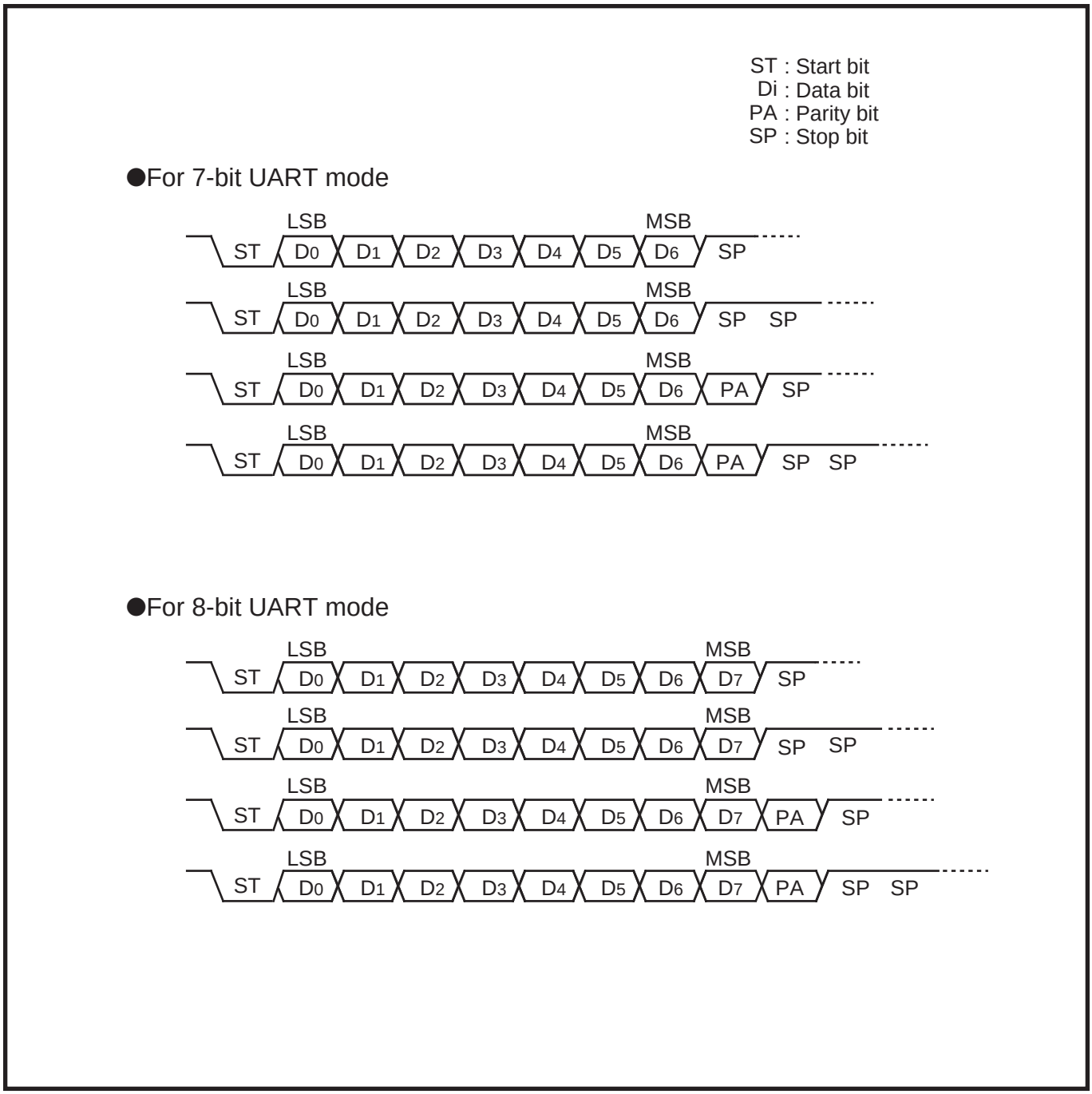


Fig. 2.5.10 All transfer data formats in UART mode

■Transmit operation in the UART mode

Transmit operation in the UART mode is described below.

●Start of transmit operation

A transmit operation is started by writing transmit data into the transmit buffer register (address 0018_{16}) in the transmit enable state.*1

●Transmit operation

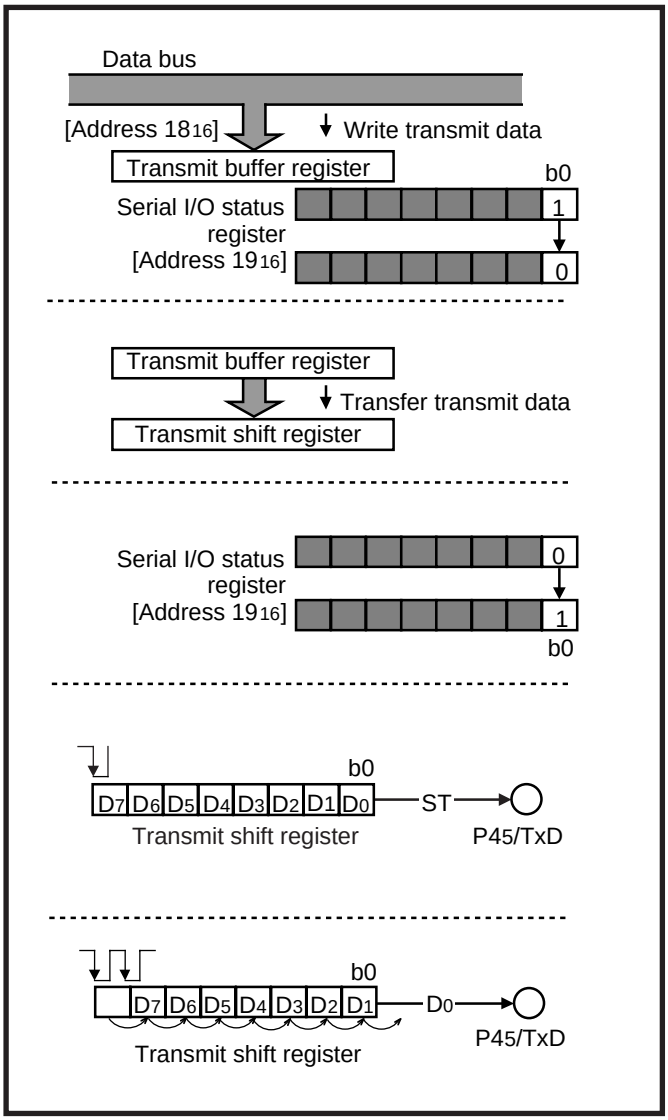
①By writing transmit data into the transmit buffer register, the transmit buffer empty flag (bit 0) of the serial I/O status register (address 0019_{16}) is cleared to "0."

②The transmit a data written in the transmit buffer register is transferred to the transmit shift register.*2

③When a data transfer from the transmit buffer register to the transmit shift register is completed, the transmit buffer empty flag is set to "1."*3

④The transmit data transferred to the transmit shift register is output from the P45/TxD pin in synchronization with the falling of the shift clock, beginning with the start bit. A start bit, a parity bit and a stop bit are automatically generated and output in accordance with the contents set in the UART control register.

⑤The data is output from the least significant bit of the transmit shift register. Each time 1-bit data is output, the data of the transmit shift register is shifted by 1 bit toward the least significant bit.



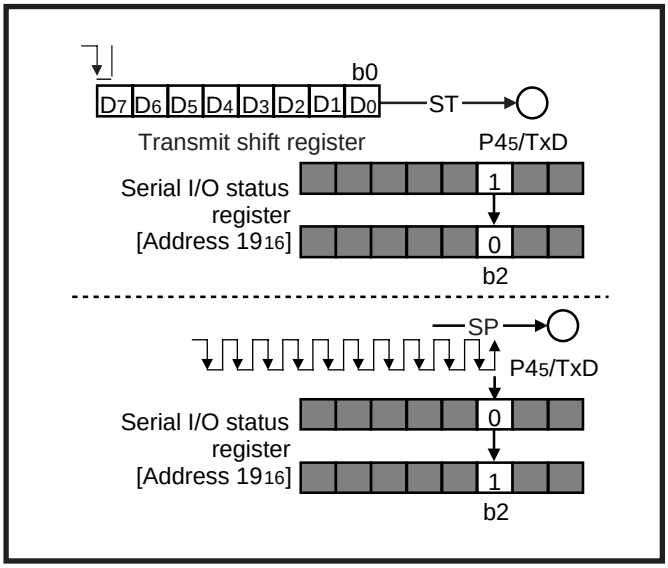
*1: Initialization of register or others for a transmit operation. Refer to **"2.5.4 Register setting example."**

*2: When the transmit interrupt source selection bit (bit 3) of the serial I/O control register (address $001A_{16}$) is set to "0," a serial I/O transmit interrupt request occurs immediately after transfer in ②. When this bit is set to "1," a transmit interrupt request occurs at the time of ⑦.

*3: While the transmit buffer empty flag is "1," it is possible to write the next transmit data into the transmit/receive buffer register.

⑥At the time when a transmit shift operation starts, the transmit shift register shift completion flag (bit 2) of the serial I/O status register is cleared to "0." ✻4

⑦After the lapse of a 1/2 period ✻5 of the shift clock from a transmission start of stop bit, the transmit shift register shift completion flag is set to "1." ✻2 ✻4



✻4: When an internal clock is used as a synchronizing clock, supplying the shift clock to the transmit shift register stops automatically at the completion of 8-bit transmission. However, when the next transmit data is written to the transmit buffer register while the transmit shift register shift completion flag is "0," supplying the shift clock is continued.

✻5: In the case of 2 stop bits, after the lapse of a 1/2 period of the shift clock from a start of the second stop bit transmission.

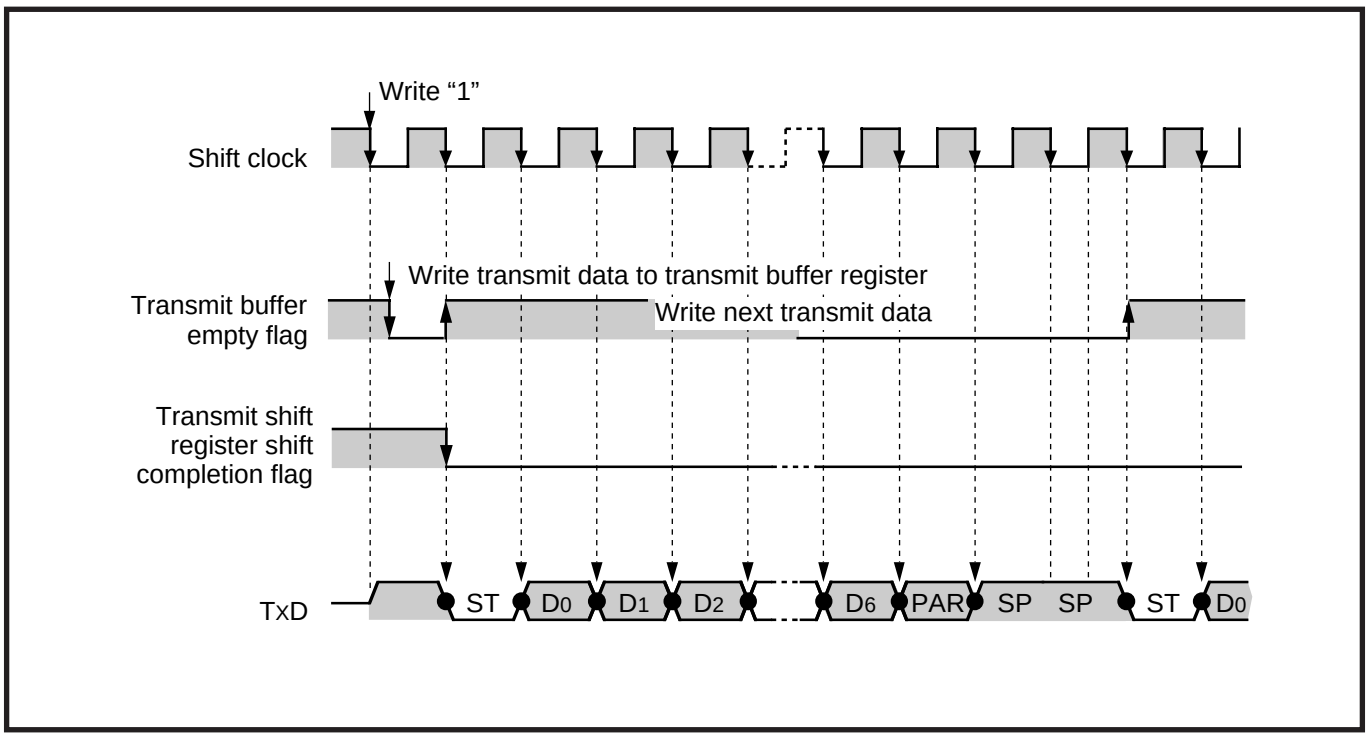


Fig. 2.5.11 Transmit timing example in UART mode

■Receive operation in the UART mode

Receive operation in the UART mode is described below.

●Start of receive operation

In the receive enable state,*1 set the receive enable bit (bit 5) of the serial I/O control register (address 001A16) into the enabled state ("1"). With this operation, a start bit is detected and a receive operation of serial data is started.

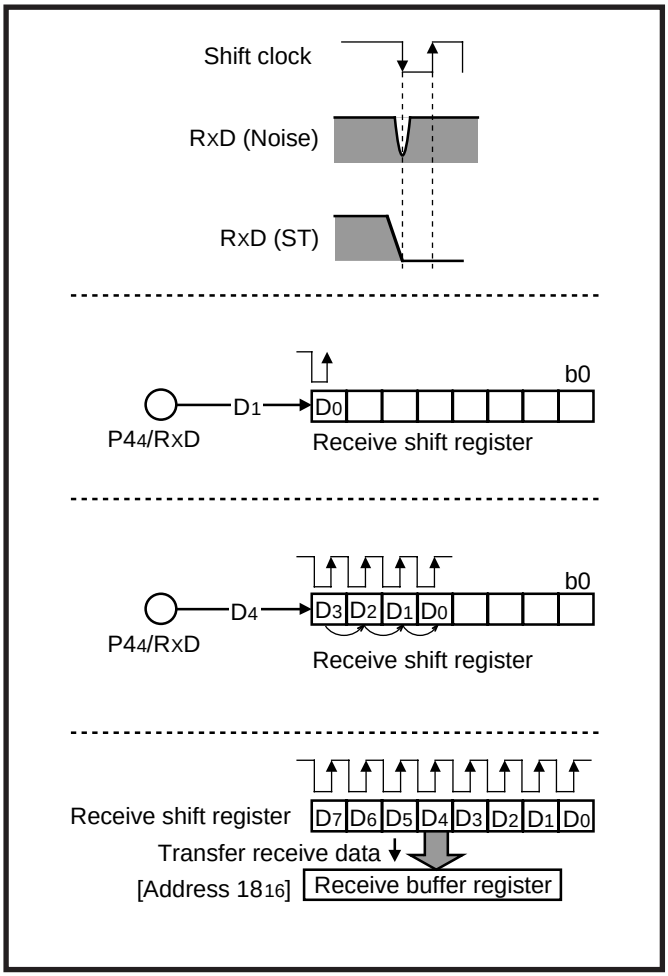
●Receive operation

①With the lapse of a 1/2 period of the shift clock from detection of the falling of the P44/RxD pin input, the P44/RxD pin level is checked. When it is "L" level, the bit is judged as a start bit.
When it is "H" level, the bit is judged as noise, so the receive operation is stopped, being put into wait status for a start bit again.

②Each 1-bit data is read into the receive shift register from the P44/RxD pin in synchronization with the rising of the shift clocks.

③The data after the detection of the start bit enters first into the most significant bit of the receive shift register. Each time 1-bit data is received, the data of the receive shift register is shifted by 1 bit toward the least significant bit.

④When a specified number of bits has been input into the receive shift register, the data of the receive shift register (address 001816) are transferred to the receive buffer register (address 001816).*2*3



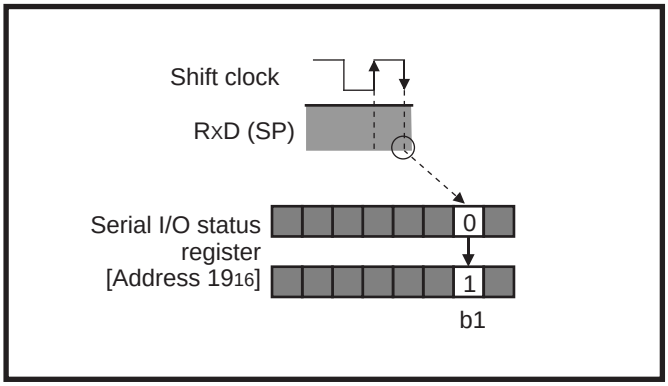
✻1: Initialization of register or others for a receive operation. Refer to **"2.5.4 Register setting example."**

✻2: When the data bit length is 7 bits, bits 0 to 6 of the receive buffer register are receive data, and bit 7 (MSB) is cleared to "0."

✻3: When data remains without reading out the data of the receive buffer register (the receive buffer full flag is "1") and yet all the receive data has been input to the receive shift register, the overrun error flag of the serial I/O status register is set to "1." At this time, the data of the receive shift register is not transferred to the receive buffer register, but the former data of the receive buffer register is held.

⑤After the lapse of a 1/2 period of the shift clock from a reception start of stop bit, the receive buffer full flag (bit 1) of the serial I/O status register is set to “1.” And a serial I/O receive interrupt request occurs.
⑥Error flag detection is performed concurrently with the occurrence of a serial I/O receive interrupt request.

✻4: The receive buffer full flag is cleared to “0” by reading out the receive buffer register.



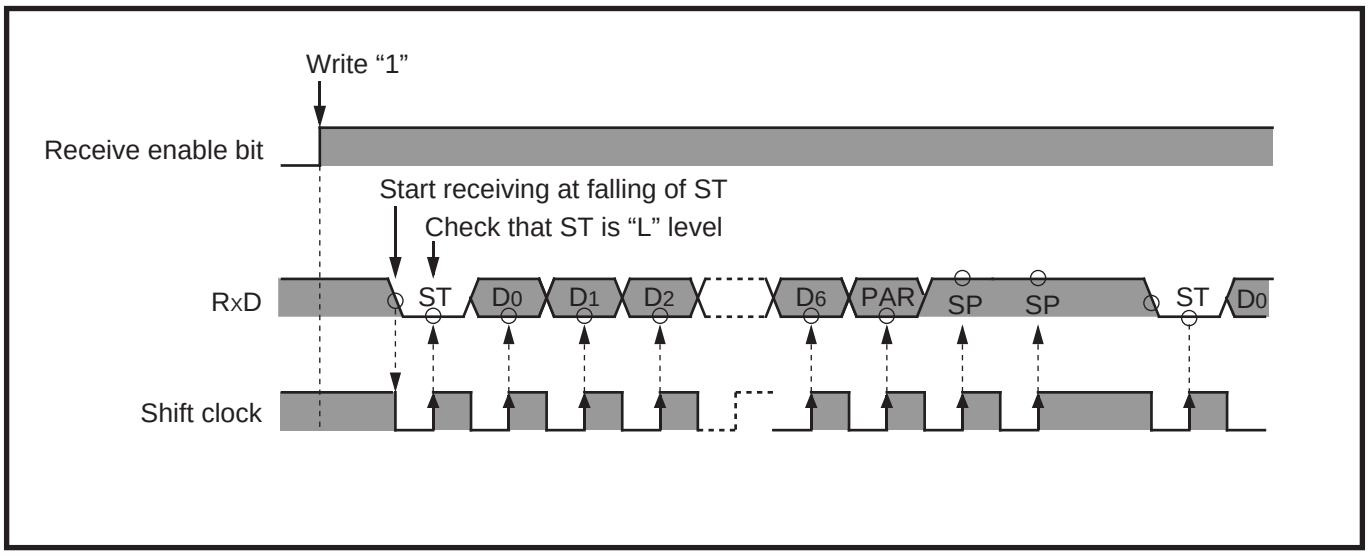


Fig. 2.5.12 Receive timing example in UART mode

(3) Processing upon occurrence of errors

■Parity error, framing error, or summing error

When a parity error, a framing error, or a summing error occurs, the flag corresponding to each error in the serial I/O status register is set to “1.” These flags are not cleared to “0” automatically, so set them to “0” by software.

These flags are set to “0” by one of the following operations.

•Set the receive enable bit to “0”
•Write data (arbitrary) into the serial I/O status register

■Overrun error

An overrun error occurs when data is already input in the receive buffer register and yet all data is input in the receive shift register.

If an overrun error occurs, the data of the receive shift register is not transferred and the data of the receive buffer register is held. At this time, even if the data of the receive buffer register is read out, the data of the receive shift register is not transferred.

Consequently, the data of the receive shift register becomes unreadable, so that the receive data becomes invalid.

If an overrun error occurs, after set the overrun error flag of the serial I/O status register to “0”, perform a receive operation again.

The overrun error flag is set to “0” by one of the following operations.

•Set the serial I/O enable bit to “0”
•Set the receive enable bit to “0”
•Write data (arbitrary) into the serial I/O status register

2.5.2 Pins

The serial I/O uses 4 pins, namely, pins for data transmit, data receive, shift clock transmit/receive, and receive enable signal output. All these pins are also used as port P4 and switched their functions by the serial I/O enable bit (bit 7) and $\overline{\text{SRDY}}$ output enable bit (bit 2) of the serial I/O control register (address $001A_{16}$).

The function of each pin is described below.

(1) Data transmit pin [TxD]

Outputs each bit of transmit data and is used as port $P4_5$.

When the serial I/O enable bit of the serial I/O control register is set to "1," this pin functions as a serial I/O data output pin.

(2) Data receive pin [RxD]

Inputs each bit of receive data and is used as port $P4_4$.

When the serial I/O enable bit of the serial I/O control register is set to "1," this pin functions as a serial I/O data input pin.

(3) Shift clock transmit/receive pin [SCLK]

■Clock synchronous mode

Inputs (receives from the outside) or outputs (supplies to the outside) a shift clock used for transmission and reception.

When the serial I/O synchronization clock selection bit (bit 1) of the serial I/O control register is set to "0" (use of internal clock), a shift clock is output to the outside. When this bit is set to "1" (use of external clock), a shift clock is input from the outside.

■UART mode

When the serial I/O synchronization clock selection bit (bit 1) of the serial I/O control register is set to "1" (use of external clock), a shift clock is supplied from the outside. When this bit is set to "0" (use of internal clock), this pin does not function.

(4) Receive enable signal output pin [$\overline{\text{SRDY}}$]

Notifies the outside of the receive enable state in the clock synchronous mode. This pin does not function in the UART mode.

•The $\overline{\text{SRDY}}$ output enable bit (bit 2) of the serial I/O control register is set to "1."

•The transmit enable bit (bit 4) of the serial I/O control register is set to "1."

When the above two conditions are satisfied, the pin level changes from "H" to "L" at the timing which data is written into the receive buffer register, notifying the outside of the receive enable state.

2.5.3 Related registers

Figure 2.5.13 shows the memory allocation of serial I/O-related registers. They are the transmit/receive buffer register, serial I/O status register, serial I/O control register, and UART control register.

(1) Transmit/receive buffer register (address 0018_{16})

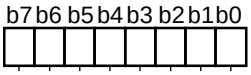
This register is used to write serial I/O transmit data or to read receive data (used for both the clock synchronous mode and the UART mode). For data transmission, transmit data is written into this register.

Received data is obtained by reading out this register.

Address	
0018_{16}	Transmit/receive buffer register (TB/RB)
0019_{16}	Serial I/O status register (SIOSTS)
$001A_{16}$	Serial I/O control register (SIOCON)
$001B_{16}$	UART control register (UARTCON)

Fig. 2.5.13 Memory allocation of serial I/O-related registers

Transmit/receive buffer register



Transmit/receive buffer register (TB/RB) [Address 18 16]

B	Functions	At reset	R	W
0 to 7	At transmit •Set "00_{16} to FF_{16}" as transmit data. •The transmit data is transferred automatically to transmit shift register by writing transmit data. At receive •When all receive data has been input into the receive shift register, the receive data is automatically transferred to this register.	?	○	○

Fig. 2.5.14 Structure of transmit/receive buffer register

(2) Serial I/O status register (address 0019$_{16}$)

This register consists of the following flags:

•flags representing the states of the registers used for transmission/reception

•error flags.

This is a read-only register.

Bit 7 is unused and set to “1” at reading.

Serial I/O status register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O status register (SIOSTS) [Address 19 $_{16}$]

B	Name	Functions	At reset	R	W
0	Transmit buffer empty flag (TBE)	0: Buffer full 1: Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0: Buffer empty 1: Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0: Transmit shift in progress 1: Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0: No error 1: Overrun error	0	○	×
4	Parity error flag (PE)	0: No error 1: Parity error	0	○	×
5	Framing error flag (FE)	0: No error 1: Framing error	0	○	×
6	Summing error flag (SE)	0: (OE) U (PE) U (FE) = 0 1: (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is allocated. This bit cannot be written to and is fixed to “1” at reading.		1	1	×

Fig. 2.5.15 Structure of serial I/O status register

■Transmit buffer empty flag (bit 0)

This bit is automatically cleared to “0” by writing transmit data into the transmit buffer register.

After the transmit data is written in the transmit buffer register, it is transferred to the transmit shift register. When this transfer is completed and the transmit buffer register becomes empty, this flag is automatically is set to “1.”

It is possible to write transmit data into the transmit buffer register only while the transmit buffer empty flag is “1.”

This flag is valid in both the clock synchronous mode and the UART mode.

■Receive buffer full flag (bit 1)

When all receive data has been input to the receive shift register and then this receive data is transferred to the receive buffer register, this flag is automatically is set to “1.”

When the transferred receive data is read out from the receive buffer register, the flag is automatically is cleared to “0.”

If all the next receive data is input to the receive shift register when the receive buffer flag is “1” (the receive buffer register is not yet read out), the overrun error flag is set to “1.”

This flag is valid in both the clock synchronous mode and the UART mode.

■Transmit shift register shift completion flag (bit 2)

When a shift operation (transmission of the first data bit) is started by shift clock after transmit data is transferred to the transmit shift register, this flag is cleared to “0.” When the shift operation is completed (completion of transmission of the last data bit), the flag is set to “1.”

This flag is valid in both the clock synchronous mode and the UART mode.

■Overrun error flag (bit 3)

If all the next receive data is input to the receive shift register when data has been input (not read out) in the receive buffer register, this flag is set to “1” (occurrence of an overrun error). This flag is set to “0” by one of the following operations.

•Set the serial I/O enable bit to “0”
•Set the receive enable bit to “0”
•Write data (arbitrary) into the serial I/O status register

This flag is valid in both the synchronous mode and the UART mode.

■Parity error flag (bit 4)

In the UART mode, this flag checks an even parity or odd parity by hardware.

When the parity of received data is different from the set parity, this flag is set to “1.”

This flag is set to “0” by one of the following operations.

•Set the receive enable bit to “0”
•Write data (arbitrary) into the serial I/O status register

This flag is valid only in the parity enable state in the UART mode.

■Framing error flag (bit 5)

In the UART mode, this flag judges whether frame synchronization is abnormal.

When the stop bit of receive data cannot be received at the set timing, this flag is set to “1.”

This flag is set to “0” by one of the following operations.

•Set the receive enable bit to “0”
•Write data (arbitrary) into the serial I/O status register

This flag is valid only in the UART mode.

■Summing error flag (bit 6)

This flag is set to “1” when an overrun error, parity error, or framing error occurs.

This flag is set to “0” by one of the following operations.

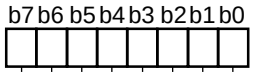
•Set the receive enable bit to “0”
•Write data (arbitrary) into the serial I/O status register

This flag is valid in both the clock synchronous mode and the UART mode.

(3) Serial I/O control register (address 001A16)

This register controls various functions related to the serial I/O, such as transmit/receive modes, clocks, and pin functions. All the bits of this register are read and written by software.

Serial I/O control register



Serial I/O control register (SIOCON) [Address 1A 16]

B	Name	Functions	At reset	R	W
0	BRG count source selection bit (CSS)	0: f(XIN) 1: f(XIN)/4	0	○	○
1	Serial I/O synchronization clock selection bit (SCS)	•In clock synchronous mode 0: BRG output/4 1: External clock input •In UART mode 0: BRG output/16 1: External clock input/16	0	○	○
2	$\overline{\text{SRDY}}$ output enable bit (SRDY)	0: P47/$\overline{\text{SRDY}}$ pin operates as I/O port P47 1: P47/$\overline{\text{SRDY}}$ pin operates as signal output pin $\overline{\text{SRDY}}$ ($\overline{\text{SRDY}}$ signal indicates receive enable state)	0	○	○
3	Transmit interrupt source selection bit (TIC)	0: When transmit buffer has emptied 1: When transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0: Transmit disabled 1: Transmit enabled	0	○	○
5	Receive enable bit (RE)	0: Receive disabled 1: Receive enabled	0	○	○
6	Serial I/O mode selection bit (SIOM)	0: Clock asynchronous serial I/O (UART) mode 1: Clock synchronous serial I/O mode	0	○	○
7	Serial I/O enable bit (SIOE)	0: Serial I/O disabled (pins P44–P47 operate as I/O pins) 1: Serial I/O enabled (pins P44–P47 operate as serial I/O pins)	0	○	○

Fig. 2.5.16 Structure of serial I/O control register

■BRG count source selection bit (bit 0)
Selects a count source to be input to the BRG. In the "0" state, an undivided XIN input signal is input to the BRG. In the "1" state, an XIN input signal divided by 4 is input to the BRG.

■Serial I/O synchronization clock selection bit (bit 1)
Selects a synchronizing clock to be used in the serial I/O.
●Clock synchronous mode
When this bit is set to "0," a BRG output divided by 4 becomes a shift clock.
In the "1" state, an external clock (P46/SCLK pin input) becomes a shift clock as it is.
●UART mode
In the "0" state, a BRG output divided by 16 becomes a shift clock. In the "1" state, an external clock (P46/SCLK pin input) divided by 16 becomes a shift clock.

■$\overline{\text{SRDY}}$ output enable bit (bit 2)
When the $\overline{\text{SRDY}}$ function is used in the clock synchronous mode, set this bit to "1." In the "0" state, the P47/$\overline{\text{SRDY}}$ pin functions as an I/O port P47.
In the UART mode, the value of this bit is invalid, so that the P47/$\overline{\text{SRDY}}$ pin functions as an I/O port P47.

■Transmit interrupt source selection bit (bit 3)
Determines a source which generates a serial I/O transmit interrupt request. In the "0" state, a serial I/O transmit interrupt request occurs at the time when the values of the transmit buffer register are transferred to the transmit shift register.
In the "1" state, a serial I/O transmit interrupt request occurs at the time when the shift operation of the transmit shift register is completed.

■Transmit enable bit (bit 4)
Controls a transmit operation. This bit controls as shown in Table 2.5.3 only when the serial I/O enable bit is "1" (serial I/O enabled). When the serial I/O enable bit is "0" (serial I/O disabled), this bit is invalid.

Table 2.5.3 Control contents of transmit enable bit

Transmit enable bit	P45/TXD pin function	Transmit buffer empty flag *1	Transmit shift register shift completion flag*2
0	Port P45	Set to "0"	
1	Data transmit pin TXD	Flag function is valid	

*1: Bit 0 of serial I/O status register
*2: Bit 2 of serial I/O status register

■Receive enable bit (bit 5)

Controls receive operation. This bit controls as shown in Table 2.5.4 only when the serial I/O enable bit (bit 7) is "1" (serial I/O enabled). When the serial I/O enable bit is "0" (serial I/O disabled), this bit is invalid.

Table 2.5.4 Control contents of receive enable bit

Receive enable bit	P44/RxD pin function	Receive buffer full flag *1	Each error flag*2
0	Port P44	Set to "0"	
1	Data receive pin RxD	Flag function is valid	

*1: Bit 1 of serial I/O status register

*2: Bits 3, 4, 5, and 6 of serial I/O status register

■Serial I/O mode selection bit (bit 6)

Selects a transmit/receive mode of the serial I/O. In the UART mode, set this bit to "0." In the clock synchronous mode, set it to "1."

■Serial I/O enable bit (bit 7)

When the serial I/O function is used, set this bit to "1."

When the bit is set to "1," the pins P44/RxD, P45/TxD, and P46/SCLK function as RxD, TxD, and SCLK respectively (Furthermore, when the $\overline{\text{SRDY}}$ output enable bit is set to "1," the P47/$\overline{\text{SRDY}}$ pin functions as an $\overline{\text{SRDY}}$ pin).

In the "0" state, they function as ports P44 to P47 respectively.

(4) UART control register (address 001B16)

Controls the transfer data format in the UART mode and the output format of the P45/TxD pin.

UART control register

b7 b6 b5 b4 b3 b2 b1 b0

UART control register (UARTCON) [Address 1B 16]

B	Name	Functions	At reset	R	W
0	Character length selection bit (CHAS)	0: 8 bits 1: 7 bits	0	○	○
1	Parity enable bit (PARE)	0: Parity checking disabled 1: Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0: Even parity 1: Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0: 1 stop bit 1: 2 stop bits	0	○	○
4	P45/TxD P-channel output disable bit (POFF)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5 to 7	Nothing is allocated. These bits cannot be written to and are fixed to "1" at reading.		1	1	×

Fig. 2.5.17 Structure of UART control register

■Character length selection bit (bit 0)

Selects data bit length of the UART transfer data format.

In the "0" state, the data bit length is 8 bits. In the "1" state, the data bit length is 7 bits.

■Parity enable bit (bit 1)

This bit is set to "1" to make a parity check and to "0" to make no parity check.

In the "1" state, the parity error flag becomes valid.

■Parity selection bit (bit 2)

Selects a parity type of the UART transfer data format.

In the "0" state, the parity type is an even parity. In the "1" state, it is an odd parity.

■Stop bit length selection bit (bit 3)

Selects a stop bit length of the UART transfer data format.

In the "0" state, the stop bit length is 1 stop bit.

In the "1" state, the stop bit length is 2 stop bits.

■P45/TxD P-channel output disable bit (bit 4)

Controls the output type of the P45/TxD pin.

In the "0" state, the output type is CMOS output in the output mode. In the "1" state, the output type is N-channel open-drain output in the output mode.

The 5 low-order bits of the UART control register can be read and written. The 3 high-order bits are unused and read-only bits. At reading, all the bits are set to "1."

Table 2.5.5 Relation between UART control register and transfer data formats

UART control register				Transfer data format
b3	b2	b1	b0	
0	X	0	0	1 ST–8 DATA–1 SP
0	X	0	1	1 ST–7 DATA–1 SP
0	X	1	0	1 ST–8 DATA–1 PA–1 SP
0	X	1	1	1 ST–7 DATA–1 PA–1 SP
1	X	0	0	1 ST–8 DATA–2 SP
1	X	0	1	1 ST–7 DATA–2 SP
1	X	1	0	1 ST–8 DATA–1 PA–2 SP
1	X	1	1	1 ST–7 DATA–1 PA–2 SP

X: “0” or “1”
ST: Start bit
DATA: Data bit
PA: Parity bit
SP: Stop bit

2.5.4 Register setting example

(1) Clock synchronous serial I/O mode

Figure 2.5.18 and Figure 2.5.19 show a transmitting method in the clock synchronous mode. Figure 2.5.20 and Figure 2.5.21 show a receiving method in the clock synchronous mode.

[Notes on use]

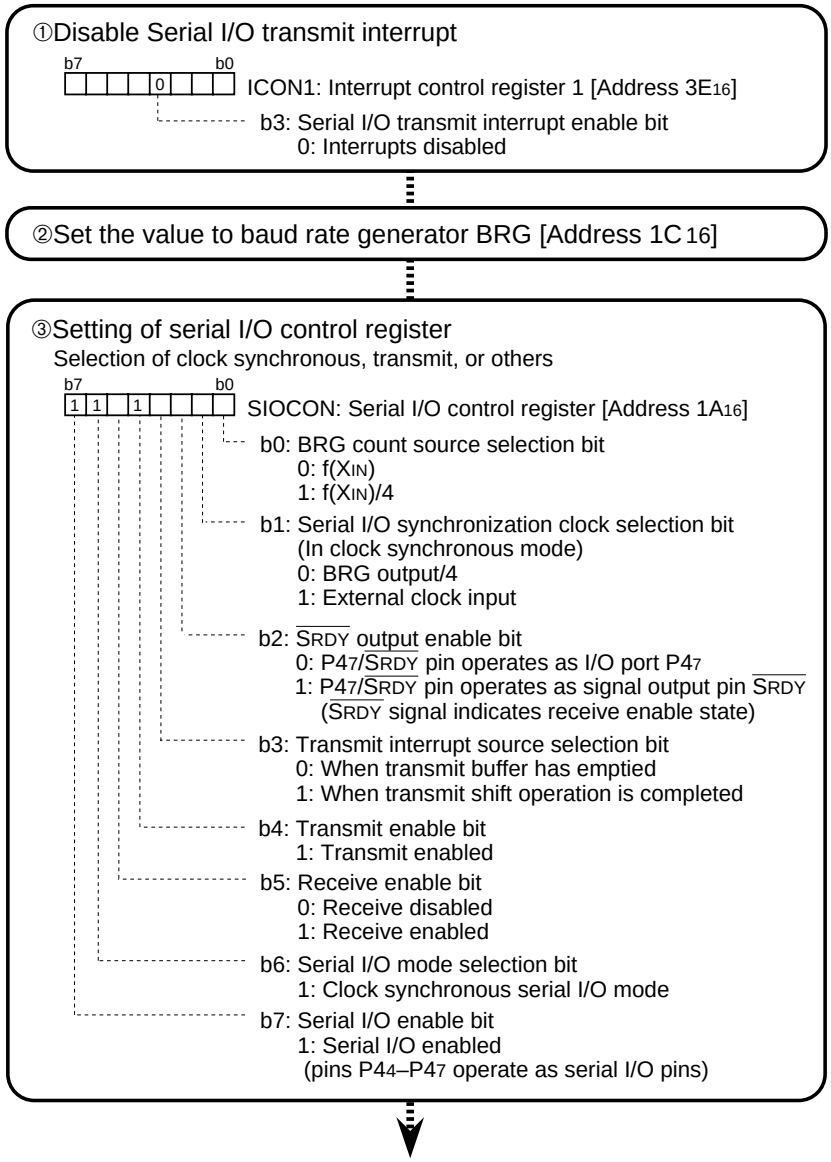
Notes 1: To use an INT pin or input port for $\overline{\text{SRDY}}$ watchdog, set as required.

2: When an external clock is selected in setting ③ below, BRG setting is not required in setting ② below.

3: In the full duplex data transfer mode, set the receive enable bit (bit 5) to "1" (receive enabled) in setting ③ below.

4: To use a serial I/O transmit interrupt, set in the following sequence.

5: When no serial I/O transmit interrupt is used, omit settings ①, ④, ⑤, ⑥ and ⑧ below.



Continued to Figure 2.5.19

Fig. 2.5.18 Transmitting method in clock synchronous mode (1)

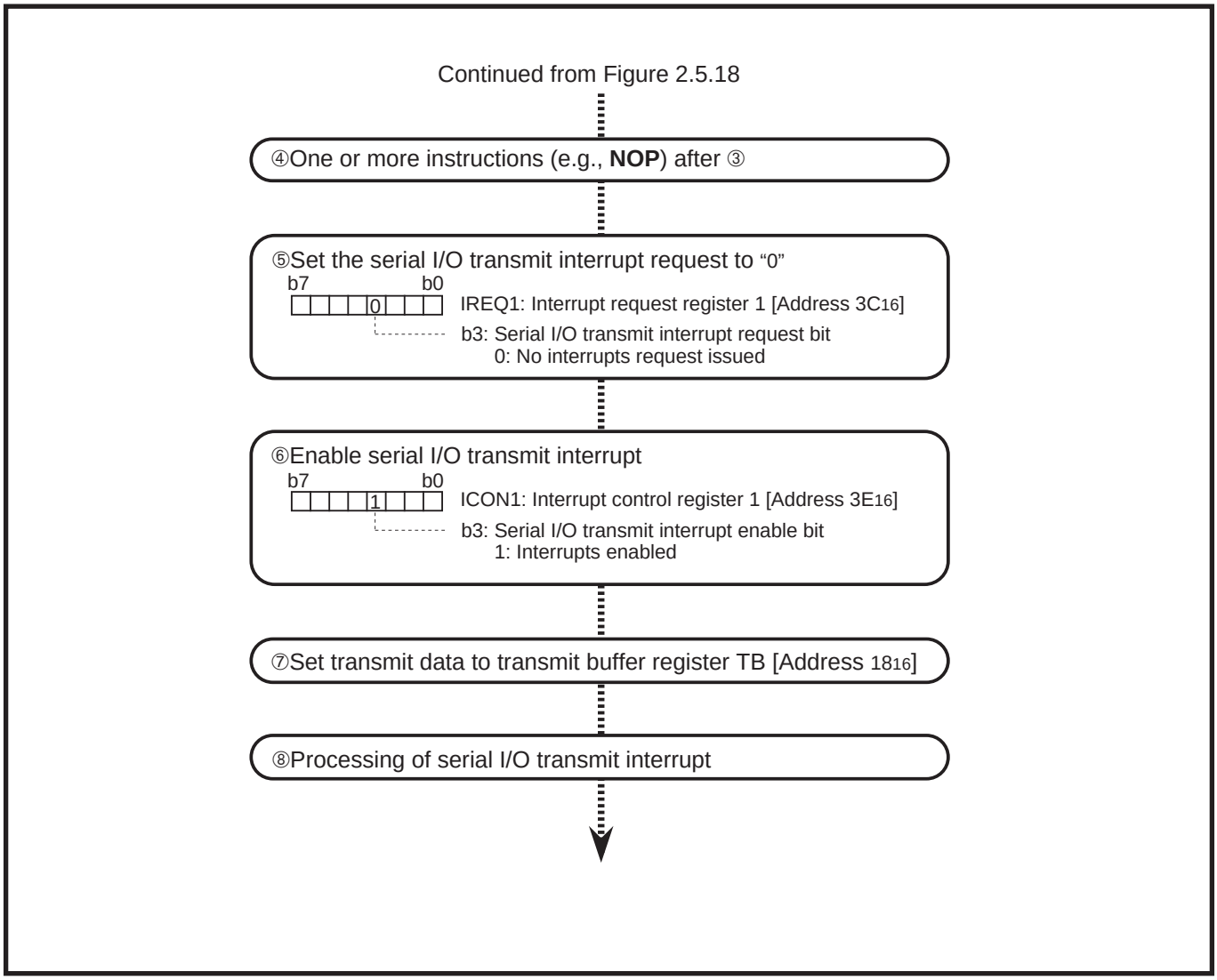


Fig. 2.5.19 Transmitting method in clock synchronous mode (2)

[Notes on use]

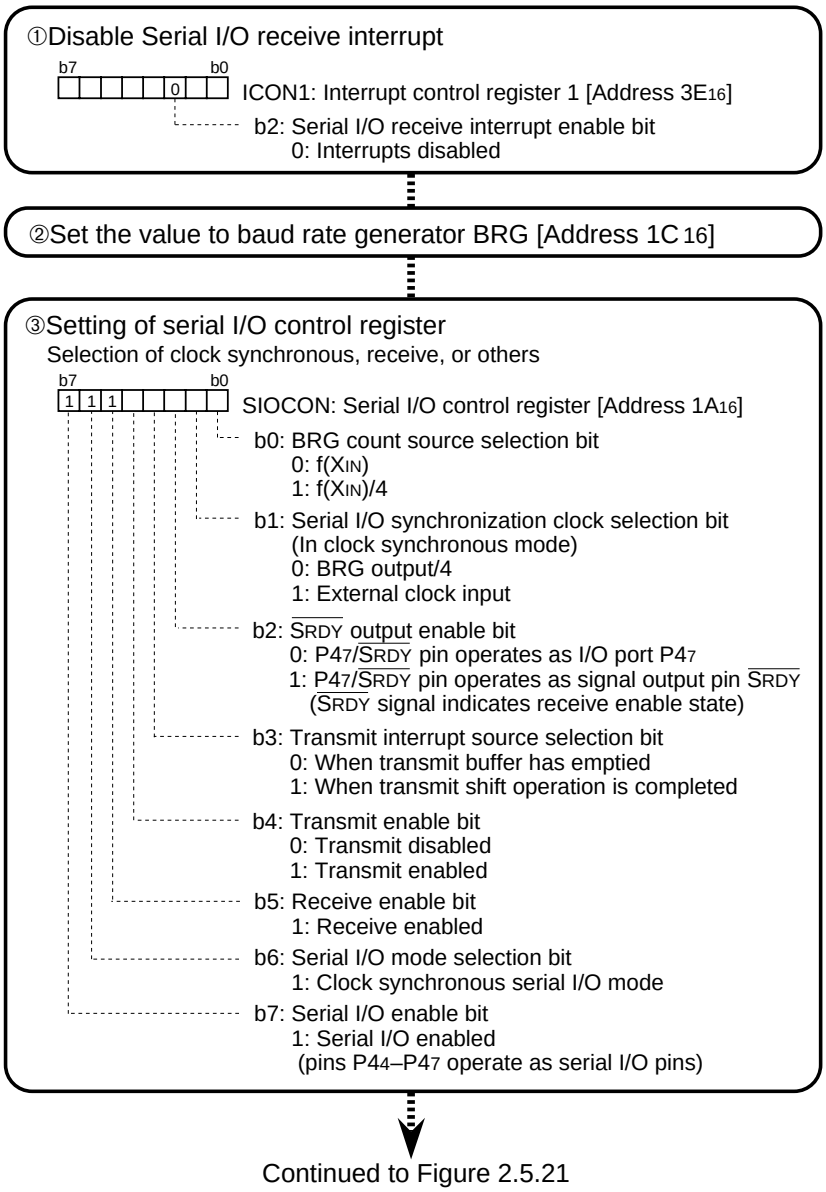
Notes 1: To use an INT pin or input port for $\overline{\text{SRDY}}$ watchdog, set as required.

2: When an external clock is selected in setting ③ below, BRG setting is not required in setting ② below.

3: In the full duplex data transfer mode, set the receive enable bit (bit 4) to "1" (receive enabled) in setting ③ below.

4: To use a serial I/O receive interrupt, set in the following sequence.

5: When no serial I/O receive interrupt is used, omit setting ①, ④, ⑤, ⑥ and ⑧ below.



Continued to Figure 2.5.21

Fig. 2.5.20 Receiving method in clock synchronous mode (1)

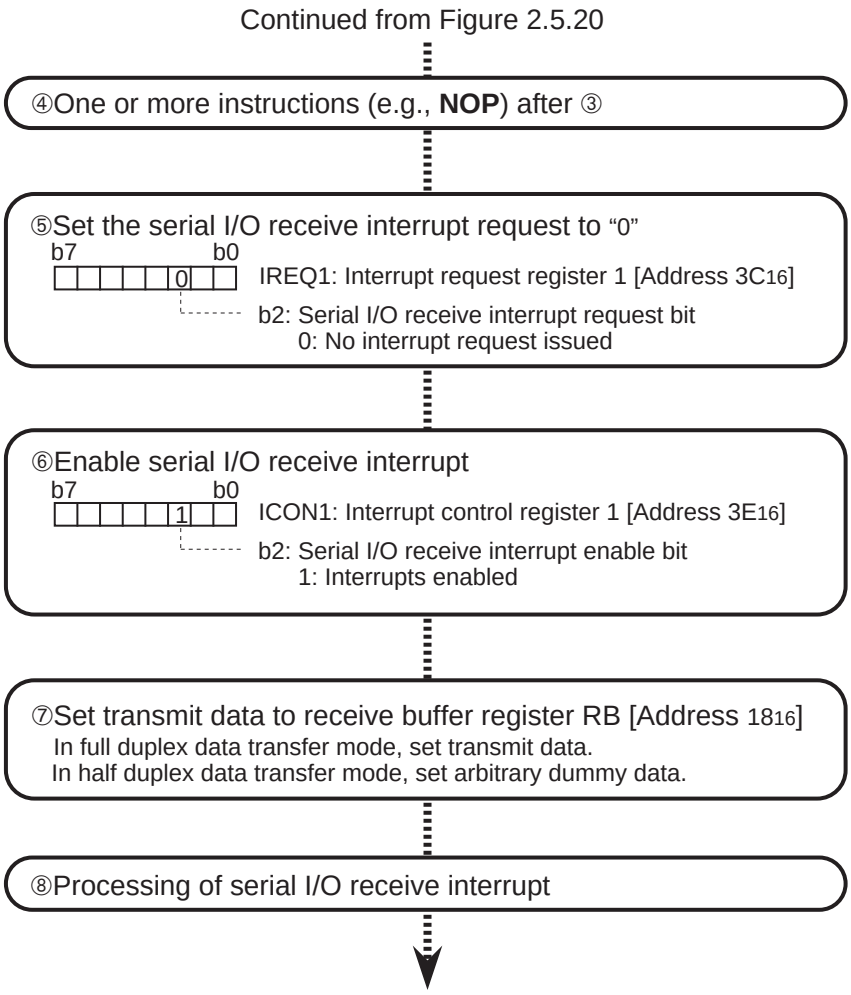


Fig. 2.5.21 Receiving method in clock synchronous mode (2)

(2) Clock asynchronous serial I/O (UART) mode

Figure 2.5.22 and Figure 2.5.23 show a transmitting method in the UART mode. Figure 2.5.24 and Figure 2.5.25 show a receiving method in the UART mode.

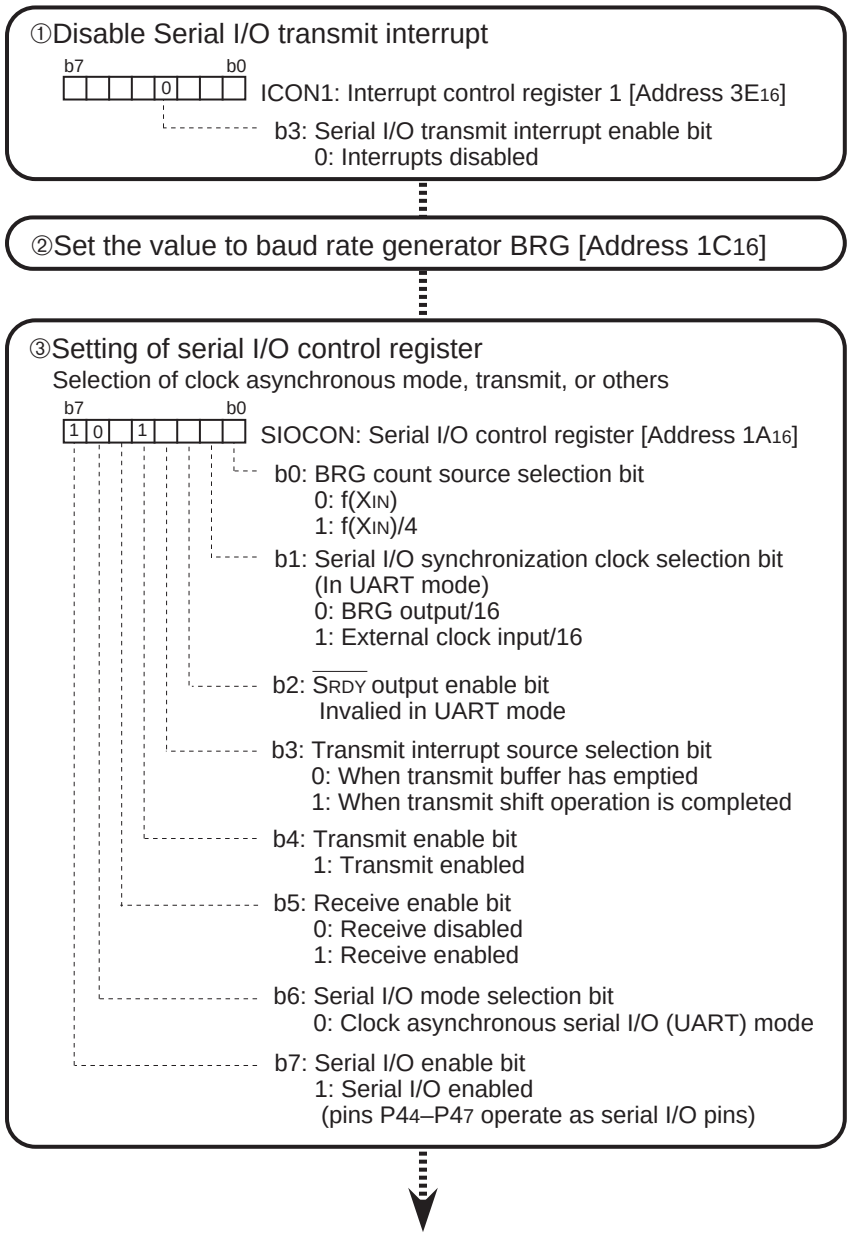
[Notes on use]

Notes 1: When an external clock is selected in setting ③ below, BRG setting is not required in setting ② below.

2: In the full duplex data transfer mode, set the receive enable bit (bit 5) to "1" (receive enabled) in setting ③ below.

3: To use a serial I/O transmit interrupt, set in the following sequence.

4: When no serial I/O transmit interrupt is used, omit setting ①, ⑤, ⑥ and ⑧ below.



Continued to Figure 2.5.23

Fig. 2.5.22 Transmitting method in UART mode (1)

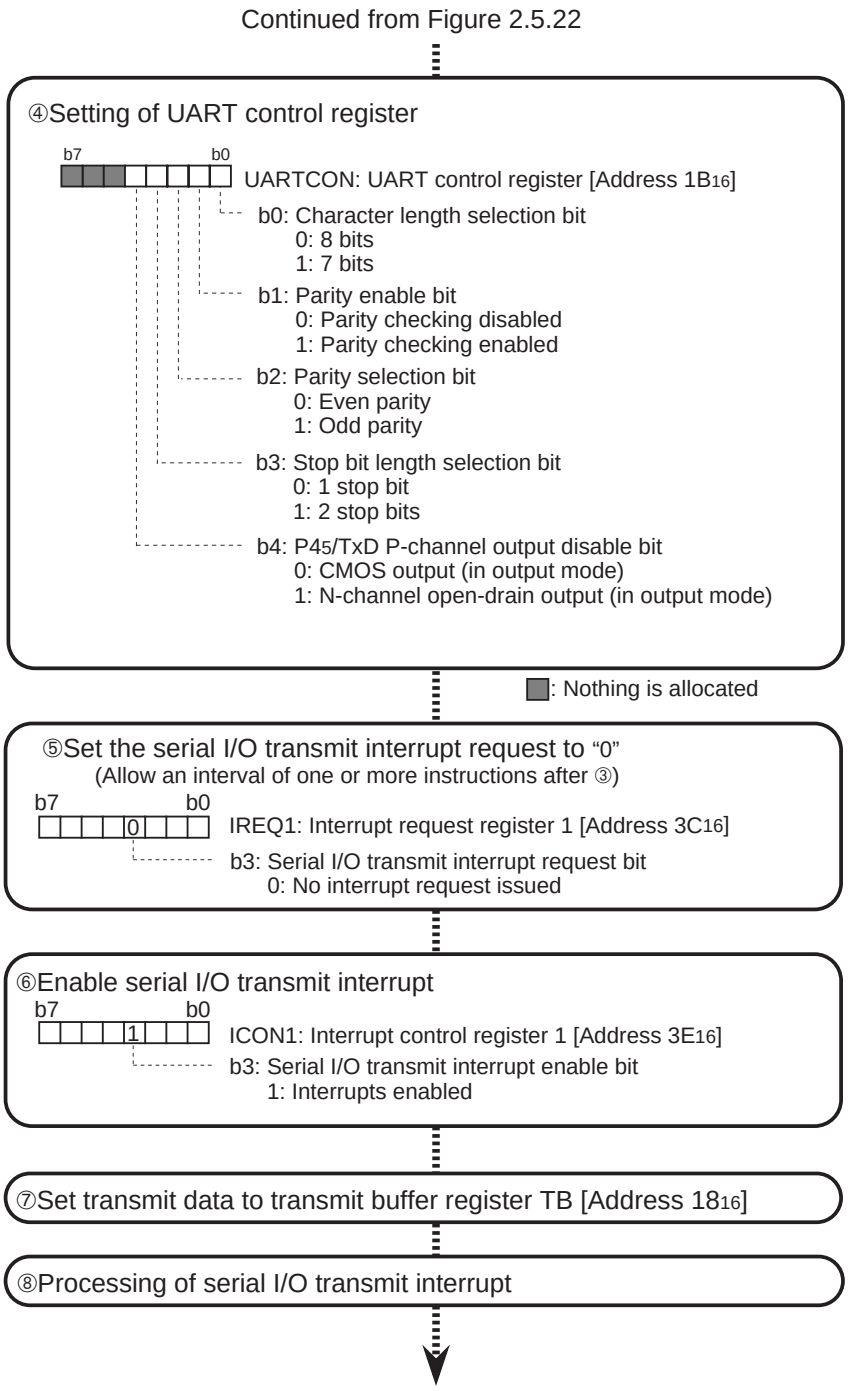


Fig. 2.5.23 Transmitting method in UART mode (2)

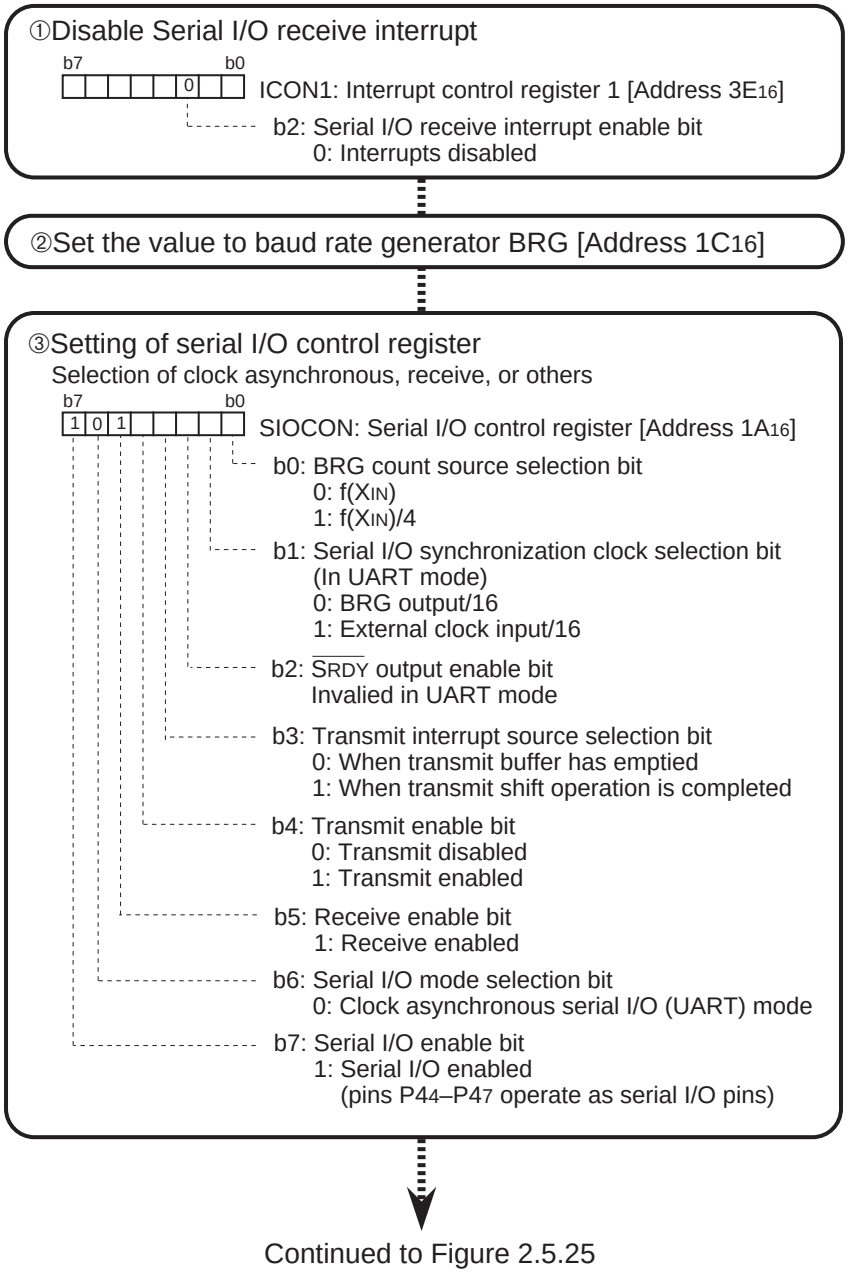
[Notes on use]

Notes 1: When an external clock is selected in setting ③ below, BRG setting is not required in setting ② below.

2: In the full duplex data transfer mode, set the receive enable bit (bit 4) to "1" (receive enabled) in setting ③ below.

3: To use a serial I/O receive interrupt, set in the following sequence.

4: When no serial I/O receive interrupt is used, omit setting ①, ⑤, ⑥ and ⑦ below.



Continued to Figure 2.5.25

Fig. 2.5.24 Receiving method in UART mode (1)

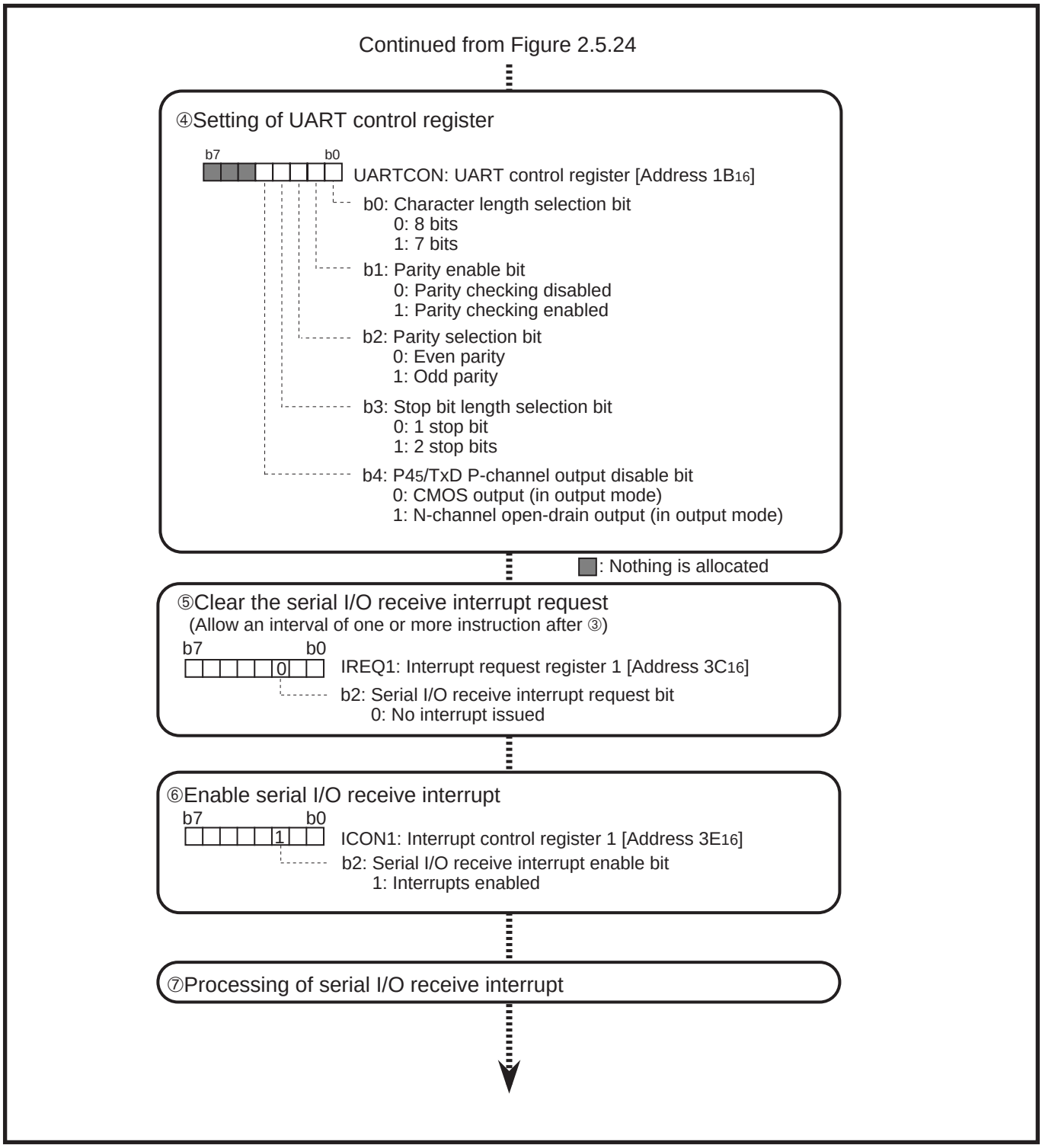


Fig. 2.5.25 Receiving method in UART mode (2)

(3) Initialization of serial I/O operation

The operating procedure of the serial I/O control register for initialization of the serial I/O operation is described below.

■Initialization of receive operation

By setting the receive enable bit (bit 5 of SIOCON) to "0" or setting the serial I/O enable bit (bit 7 of SIOCON) to "0," the receive operation is stopped and initialized as shown below. The initialization items of receive operation are as follows.

•Stopping and initializing the shift clock to the receive shift register.
•Setting the receive shift register to "0."
•Setting each error flag (overrun error flag, parity error flag, framing error flag, summing error flag) to "0."
•Setting the receive buffer full flag (RBF) to "0."

■Initialization of transmit operation

Basically, the transmit operation is stopped and initialized by setting the transmit enable bit (bit 4 of SIOCON) to "0." The initialization items of transmit operation are as follows.

•Stopping and initializing the shift clock to the transmit shift register
•Setting the receive shift register to "0" (However, when an external clock is used in the clock synchronous mode, the receive shift register is not set to "0" unless the input clock of the SCLK pin is "H").
•Setting the transmit buffer empty flag (bit 0 of SIOSTS) and the transmit shift register shift completion flag (bit 2 of SIOSTS) to "0."
(When bit 4 is set to "0," bits 0 and 2 are cleared to "0" forcibly. After that, when bit 4 is set to "1," bits 0 and 2 are set to "1.")

When all conditions below are satisfied, initialization is not performed only by setting bit 4 of SIOCON to "0." It is also necessary to set bit 5 of SIOCON to "0."

•In the full duplex data transfer
•In the clock synchronous mode
•When an internal clock is used
•When bit 5 of SIOCON is "0" (receive enabled)

In the clock synchronous mode of the full duplex data transfer, the same clock is used for transmission and reception.

When an internal clock is used, the shift clock is started by writing data into the receive buffer at both transmission and reception, so both transmit and receive operations use a clock generating circuit of the transmitter.

Because of this, the serial I/O is designed so that even if only a receive operation is performed, the transmit circuit may be operated internally to generate a shift clock when an internal clock is used in the clock synchronous mode. Accordingly, note that the transmitter may operate even when bit 4 of SIOCON is "0." The transmit operation cannot be initialized only by setting the serial I/O enable bit (bit 7 of SIOCON) to "0."

(4) Processing upon occurrence of an errors

■Parity error, framing error, or summing error

If a parity error, a framing error, or a summing error occurs, the flag corresponding to each error in the serial I/O status register is set to "1." These flags cannot be cleared to "0" automatically, so set them to "0" by software.

The parity error flag, framing error flag, and summing error flag is set to "0" by setting the receive enable bit to "0" or writing dummy data into the serial I/O status register.

■Overrun error

An overrun error occurs when data is already input in the receive buffer register and yet all data is input in the receive shift register.

If an overrun error occurs, the data of the receive shift register is not transferred and the data of the receive buffer register is held. At this time, even if the data of the receive buffer register is read out, the data of the receive shift register is not transferred.

Consequently, the data of the receive shift register becomes unreadable, so that the receive data becomes invalid.

If an overrun error occurs, after set the overrun error flag of the serial I/O status register to "0", perform a receive operation again.

The overrun error flag is set to "0" by one of the following operations.

•Set the serial I/O enable bit to "0"
•Set the receive enable bit to "0"
•Write data (arbitrary) into the serial I/O status register

2.5.5 Notes on use

(1) Notes on clock selection

The 3822 group can select either internal clock or external clock as a synchronizing clock. When an external clock is selected as an synchronizing clock in the clock synchronous mode, note the following.

■In the clock synchronous mode

①For an external clock source, when the duty cycle is 50%, use the following clock.

1.25 MHz or less.....at V_{CC} = 4.0 V to 5.5 V
500 kHz or less.....at V_{CC} = 2.5 V to 4.0 V

To change the duty cycle, set the "H" and "L" widths as follows.
370 ns min. ..... at V_{CC} = 4.0 V to 5.5 V
950 ns min. ..... at V_{CC} = 2.5 V to 4.0 V

②The shift operation of the transmit shift register or the receive shift register is continued while synchronizing clocks are input to the serial I/O circuit. Accordingly, stop a synchronizing clock input after 8 clocks are input.
When the internal clock is selected, the synchronizing clock input is automatically stopped.

③To select an external clock as a synchronizing clock at data transmission, set the transmit enable bit to "1" and write data into the transmit buffer register while the S_{CLK} signal is "H."

When an external clock is selected as a synchronizing clock in the UART mode, note the following.

■In the UART mode

For an external clock source, when the duty ratio is 50%, use the following clock.
5 MHz or less.....at V_{CC} = 4.0 V to 5.5 V
2 MHz or less.....at V_{CC} = 2.5 V to 4.0 V

To change the duty cycle, set the "H" and "L" widths as follows.
93 ns min. .....at V_{CC} = 4.0 V to 5.5 V
238 ns min. .....at V_{CC} = 2.5 V to 4.0 V

(2) For serial I/O transmit or receive interrupts

①For a serial I/O transmit interrupt, set a value in the serial I/O control register, then set the serial I/O transmit interrupt request bit (bit 3 at address $003C_{16}$) to "0" with the **CLB** instruction.
②After setting ①, set the serial I/O transmit enable bit (bit 3 at address $003E_{16}$) to "1."
③For a serial I/O receive interrupt, set a value in the serial I/O control register, then set the serial I/O receive interrupt request bit (bit 2 at address $003C_{16}$) to "0" with the **CLB** instruction.
④After setting ③, set the serial I/O receive interrupt enable bit (bit 2 at address $003E_{16}$) to "1."

(3) Transmit interrupt request when the transmit enable bit is "1"

When the transmit enable bit is set to "1," the transmit buffer empty flag and the transmit shift register shift completion flag are set to "1." Accordingly, even if either timing is selected as transmit interrupt generating timing, an serial I/O transmit interrupt request occurs and the serial I/O transmit interrupt request bit is set to "1."

To use a serial I/O transmit interrupt, set the transmit enable bit to "1," then set the serial I/O transmit interrupt request bit to "0." After that, set the serial I/O transmit interrupt enable bit to "1" (interrupts enabled).

(4) For disabling transmission after completion of 1-byte data transmission

As a means to know the completion of data transmission, a reference to the transmit shift register shift completion flag (TSC flag) is available in the 3822 group.

The TSC flag is cleared to "0" during data transmission. Upon the completion of data transmission, this flag is set to "1." Accordingly, after confirming that the TSC flag is set, disable transmission. The transmission can thus be terminated after 1-byte transmission. However, the TSC flag is set to "1" even when the serial I/O is set to "1" (serial I/O enabled). After that, it is not cleared to "0" until transmission is started by generating a shift clock. For this reason, if transmission is disabled by referring to the TSC flag at this time, data is not transmitted. After the transmission is started, refer to the TSC flag.

(5) When the $P4_5$/TxD pin is used as an N-channel open-drain output

Bit 4 of the UART control register (address $001B_{16}$) is a P-channel output disable bit of $P4_5$/TxD pin. The bit 4 is valid in an ordinary port, in the clock synchronous mode, or in the UART mode.

When this bit is "0," the ordinary CMOS output is selected. When the bit is "1," the N-channel open-drain output is selected.

However, do not apply to the $P4_5$/TxD a voltage of V_{CC} + 0.3 V or more even when it is used as a serial I/O function pin of the N-channel open-drain output.

2.6 A-D converter

2.6.1 Explanation of operations

The operations of the A-D converter are described below.

(1) When an internal trigger is selected

(To cause an ADT/A-D conversion interrupt request upon completion of A-D conversion)

■By setting bit 3 of the A-D control register (address 0034_{16}) to "0," A-D conversion is started.

■Upon the completion of A-D conversion, bit 3 of the A-D control register is set to "1." At the same time, an ADT/A-D conversion interrupt request occurs.

(2) When an external trigger is selected

(To cause an ADT/A-D conversion interrupt request upon completion of A-D conversion)

■By setting bit 3 of A-D cintrol register to "0" and then inputting a falling signal to the ADT pin, A-D conversion is started.

■Upon the completion of A-D conversion, bit 3 of the A-D control register is set to "1." At the same time, an ADT/A-D conversion interrupt request occurs.

(3) When an external trigger is selected

(To cause an ADT/A-D conversion interrupt request upon inputting a falling signal to the ADT pin)

■By setting bit 3 of A-D cintrol register to "0" and then inputting a falling signal to the ADT pin, A-D conversion is started. At the same time, an ADT/A-D conversion interrupt request occurs.

■Upon the completion of A-D conversion, bit 3 of the A-D control register is set to "1."

2.6.2 Conversion method

As an A-D conversion method, successive comparison approximation is adopted.
The comparison voltage "V_{ref}" internally generated is compared with the analog input voltage "V_{IN}" which is input from an analog input pin (AN_0–AN_7) and the result is input successively to each bit of the A-D conversion register (address 0035_{16}) to obtain a digital value.

(1) Function of each block

The function of each block in the A-D converter is shown below.

■Comparison voltage generator (resistor ladder)
Divides the voltage between the AV_{SS} pin and the V_{REF} pin by 256 and output a divided voltage to the comparator as comparison voltage "V_{ref}."

■Channel selector
Connects an analog input pin selected by bits 2 to 0 of the A-D control register (address 0034_{16}) to the comparator.

■Comparator
Compares the analog input voltage "V_{IN}" with the comparison voltage "V_{ref}" and input the result to the A-D conversion register.

(2) Internal Operation

At the time when the A-D conversion is started, the following operations are automatically performed.

■The A-D conversion register becomes "00_{16}."
■The most significant bit of the A-D conversion register is set to "1."
■The comparison voltage "V_{ref}" is input to the comparator. The comparison voltage "V_{ref}" is specified by the A-D conversion register contents "n" and the reference voltage "V_{REF}" which is input from the V_{REF} pin.

Table 2.6.1 shows an expression for the comparison voltage "V_{ref}."

Table 2.6.1 Expression for comparison voltage "V_{ref}"

A-D conversion register contents "n" (decimal notation)	V_{ref} (V)
0	0
1 to 255	$\frac{V_{REF}}{256} \times (n - 0.5)$

2.6 A-D converter

■The comparison voltage "V_{ref}" is compared 8 times with the analog input voltage "V_{IN}." Each time a comparison ends, the result is input to the A-D conversion register. With a change of the A-D conversion register, the comparison voltage "V_{ref}" changes, too.
Figure 2.6.1 shows changes in the A-D conversion register and comparison voltage during A-D conversion.

①Determination of the most significant bit (bit 7) of the A-D conversion register
Bit 7 is determined by the first comparison result.
The comparison voltage "V_{ref}" is compared with the analog input voltage "V_{IN}" and the result determines bit 7 as follows.

When $V_{ref} < V_{IN}$: bit 7 holds "1"
When $V_{ref} > V_{IN}$: bit 7 becomes "0"

②Determination of bits 6 to 0 of the A-D conversion register
Bit 6 is determined by the second comparison result.
First, bit 6 of the A-D conversion register is set to "1." Next, the comparison voltage "V_{ref}" is compared with the analog input voltage "V_{IN}" and the result determines bit 6 as follows.

When $V_{ref} < V_{IN}$: bit 6 holds "1"
When $V_{ref} > V_{IN}$: bit 6 becomes "0"

Likewise, bits 5 to 0 are determined by the third to eighth comparison results.

With the above operations, the digital value (contents of the A-D conversion register) corresponding to the analog input voltage "V_{IN}" is determined by one bit at a time.

■Upon the completion of A-D conversion, bit 3 of the A-D control register is set to "1."
An A-D conversion result can be obtained by reading out the A-D conversion register after bit 3 of the A-D control register is set to "1."
The A-D conversion result is held in the A-D conversion register until bit 3 of the A-D control register is set to "1" again after the completion of the next A-D conversion.

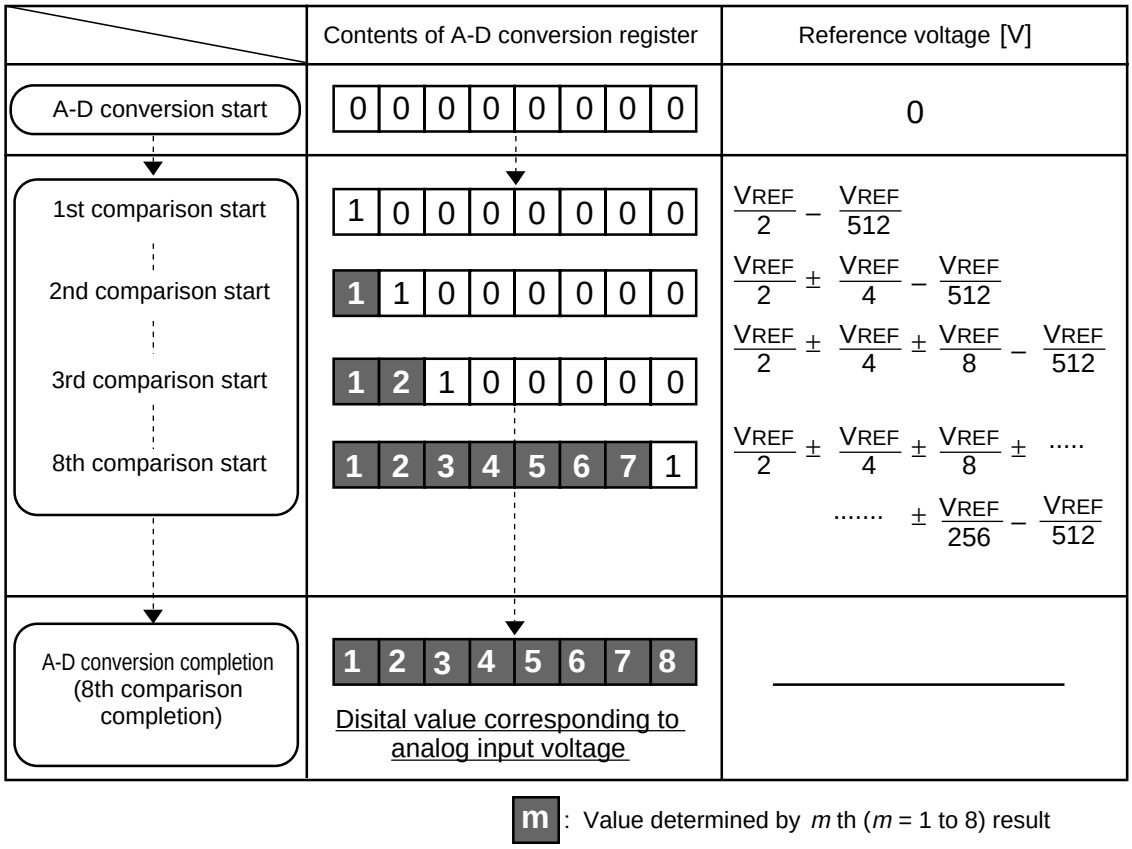


Fig. 2.6.1 Changes in A-D conversion register and comparison voltage during A-D conversion

(3) Conversion time

In the high-speed operation mode, A-D conversion terminates in a maximum 50 cycles (12.5 μs at f(XIN) = 8 MHz) after a start of A-D conversion.

In the middle-speed operation mode, A-D conversion terminates in a maximum 56 cycles (14 μs at f(XIN) = 8 MHz) after a start of A-D conversion.

For the A-D converter, the main clock input oscillation frequency f(XIN) divided by 2 is used (Note 1), so A-D conversion time is obtained basically by the following expression.

$$\text{Conversion clock period} = \frac{2}{f(X_{IN})}$$

A-D conversion time = Conversion clock period × Conversion cycle

However, the number of conversion cycles varies depending on internal clock ϕ and trigger.

Notes 1: Use the A-D converter in the state where bits 5 and 7 of the CPU mode register (address $003B_{16}$) are "0" (high-speed mode or middle-speed mode).

As the comparator is composed of a capacitance coupling, use the A-D converter in a state of f(XIN) ≥ 500 kHz.

2: When an external trigger is selected, the A-D conversion being executed is stopped by inputting a falling signal to the ADT pin during A-D conversion, and A-D conversion is resumed.

The A-D conversion register holds the previous conversion result until A-D conversion is completed.

2.6 A-D converter

(4) Equivalent connection diagram

Figure 2.6.2 shows an A-D converter equivalent connection diagram.

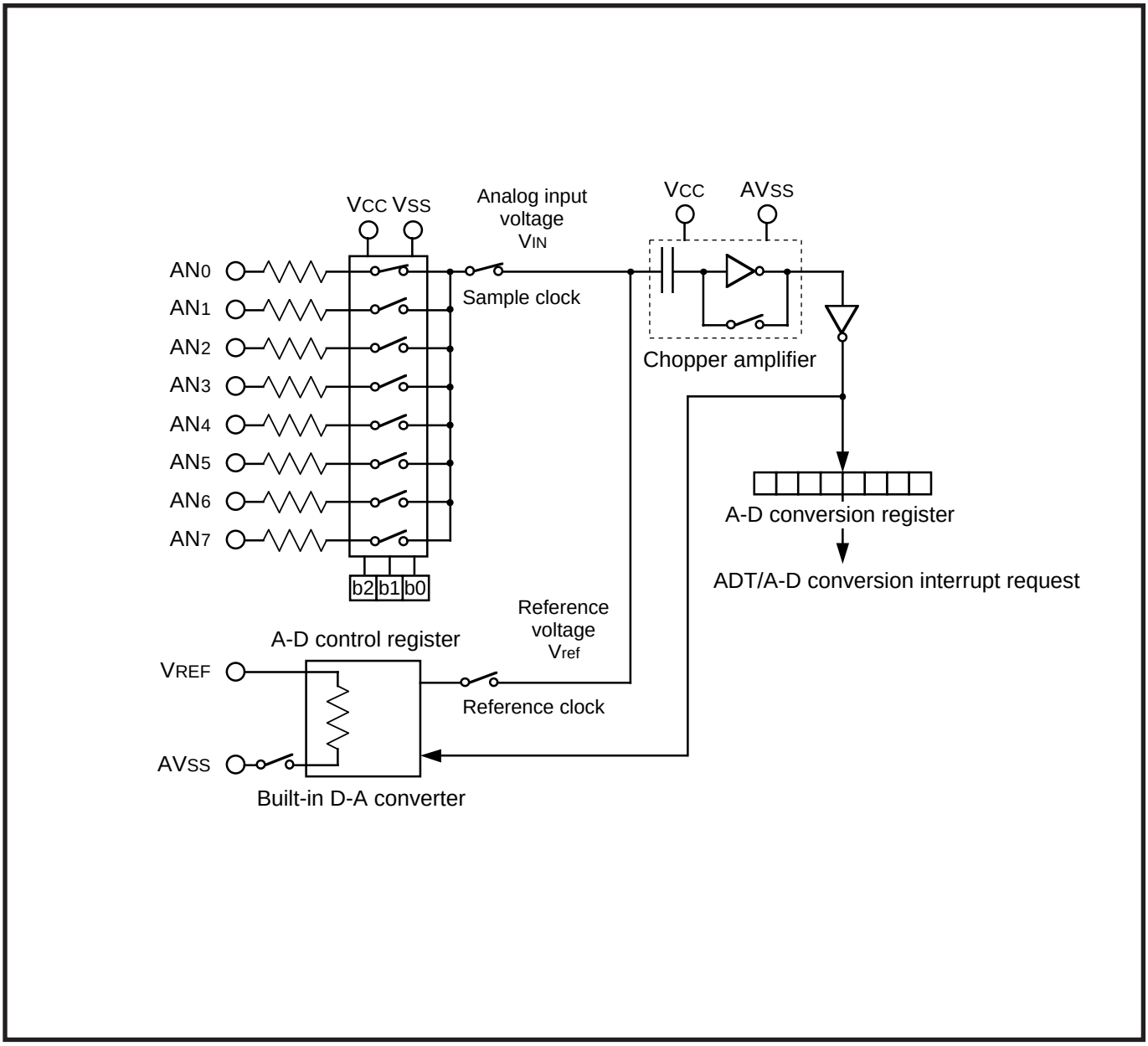


Fig. 2.6.2 A-D converter equivalent connection diagram

2.6.3 Pins

Table 2.6.2 shows a list of pin functions used in the A-D converter.

Table 2.6.2 List of pin functions used in A-D converter

Pins	Name	Functions
AN_0–AN_7	Analog input	•Analog input voltage input pins. •Apply a voltage of AV_{SS}–V_{CC}. •These pins are also used as $P6_0$–$P6_7$.
ADT	External trigger input	•External trigger input pin. •This pin is also used as $P5_7$.
V_{REF}	Reference voltage input	•Reference voltage input pin. •Apply a voltage of 2 V–V_{CC}.
AV_{SS}	Analog power source voltage input	•GND input pin. •Apply the same voltage as the V_{SS} pin.

(1) Pin-related setting

■Analog input pins (AN_0–AN_7)

When using the A-D converter, select a pin to be used as an analog input pin by bits 2 to 0 of the A-D control register (address 0034_{16}).

Use the A-D converter in the state where the bit of the port P6 direction register (address $000D_{16}$) corresponding to the pin used as an analog input pin is "0."

■External trigger input pin (ADT)

When using the external trigger, set bit 5 of the A-D control register to "1."

Use the A-D converter in the state where bit 7 of the port P5 direction register (address $000B_{16}$) is "0."

Note: The ports P5 and P6 direction registers are not readable. To set these registers, use the **STA** instruction, **LDM** instruction, or other instructions.

2.6 A-D converter

2.6.4 Related registers

Figure 2.6.3 shows a memory allocation of the A-D converter-related registers. Each register is described below.

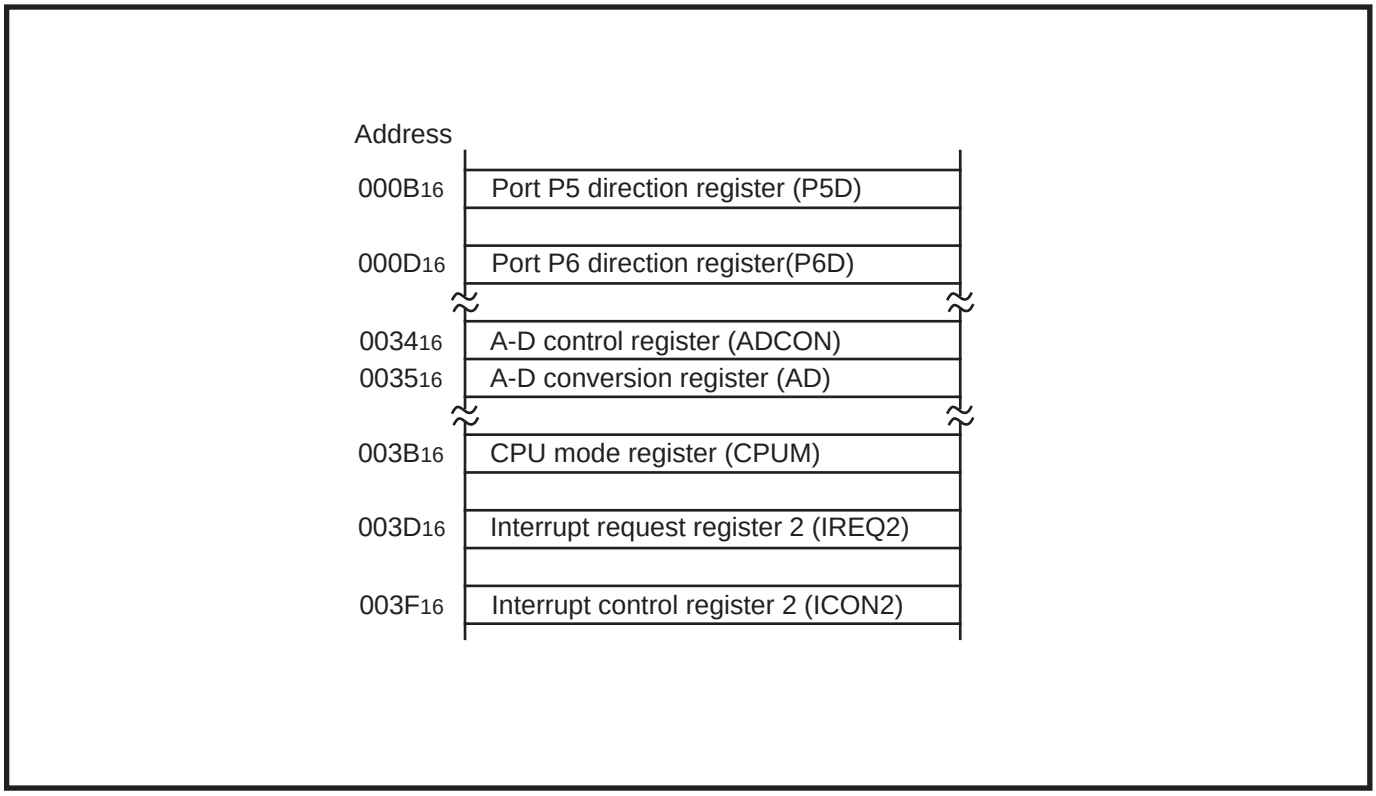
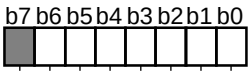


Fig. 2.6.3 Memory allocation of A-D converter-related registers

(1) A-D control register (address 0034$_{16}$)

The A-D control register consists of bits which controls for the A-D converter. Figure 2.6.4 shows the structure of the A-D control register. Each bit is described below.

A-D control register



A-D control register (ADCON) [Address 34 $_{16}$]

B	Name	Functions	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0: AN$_0$ 0 0 1: AN$_1$ 0 1 0: AN$_2$ 0 1 1: AN$_3$ 1 0 0: AN$_4$ 1 0 1: AN$_5$ 1 1 0: AN$_6$ 1 1 1: AN$_7$	0	○	○
1			0	○	○
2			0	○	○
3	A-D conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
4	V_{REF} input switch bit	0: OFF 1: ON	0	○	○
5	A-D external trigger valid bit	0: A-D external trigger invalid (internal trigger selected) 1: A-D external trigger valid (external trigger selected)	0	○	○
6	Interrupt source selection bit	0: At A-D conversion completed 1: At falling of ADT pin input	0	○	○
7	Nothing is allocated. This bit cannot be written to and is fixed “0” at reading.		0	0	×

Note: When an internal trigger is selected, A-D conversion is started by setting bit 3 to “0.” Writing only “0” to bit 3 is valid. Even if “1” is written to bit 3, it is not set to “0.”
Accordingly, to write values to the ADCON without affecting bit 3, set bit 3 to “1.”

Fig. 2.6.4 Structure of A-D control register

2.6 A-D converter

■Analog input pin selection bits: bits 2 to 0
Select an analog input pin.
The pins which are not used as analog input pins of port P6 function as programmable I/O ports.

■A-D conversion completion bit: bit 3
Indicates the operating state of the A-D converter. During A-D conversion, this bit is set to "1" after completion of A-D conversion. When an internal trigger is selected, A-D conversion is started by setting this bit to "0." (Note)

■VREF input switch bit: bit 4
Connects the VREF pin to the comparison voltage generator. When the A-D converter is used, be sure to set this bit to "1." When the A-D converter is not used, the power dissipation is reduced by setting this bit to "0."

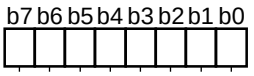
■A-D external trigger valid bit: bit 5
Determines whether A-D conversion is started by an external trigger or internal trigger.

■Interrupt source selection bit: bit 6
Selects ADT/A-D conversion interrupt request generating timing.

Note: When an internal trigger is selected, set the A-D conversion completion bit after setting bits 2 to 0 and bits 6 to 4 of the A-D control register.

(2) A-D conversion register (address 0035_{16})
The A-D conversion register stores A-D conversion results. This is a read-only register. Figure 2.6.5 shows the structure of the A-D conversion register.

A-D conversion register



A-D conversion register (AD) [Address 35_{16}]

B	Functions	At reset	R	W
0 to 7	Read-only register which stores A-D conversion results.	0	○	×

Fig. 2.6.5 Structure of A-D conversion register

(3) CPU mode register (address 003B16)

The CPU mode register consists of the stack page selection bit and control bits for the internal system clock ϕ. Use the A-D converter in the state where bits 5 and 7 of this register are "0" (high-speed mode or middle-speed mode).

Figure 2.6.6 shows the structure of the CPU mode register.

The operating clock of the A-D converter is the main clock input frequency f(XIN)/2. Use the A-D converter in the state of f(XIN) ≥ 500 kHz.

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	b1b0 00: Single-chip mode 01: 10: 11: } Not available	0	○	○
1			0	○	○
2	Stack page selection bit	0: 0 page 1: 1 page	0	○	○
3	Fix this bit to "1."		1	1	1
4	Port XC switch bit	0: I/O port 1: XCIN, XCOUT	0	○	○
5	Main clock (XIN–XOUT) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bit	0: f(XIN)/2 (high-speed mode) 1: f(XIN)/8 (middle-speed mode)	1	○	○
7	Internal system clock selection bit	0: XIN–XOUT selected (middle-/high-speed mode) 1: XCIN–XCOUT selected (low-speed mode)	0	○	○

Fig. 2.6.6 Structure of CPU mode register

(4) Port P5 direction register (address 000B16)

The port P5 direction register switches the I/O direction of port P5. When an external trigger is selected, hold bit 7 of this register at "0."

Figure 2.6.7 shows the structure of the port P5 direction register.

Port P5 direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port P5 direction register (P5D) [Address B16]

B	Name	Functions	At reset	R	W
0	Port P5 direction register	0 : Port P50 input mode 1 : Port P50 output mode	0	×	○
1		0 : Port P51 input mode 1 : Port P51 output mode	0	×	○
2		0 : Port P52 input mode 1 : Port P52 output mode	0	×	○
3		0 : Port P53 input mode 1 : Port P53 output mode	0	×	○
4		0 : Port P54 input mode 1 : Port P54 output mode	0	×	○
5		0 : Port P55 input mode 1 : Port P55 output mode	0	×	○
6		0 : Port P56 input mode 1 : Port P56 output mode	0	×	○
7		0 : Port P57 input mode 1 : Port P57 output mode	0	×	○

Note : Port P5 direction register cannot be read out (refer to **"2.1 I/O pins"**).

Fig. 2.6.7 Structure of port P5 direction register

(5) Port P6 direction register (address 000D16)

The port P6 direction register switches the I/O direction of port P6. Hold the bit of this register which corresponds to the port used as an analog input pin at "0."

Figure 2.6.8 shows the structure of the port P6 direction register.

Port P6 direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port P6 direction register (P6D) [Address 0D 16]

B	Name	Functions	At reset	R	W
0	Port P6 direction register	0: Port P60 input mode 1: Port P60 output mode	0	×	○
1		0: Port P61 input mode 1: Port P61 output mode	0	×	○
2		0: Port P62 input mode 1: Port P62 output mode	0	×	○
3		0: Port P63 input mode 1: Port P63 output mode	0	×	○
4		0: Port P64 input mode 1: Port P64 output mode	0	×	○
5		0: Port P65 input mode 1: Port P65 output mode	0	×	○
6		0: Port P66 input mode 1: Port P66 output mode	0	×	○
7		0: Port P67 input mode 1: Port P67 output mode	0	×	○

Note: Port P6 direction register cannot be read out (refer to **"2.1 I/O pins"**).

Fig. 2.6.8 Structure of port P6 direction register

(6) Interrupt request register 2 (IREQ2)

The interrupt request register 2 (address 003D16) indicates whether an interrupt request has occurred or not.

Figure 2.6.9 shows the structure of the interrupt request register 2.

The occurrence of an ADT/A-D conversion interrupt request causes bit 6 to be set to "1." The bit 6 is automatically cleared to "0" by the acceptance of the ADT/A-D conversion interrupt request.

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

The occurrence of the ADT/A-D conversion interrupt is controlled by the ADT/A-D conversion interrupt enable bit (refer to the next item).

For details of interrupts, refer to **"2.2 Interrupts."**

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

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

B	Name	Functions	At reset	R	W
0	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key input interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	ADT/A-D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated. This bit cannot be written to and is fixed to "0" at reading.		0	0	×

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

Fig. 2.6.9 Structure of interrupt request register 2

(7) Interrupt control register 2 (ICON2)

The interruot contorol register 2 (address 003F16) controls each interrupt request source.

Figure 2.6.10 shows the structure of the interrupt control register 2.

When bit 6 is "0," the ADT/A-D interrupt request is disabled. When bit 6 is "1," the ADT/A-D interrupt request is enabled.

The bit 6 can be set to "0" or "1" by software.

For details of interrupts, refer to **"2.2 Interrupts."**

Interrupt control register 2

b7	b6	b5	b4	b3	b2	b1	b0
0							

Interrupt control register 2 (ICON2) [Address 3F 16]

B	Name	Functions	At reset	R	W
0	CNTR0 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	CNTR1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Timer 1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	INT2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	INT3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Key input interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	ADT/A-D conversion interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Fix this bit to "0."		0	0	0

Fig. 2.6.10 Structure of interrupt control register 2

2.6.5 Measuring various A-D converter standard characteristics

The measuring various A-D converter standard characteristics is described below.

(1) Absolute accuracy

The absolute accuracy is the difference expressed in LSB between an output code obtained by actual measurement and an expected output code of the A-D converter with ideal characteristics.

The analog input voltage at absolute accuracy measurement is assumed to be a mid-point of the input voltage width (= 1 LSB) which outputs the same code from the A-D converter with ideal characteristics. For example, when V_{REF} = 5.12 V, the width of 1 LSB is 20 mV. So 0 mV, 20 mV, 40 mV, 60 mV .....or 5120 mV is selected as an analog input voltage.

When the A-D converter is actually used, the analog input voltage range is AV_{SS} to V_{REF}. But if the V_{REF} value is lowered, the accuracy degrades. Every output code for voltage of V_{REF}–V_{CC} is "FF_{16}." Figure 2.6.11 shows the absolute accuracy of the A-D converter. Absolute accuracy = ±2 LSB indicates that when the analog input voltage is 100 mV, the output code expected from the ideal A-D converter is "05_{16}" but the actual A-D conversion result is in the range of "03_{16}" to "07_{16}."

The absolute accuracy includes a zero error and a full-scale error but not a quantization error.

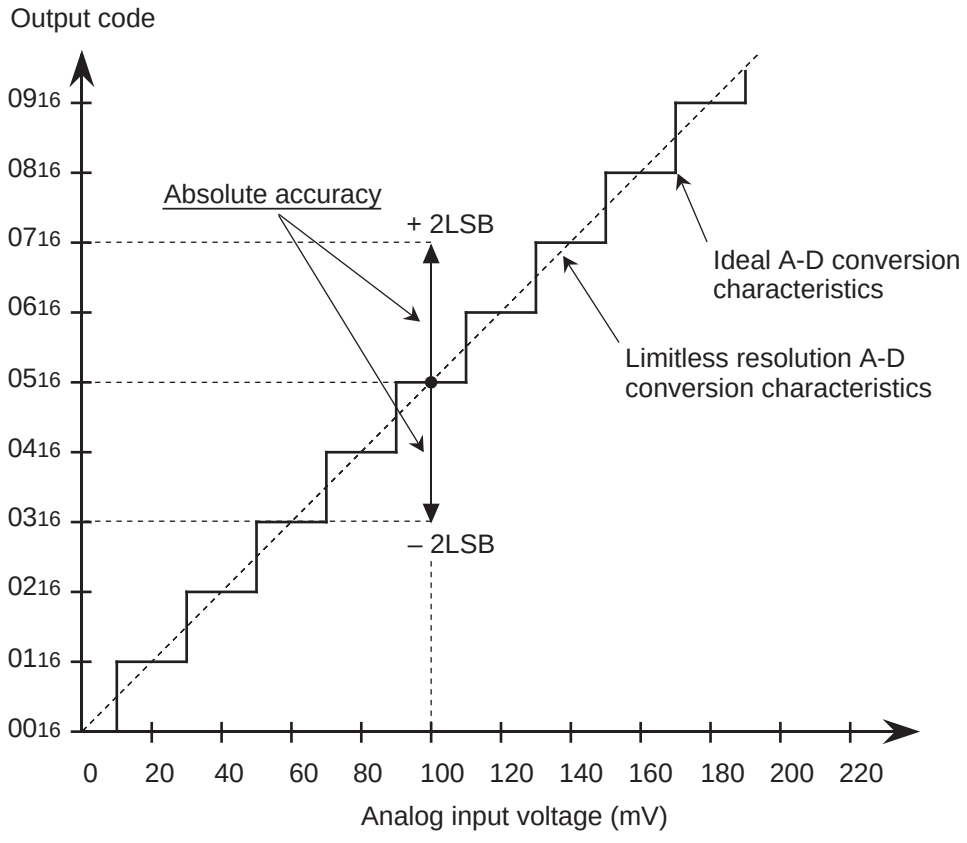


Fig. 2.6.11 Absolute accuracy of A-D converter

(2) Differential non-linearity error

The differential non-linearity error indicates the difference between the analog input voltage width in which the same code is output at actual measurement and the input voltage width (= 1 LSB) which outputs the same output code from the A-D converter with ideal characteristics. For example, when V_{REF} = 5.12 V, the width of 1 LSB is 20 mV. However, when differential non-linearity error = ±1 LSB, the analog input voltage width which outputs the same code is 0 mV to 40 mV.

Figure 2.6.12 shows the differential non-linearity error of the A-D converter.

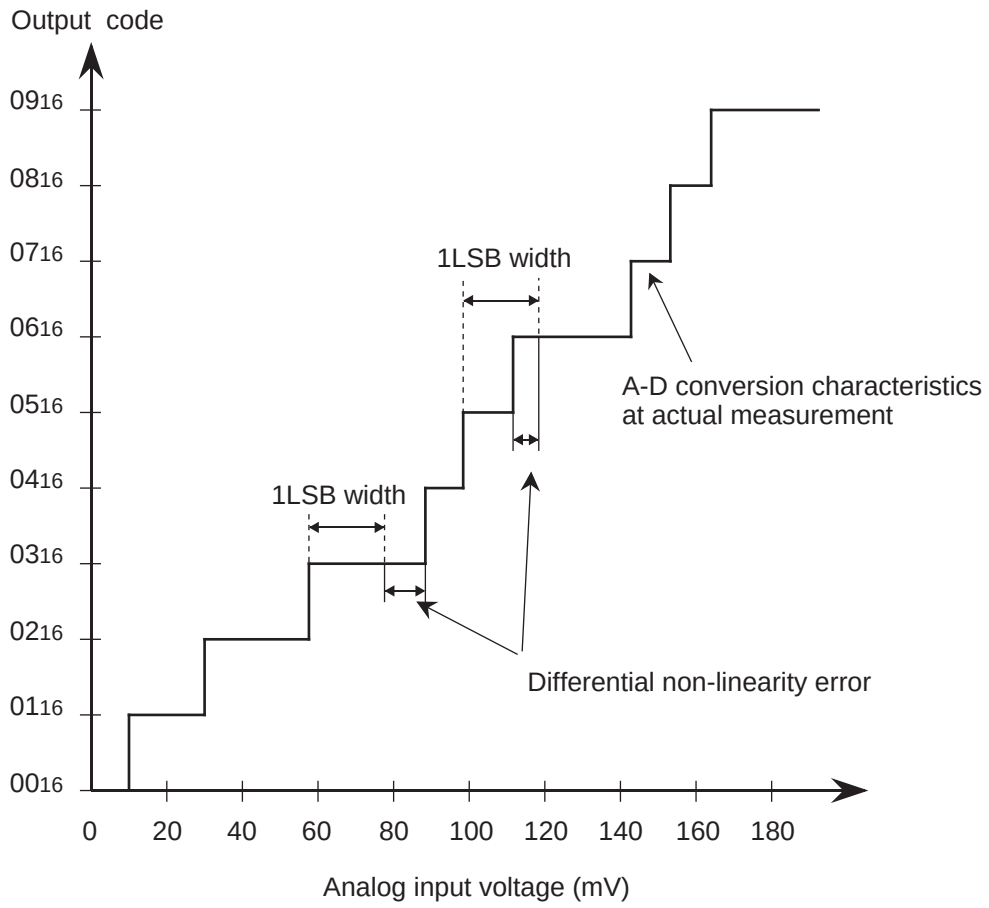


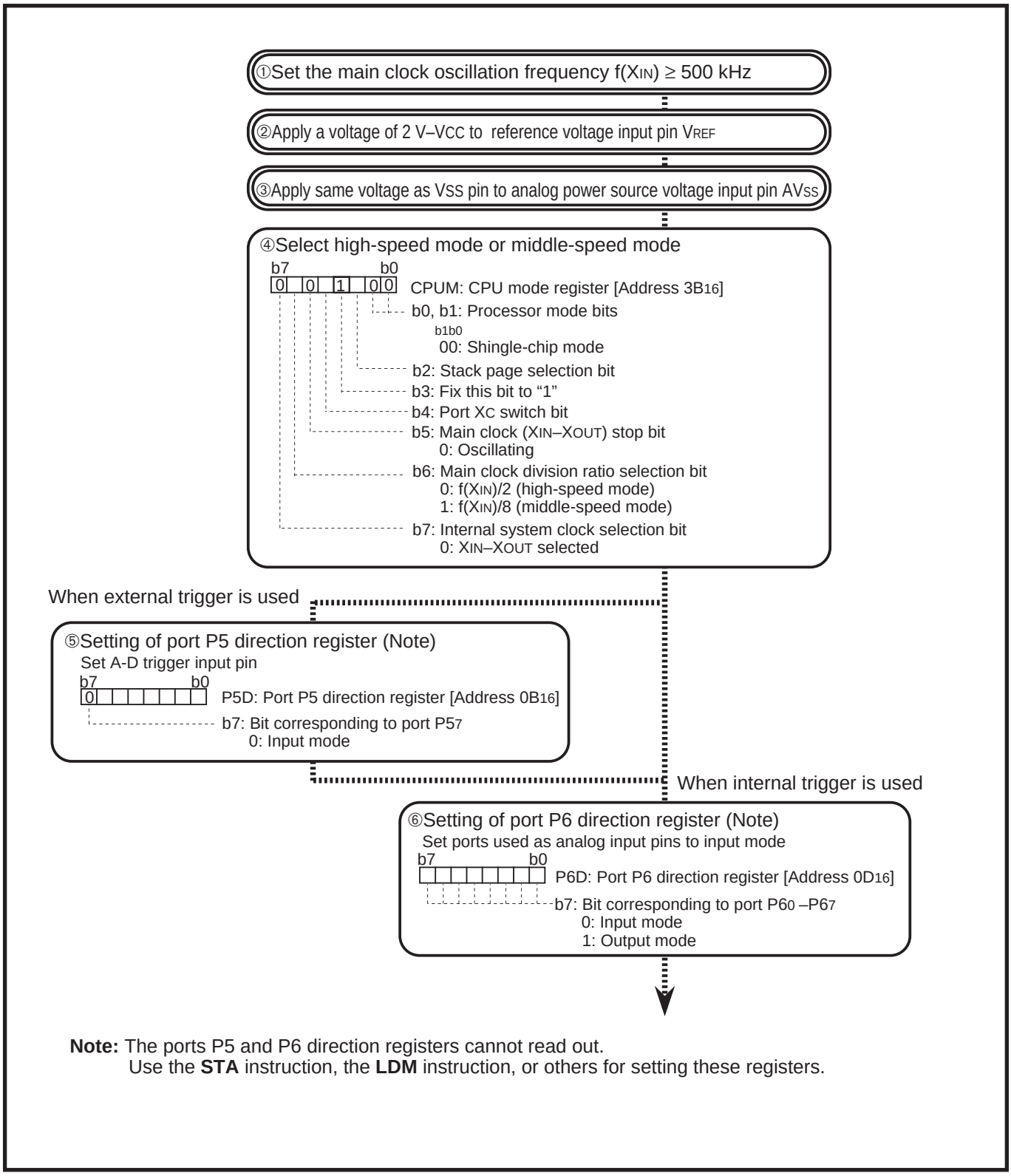
Fig. 2.6.12 Differential non-linearity error of A-D converter

2.6.6 Register setting example

A register setting example when the A-D converter is used is described below.

(1) Operating conditions

To use the A-D converter, first set as shown in Figure 2.6.13.



Note: The ports P5 and P6 direction registers cannot read out.
Use the **STA** instruction, the **LDM** instruction, or others for setting these registers.

Fig. 2.6.13 Operating conditions for using A-D converter

(2) Register initialization example

Figure 2.6.14 and Figure 2.6.15 show a register initialization example when an internal trigger is selected. Figure 2.6.16 and Figure 2.6.17 show a register initialization example when an external trigger is selected.

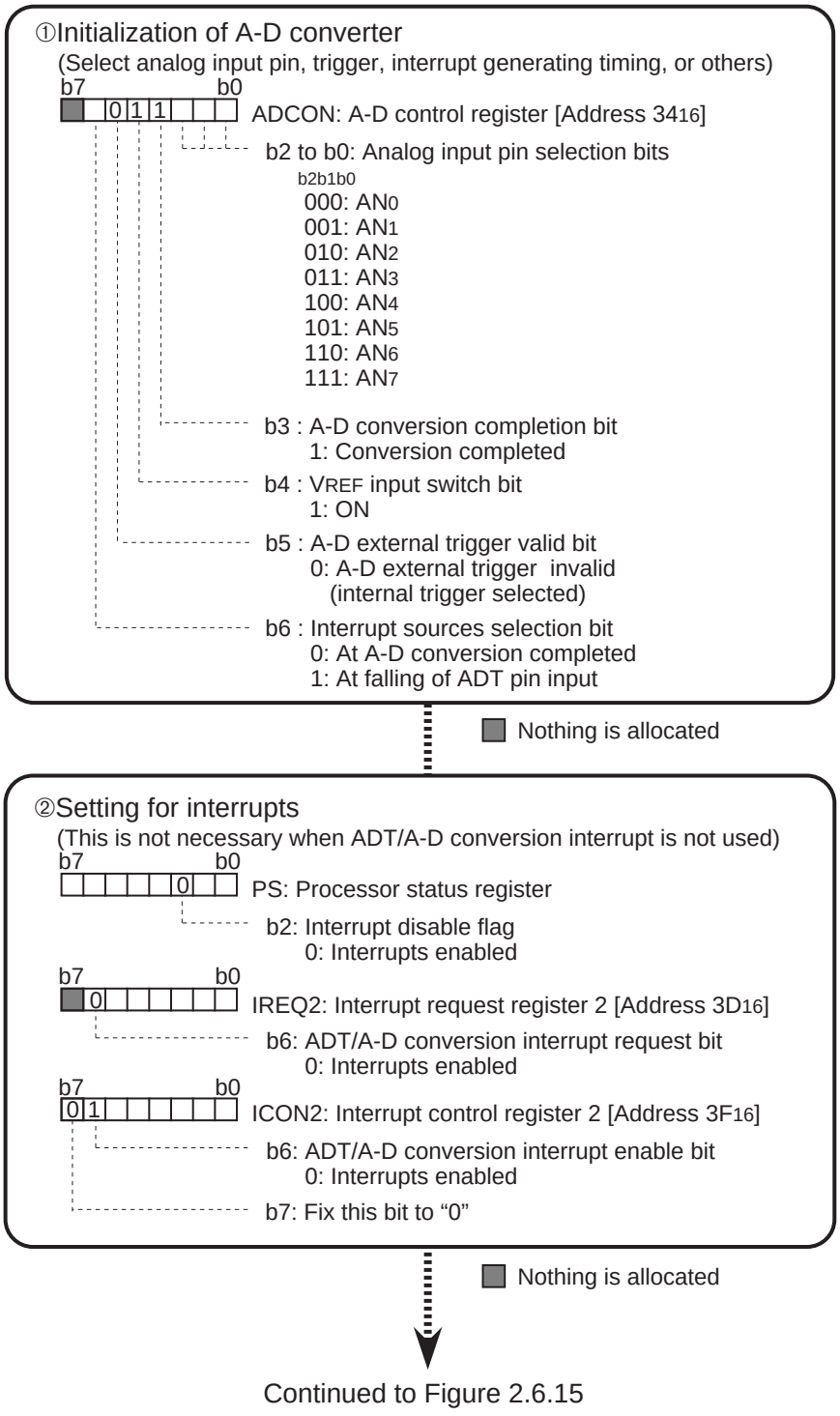


Fig. 2.6.14 Register initialization example when internal trigger is selected (1)

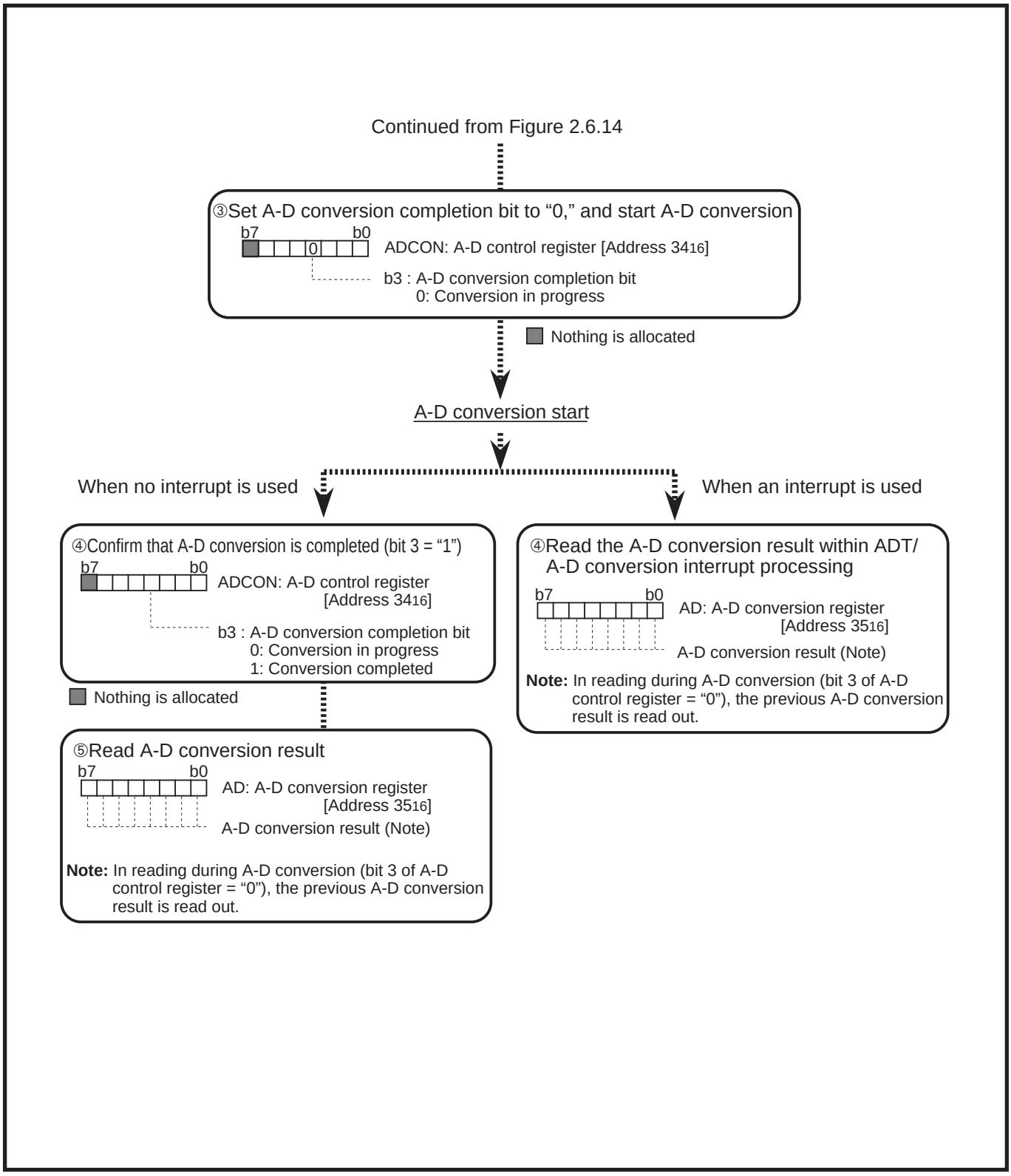


Fig. 2.6.15 Register initialization example when internal trigger is selected (2)

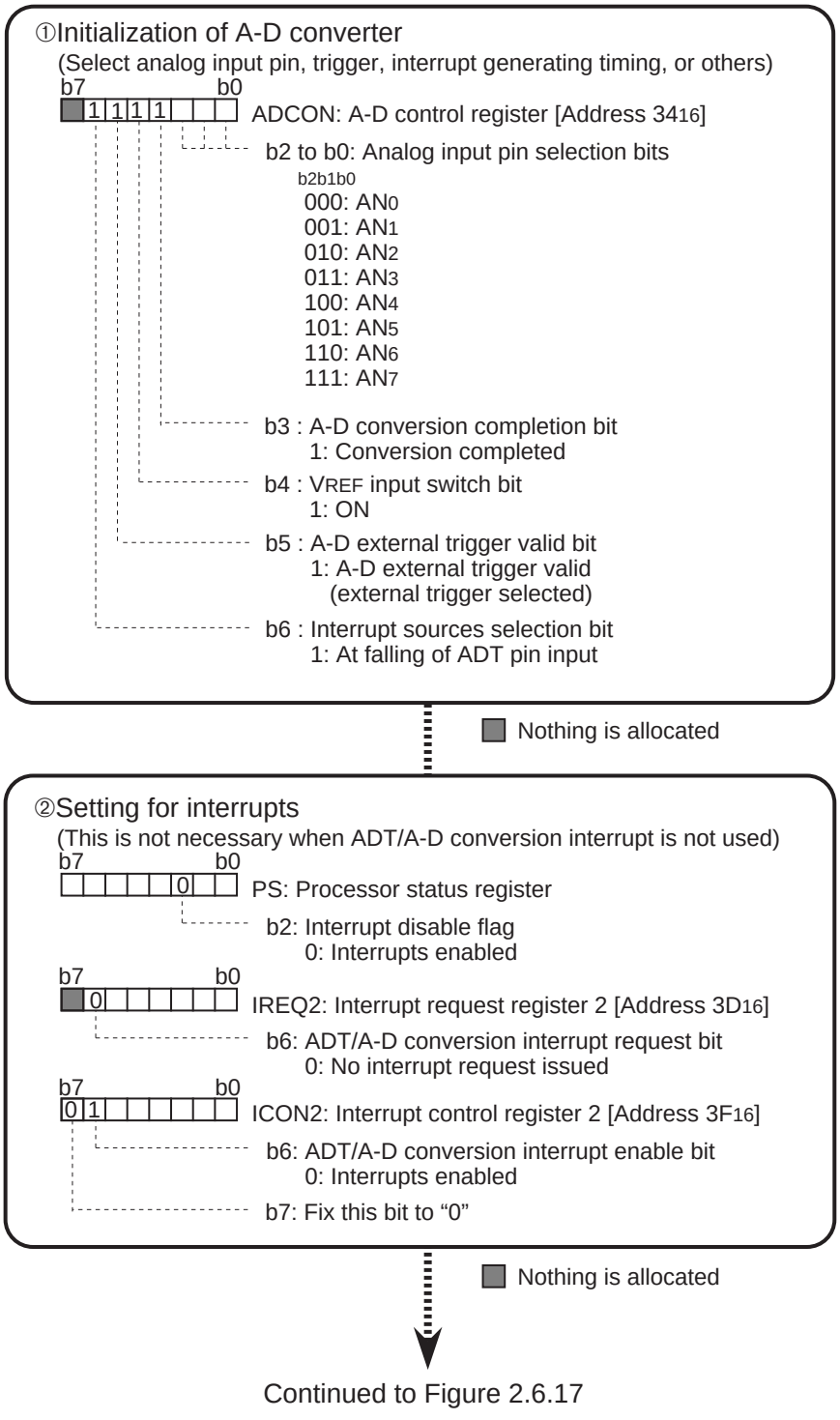


Fig. 2.6.16 Register initialization example when external trigger is selected (1)

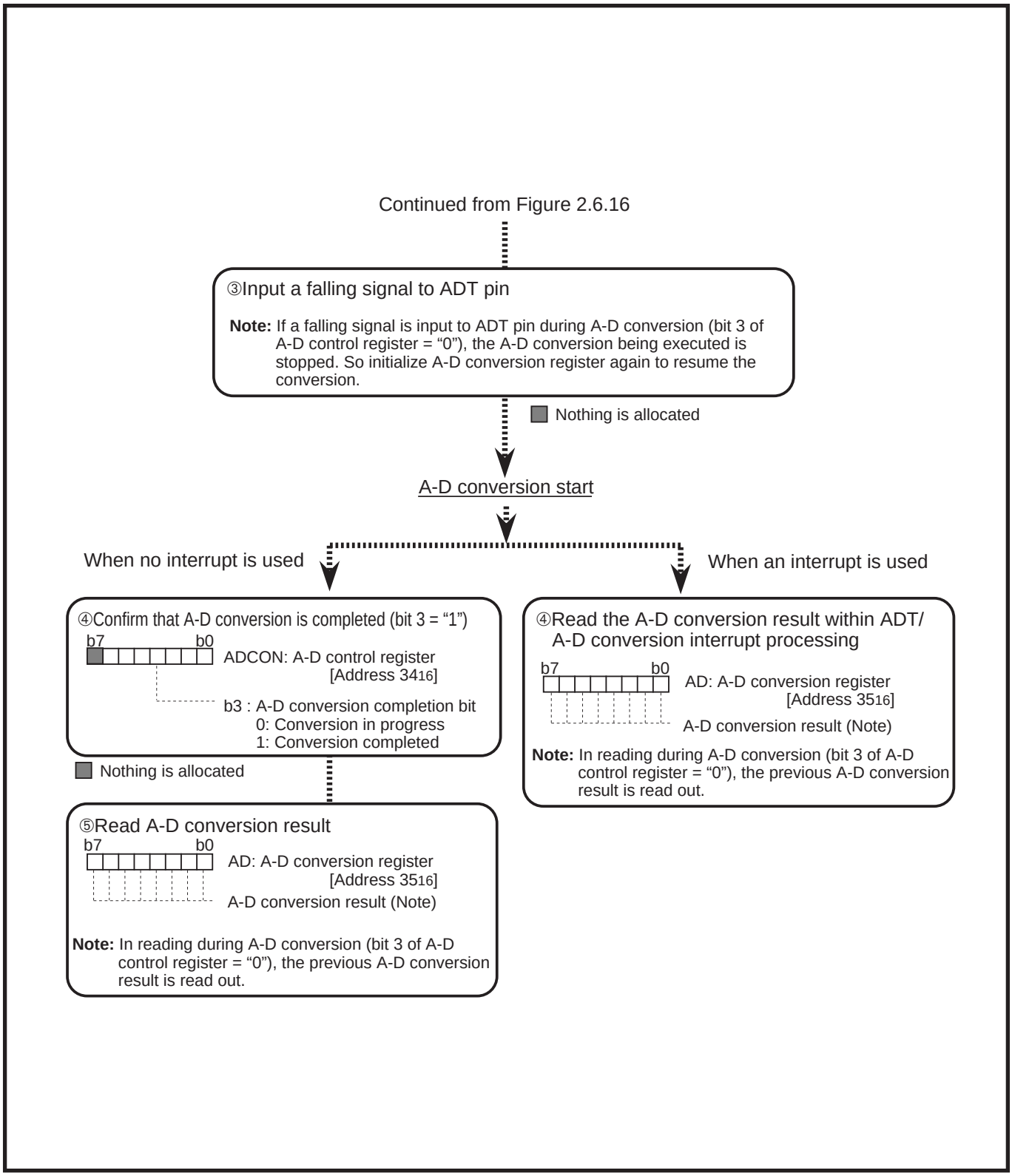


Fig. 2.6.17 Register initialization example when external trigger is selected (2)

2.6.7 Application example: Detection of battery voltage and battery temperature

Outline: The battery voltage and its temperature are detected by using the A-D converter.

Specification: •A-D conversion is performed every second and the data on battery voltage and battery temperature are input.

•With an ADT/A-D conversion interrupt that occurs upon completion of A-D conversion, voltage data or temperature data is input. An analog input pin is also selected.

Figure 2.6.18 shows an example of a peripheral circuit, Figure 2.6.19, setting of related registers, Figure 2.6.20, the control procedure.

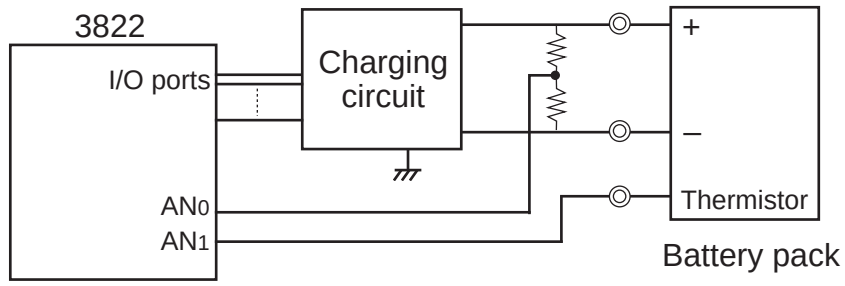


Fig. 2.6.18 Example of peripheral circuit

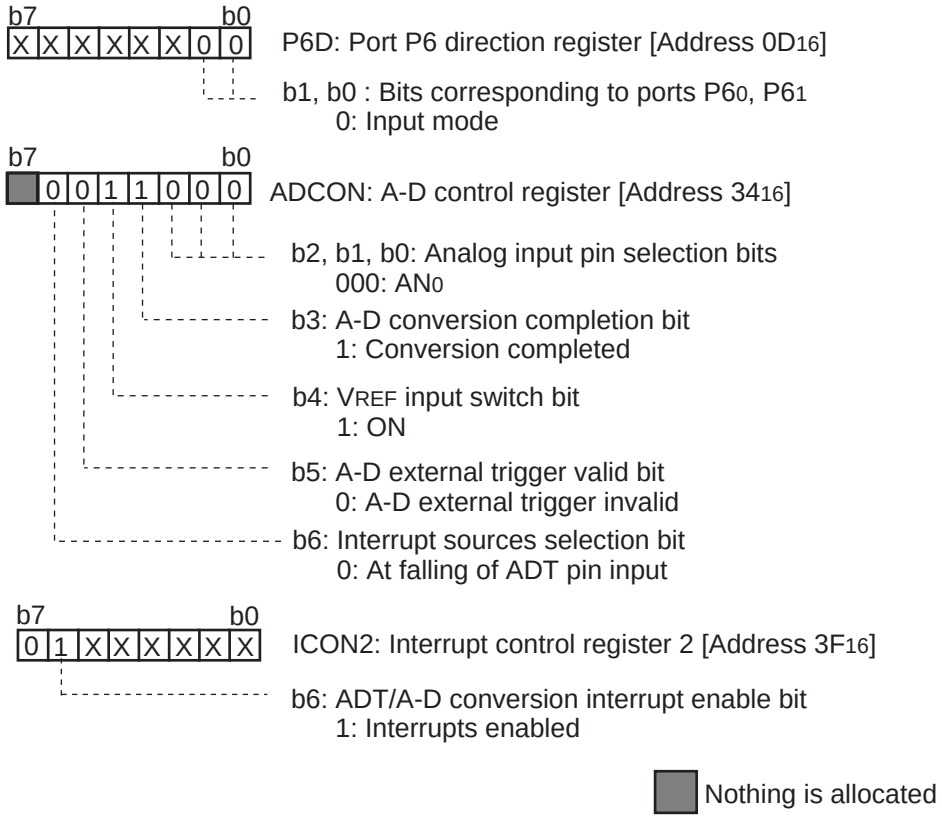


Fig. 2.6.19 Setting of related registers

2.6 A-D converter

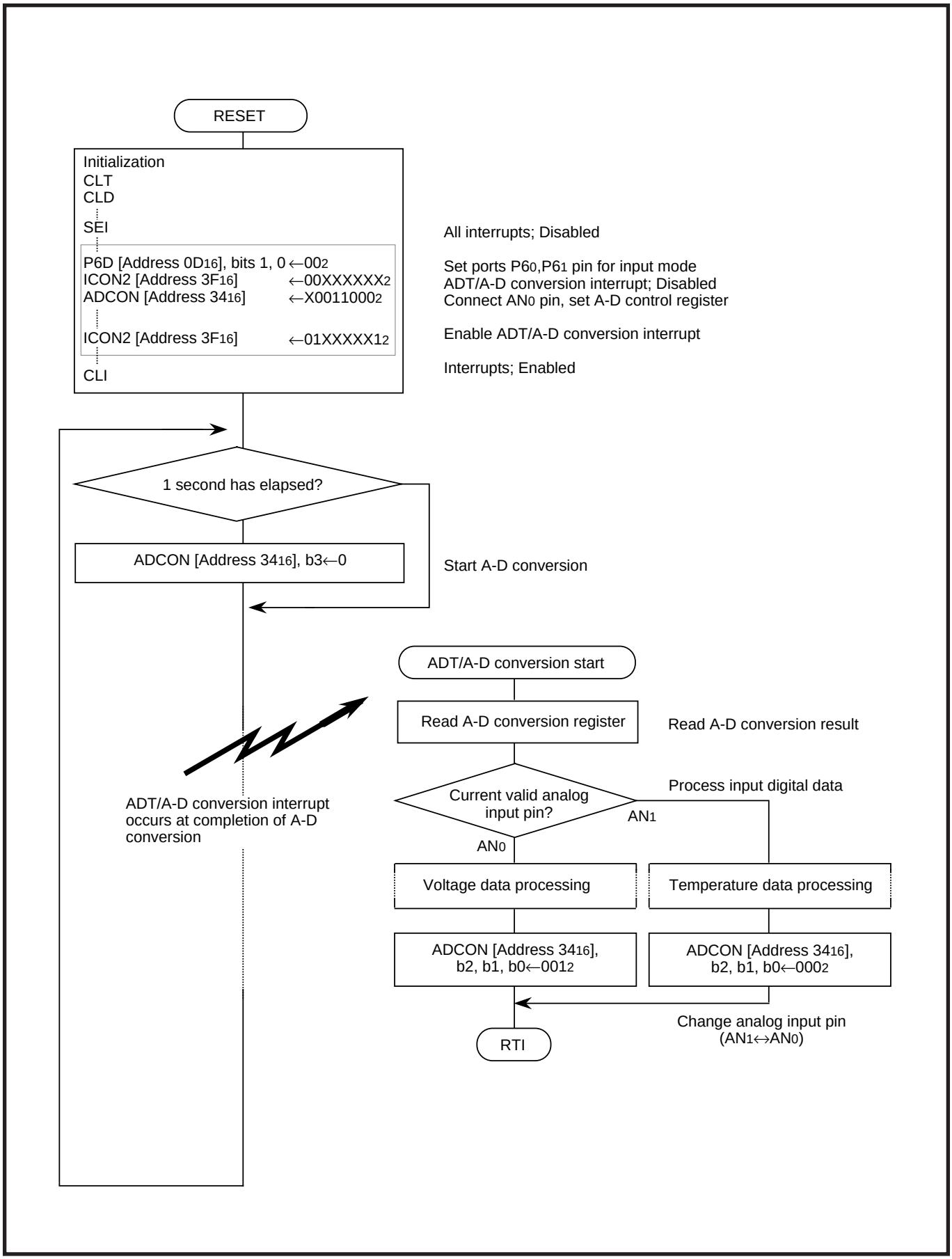


Fig. 2.6.20 Control procedure

2.6.8 Notes on use

When using the A-D converter, notes the following.

(1) Operating conditions for using A-D converter

Operate the A-D converter in the following conditions.

■The comparator is composed of a capacitance coupling. If the oscillation frequency is low, the charge will be lost. Accordingly, make sure that f(X_{IN}) at least 500 kHz during A-D conversion.
Do not execute the **STP** instruction or **WIT** instruction during A-D conversion.

■When an external trigger is selected, the A-D conversion being executed is stopped by inputting a falling signal to the ADT pin during A-D conversion, and A-D conversion is resumed.

■Apply a voltage of 2 V–V_{CC} to the reference voltage input pin V_{REF}. Note that if the V_{REF} value is lowered, the accuracy degrades.

■Apply the same voltage as the V_{SS} pin to the analog power source voltage input pin AV_{SS}.

■Set the port used as an analog input pin for the input mode.
(Corresponding bit of port P6 direction register (address $000D_{16}$) = “0”) (Note)

Note: The port P6 direction register cannot read out. Use the **STA** instruction or **LDM** instruction to set the port P6 direction register.

■When not using the A-D converter, connect the A-D converter power source pin AV_{SS} to V_{SS} line which is the analog system.

(2) Other notes

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 comparison precision to be worse.

2.7 LCD drive control circuit

The 3822 group includes a controller/drivers of Liquid Crystal Display (LCD). This section describes an explanation of LCD control circuit operations, pins, related registers, usage and application examples.

2.7.1 Explanation of operations

(1) LCD drive waveform example

Refer to **"CHAPTER 1 Hardware, LCD drive control circuit."**

(2) LCD drive timing

The frequency of the internal signal LCDCK and the frame frequency to generate LCD drive timing are as follows.

$$f(\text{LCDCK}) = \frac{\text{Count source frequency for LCDCK}}{\text{Division ratio of LCD circuit divider}}$$

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

2.7.2 Pins

SEG0–SEG11 are used as pins for LCD display. The pins P34/SEG12–P37/SEG15 and P00/SEG16–P07/SEG23 and P10/SEG24–P17/SEG31 are available as segment output pins (SEG12–SEG31). By switching the corresponding registers, the segment output pin, I/O pin or input pin is selected.

Table 2.7.1 shows the pin function by setting segment output enable register and Table 2.7.2 shows the pin functions by setting the corresponding registers when they are not used as segment output pins.

Table 2.7.1 Pin functions by setting segment output enable register

Pins	Setting		Pin function
	Register	Value	
P34/SEG12 –P37/SEG15	SEG (Address 003816) b0 (Bit 0 of segment output enable register)	1	Segment output
		0	Input port
P00/SEG16, P01/SEG17	SEG (Address 003816) b1 (Bit 1 of segment output enable register)	1	Segment output
		0	I/O port
P02/SEG18– P07/SEG23	SEG (Address 003816) b2 (Bit 2 of segment output enable register)	1	Segment output
		0	I/O port
P10/SEG24, P11/SEG25	SEG (Address 003816) b3 (Bit 3 of segment output enable register)	1	Segment output
		0	I/O port
P12/SEG26	SEG (Address 003816) b4 (Bit 4 of segment output enable register)	1	Segment output
		0	I/O port
P13/SEG27– P17/SEG31	SEG (Address 003816) b5 (Bit 5 of segment output enable register)	1	Segment output
		0	I/O port

Note: When the microcomputer is in the reset state, the I/O or segment output pins are pulled down, so that a “L” level is output from segment-only pins.

2.7 LCD drive control circuit

Table 2.7.2 Pin functions by setting the corresponding registers when they are not used as segment output pins

Ports	Setting		Pin function
	Register	Value	
$P3_4$–$P3_7$	PULLA (Address 0016_{16}) b3 (Bit 3 of PULL register A)	1	Pull-down pin
		0	Input port
$P0_0$–$P0_7$	P0D (Address 0001_{16}) b0 (Bit 0 of port P0 direction register)	1	Output port
		0	Input port
	PULLA (Address 0016_{16}) b0 (Bit 0 of PULL register A)	1	Pull-down pin (Valid when bit 0 of port P0 direction register is "0")
		0	No pull-down
$P1_0$–$P1_7$	P1D (Address 0003_{16}) b0 (Bit 0 of port P1 direction register)	1	Output port
		0	Input port
	PULLA (Address 0016_{16}) b1 (Bit 1 of PULL register A)	1	Pull-down pin (Valid when bit 0 of port P1 direction register is "0")
		0	No pull-down

(1) Segment output pins (SEG_0–SEG_{31})

Up to 32 segment outputs can be selected. Table 2.7.3 shows setting of segment output pins for LCD display.

Table 2.7.3 Setting of segment output pins for LCD display

Pins	Setting
SEG_0–SEG_{11}	Segment output-only pin
$P3_4/SEG_{12}$–$P3_7/SEG_{15}$	Ports $P3_4$–$P3_7$ are used as segment signal output pins (SEG_{12}–SEG_{15}) by setting bit 0 of the segment output enable register (address 0038_{16}) to "1."
$P0_0/SEG_{16}$, $P0_1/SEG_{17}$	Ports $P0_0$ and $P0_1$ are used as segment signal output pins (SEG_{16}, SEG_{17}) by setting bit 1 of the segment output enable register (address 0038_{16}) to "1."
$P0_2/SEG_{18}$–$P0_7/SEG_{23}$	Ports $P0_2$–$P0_7$ are used as segment signal output pins (SEG_{18}–SEG_{23}) by setting bit 2 of the segment output enable register (address 0038_{16}) to "1."
$P1_0/SEG_{24}$, $P1_1/SEG_{25}$	Ports $P1_0$ and $P1_1$ are used as segment signal output pins (SEG_{24}, SEG_{25}) by setting bit 3 of the segment output enable register (address 0038_{16}) to "1."
$P1_2/SEG_{26}$	Ports $P1_2$ is used as segment signal output pins (SEG_{26}) by setting bit 4 of the segment output enable register (address 0038_{16}) to "1."
$P1_3/SEG_{27}$–$P1_7/SEG_{31}$	Ports $P1_3$–$P1_7$ are used as segment signal output pins (SEG_{27}–SEG_{31}) by setting bit 5 of the segment output enable register (address 0038_{16}) to "1."

(2) Ports P0, P1 and $P3_4$–$P3_7$

When pins $P3_4/SEG_{12}$–$P3_7/SEG_{15}$, $P0_0/SEG_{16}$–$P0_7/SEG_{23}$, $P1_0/SEG_{24}$–$P1_7/SEG_{31}$ are not used as segment outputs, they can be used as input ports $P3_4$–$P3_7$ and as I/O ports P0 and P1.

Table 2.7.4 shows the setting of input ports $P3_4$–$P3_7$ and I/O ports P0, P1.

Table 2.7.4 Setting of input ports $P3_4$–$P3_7$ and I/O ports P0, P1

Ports	Setting
$P3_4$–$P3_7$	By setting bit 0 of segment output enable register (address 0038_{16}) to "0"
$P0_0$, $P0_1$	By setting bit 1 of segment output enable register (address 0038_{16}) to "0"
$P0_2$–$P0_7$	By setting bit 2 of segment output enable register (address 0038_{16}) to "0"
$P1_0$, $P1_1$	By setting bit 3 of segment output enable register (address 0038_{16}) to "0"
$P1_2$	By setting bit 4 of segment output enable register (address 0038_{16}) to "0"
$P1_3$–$P1_7$	By setting bit 5 of segment output enable register (address 0038_{16}) to "0"

(3) $P3_4$–$P3_7$, P0 and P1 pull-down pins

When pins $P3_4/SEG_{12}$–$P3_7/SEG_{15}$, $P0_0/SEG_{16}$–$P0_7/SEG_{23}$, $P1_0/SEG_{24}$–$P1_7/SEG_{31}$ are not used as ports, it is possible to exert pull-down control. Table 2.7.5 shows the setting of pull-down pins.

Table 2.7.5 Setting of pull-down pins

Pins	Setting
$P3_4/SEG_{12}$–$P3_7/SEG_{15}$	By setting bit 0 of the segment output enable register (address 0038_{16}) to "0," then setting bit 3 of PULL register A (address 0016_{16}) to "1."
$P0_0/SEG_{16}$–$P0_7/SEG_{23}$	By setting bits 1 and 2 of the segment output enable register (address 0038_{16}) to "0," next setting bit 0 of the port P0 direction register (address 0001_{16}) to "0," then setting bit 0 of PULL register A (address 0016_{16}) to "1."
$P1_0/SEG_{24}$–$P1_7/SEG_{31}$	By setting bits 3 to 5 of the segment output enable register (address 0038_{16}) to "0," next setting bit 1 of the port P1 direction register (address 0003_{16}) to "0," then setting bit 1 of PULL register A (address 0016_{16}) to "1."

2.7 LCD drive control circuit

2.7.3 Related registers

Figure 2.7.1 shows the memory allocation of LCD display-related registers.

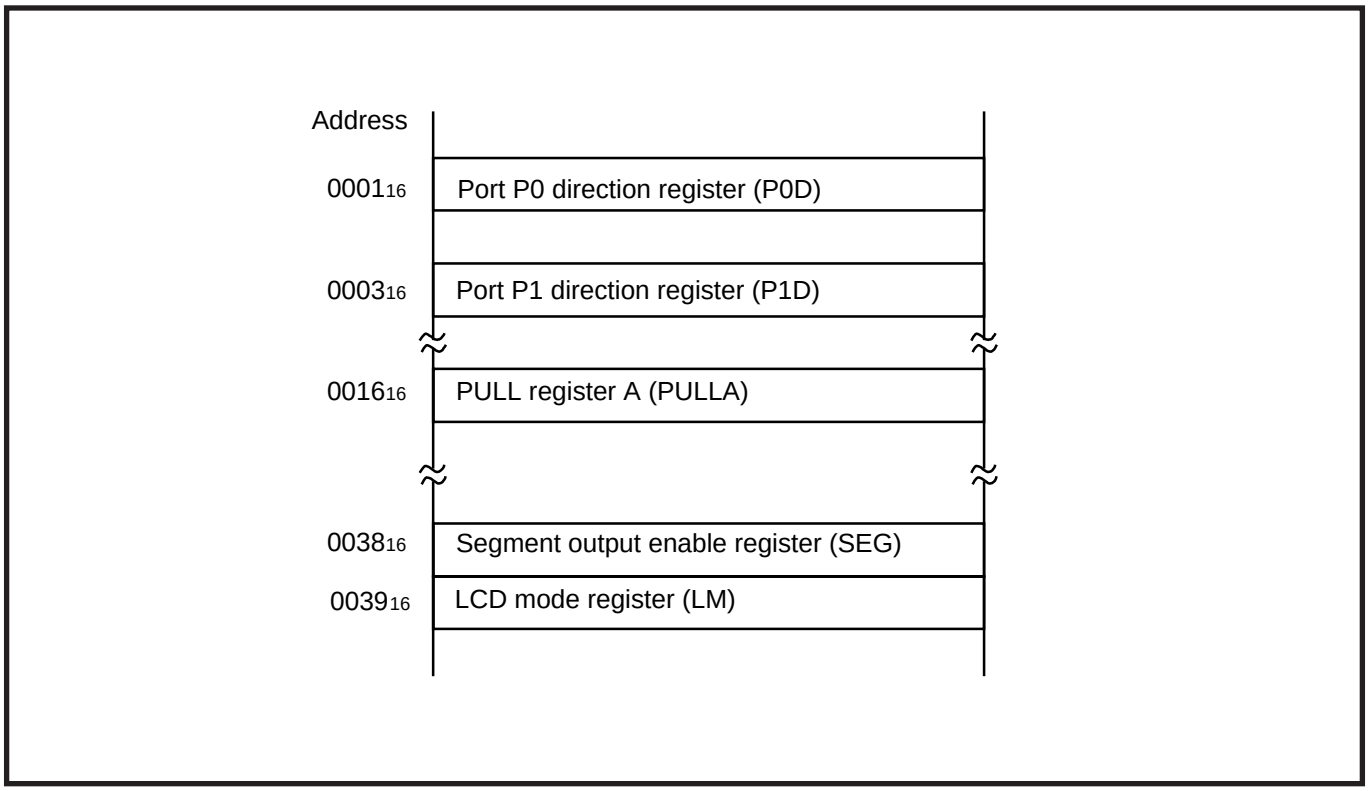


Fig. 2.7.1 Memory allocation of LCD display-related registers

(1) Segment output enable register (address 0038_{16})

The pins $P3_4$/SEG_{12}–$P3_7$/SEG_{15}, $P0_0$/SEG_{16}–$P0_7$/SEG_{23}, $P1_0$/SEG_{24}–$P1_7$/SEG_{31} can be used as segment output pins by setting bits 0 to 5 of the segment output enable register (address 0038_{16}).

The pins corresponding to the bits which are set to "1" among bits 0 to 5 of the segment output enable register (address 0038_{16}) are used as segment output pins. The pins corresponding to the bits which are set to "0" are used as I/O ports or input ports.

Figure 2.7.2 shows the structure of the segment output enable register.

Segment output enable register

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Segment output enable register (SEG) [Address 38_{16}]

B	Name	Functions	At reset	R	W
0	Segment output enable bit 0	0: Input ports $P3_4$–$P3_7$ 1: Segment output SEG_{12}–SEG_{15}	0	○	○
1	Segment output enable bit 1	0: I/O ports $P0_0$, $P0_1$ 1: Segment output SEG_{16}, SEG_{17}	0	○	○
2	Segment output enable bit 2	0: I/O ports $P0_2$–$P0_7$ 1: Segment output SEG_{18}–SEG_{23}	0	○	○
3	Segment output enable bit 3	0: I/O ports $P1_0$, $P1_1$ 1: Segment output SEG_{24}, SEG_{25}	0	○	○
4	Segment output enable bit 4	0: I/O ports $P1_2$ 1: Segment output SEG_{26}	0	○	○
5	Segment output enable bit 5	0: I/O ports $P1_3$–$P1_7$ 1: Segment output SEG_{27}–SEG_{31}	0	○	○
6,7	Fix these bits to "0."		0	0	0

Fig. 2.7.2 Structure of segment output enable register

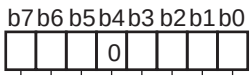
2.7 LCD drive control circuit

(2) LCD mode register (address 0039_{16})

The LCD mode register controls various functions of the LCD controller/driver. Figure 2.7.3 shows the structure of the LCD mode register.

- ●Bits 0, 1 : Duty ratio selection bits
  Select a duty ratio number fit for the LCD panel used.
- ●Bit 2 : Bias control bit
  Select a bias value fit for the LCD panel used.
- ●Bit 3 : LCD enable bit
  Turns on and off the LCD. When this bit is set to "1," the bits which are set to "1" in the LCD display RAM are displayed on the LCD. When this bit is set to "0," the whole LCD display is turned off.
- ●Bit 4 : Unused
  Always set this bit to "0."
- ●Bits 5, 6 : LCD circuit divider division ratio selection bits
  Used to select a division ratio for generating the frequency of the LCDCK, which is the clock for the LCD timing controller. Select a division ratio so as to generate LCDCK fit for the LCD panel used.
- ●Bit 7 : LCDCK count source selection bit
  Used to select a count source of the above LCDCK. At transition from the high-speed, middle-speed or low-speed mode to the low-power operation, or others, change the count source as required.

LCD mode register



LCD mode register (LM) [Address 39_{16}]

B	Name	Functions	At reset	R	W
0	Duty ratio selection bits	b1b0 00: Not available 01: 2 (use COM_0, COM_1) 10: 3 (use COM_0–COM_2) 11: 4 (use COM_0–COM_3)	0	○	○
1			0	○	○
2	Bias control bit	0: 1/3 bias 1: 1/2 bias	0	○	○
3	LCD enable bit	0: LCD OFF 1: LCD ON	0	○	○
4	Fix this bit to "0."		0	0	0
5	LCD circuit divider division ratio selection bits (Note 1)	b6b5 00: LCDCK count source 01: LCDCK count source/2 10: LCDCK count source/4 11: LCDCK count source/8	0	○	○
6					
7	LCDCK count source selection bit (Note 2)	0: $f(X_{CIN})/32$ 1: $f(X_{IN})/8192$	0	○	○

Notes 1: Reference values at $f(X_{IN})$ = 8 MHz
00: 977 Hz
01: 488 Hz
10: 244 Hz
11: 122 Hz
2: LCDCK is a clock for a LCD timing controller.

Fig. 2.7.3 Structure of LCD mode register

2.7 LCD drive control circuit

(3) Port P0 direction register (address 0001_{16})

When it is specified that pins $P0_0$/SEG_{16}–$P0_7$/SEG_{23} are used as I/O ports by bits 1 and 2 of the segment output enable register (address 0038_{16}), the setting of the port P0 direction register is valid. When bit 0 of the port P0 direction register (address 0001_{16}) is set to "1," port P0 is an output port. When this bit is set to "0," the port is an input port, so that the setting of bit 0 of the PULL register A (address 0016_{16}) becomes valid. At reset, bit 0 of the port P0 direction register is set to "0."
Figure 2.7.4 shows the structure of the port P0 direction register.

Port P0 direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port P0 direction register (P0D) [Address 01 16]

B	Name	Functions	At reset	R	W
0	Port P0 direction register	0: All bits are input mode 1: All bits are output mode	0	×	○
1 to 7	Nothing is allocated. These bits cannot be written to and be read out.		0	×	×

Fig. 2.7.4 Structure of port P0 direction register

(4) Port P1 direction register (address 0003_{16})

When it is specified that pins $P1_0$/SEG_{24}–$P1_7$/SEG_{31} are used as I/O ports by bits 3 to 5 of the segment output enable register (address 0038_{16}), the setting of the port P1 direction register is valid. When bit 0 of the port P1 direction register (address 0003_{16}) is set to "1," port P1 is an output port. When this bit is set to "0," the port is an input port, so that the setting of bit 1 of the PULL register A (address 0016_{16}) becomes valid. At reset, bit 0 of the port P1 direction register is set to "0." Figure 2.7.5 shows the structure of the port P1 direction register.

Port P1 direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port P1 direction register (P1D) [Address 03 16]

B	Name	Functions	At reset	R	W
0	Port P1 direction register	0: All bits are input mode 1: All bits are output mode	0	×	○
1 to 7	Nothing is allocated. These bits cannot be written to and be read out.		0	×	×

Fig. 2.7.5 Structure of port P1 direction register

2.7 LCD drive control circuit

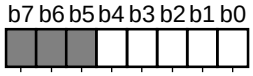
(5) PULL register A (address 0016_{16})

When ports P0 and P1 are set for the input mode, the setting of bits 0 and 1 of the PULL register A is valid.

The pull-down function of ports P0, P1 and $P3_4$–$P3_7$ is made effective by setting bits 0, 1 and 3 of the PULL register A to "1." When ports P0 and P1 are set for output mode by bit 0 of the port P0/P1 direction registers, the setting of the PULL register A is invalid.

Figure 2.7.6 shows the structure of the PULL register A.

PULL register A



PULL register A (PULLA) [Address 16_{16}]

B	Name	Function	At reset	R	W
0	Port $P0_0$–$P0_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
1	Port $P1_0$–$P1_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
2	Port $P2_0$–$P2_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
3	Port $P3_0$–$P3_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
4	Port $P7_0$, $P7_1$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
5 to 7	Nothing is allocated. These bits cannot be written to and are fixed to "0" at reading.		0	0	×

Note: For ports set for the output mode, pull-up or pull-down is impossible.

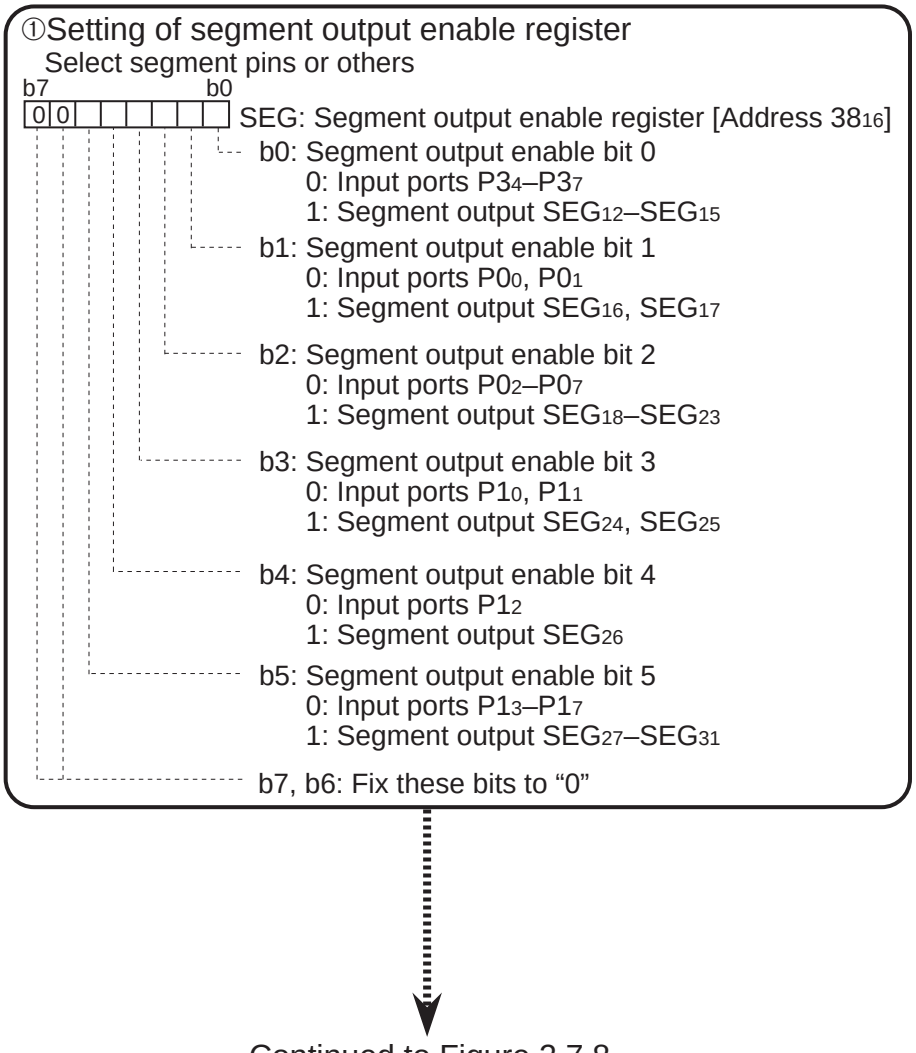
Fig. 2.7.6 Structure of PULL register A

2.7.4 Register setting example

Figure 2.7.7 and Figure 2.7.8 show an example of setting registers for LCD display.

[Note on use]

Note : For pulling down ports P0, P1 and $P3_4$–$P3_7$, refer to **"2.7.2 Pins, (3) Ports $P3_4$–$P3_7$, P0 and P1 pull-down pins"**



Continued to Figure 2.7.8

Fig. 2.7.7 Example of setting registers for LCD display (1)

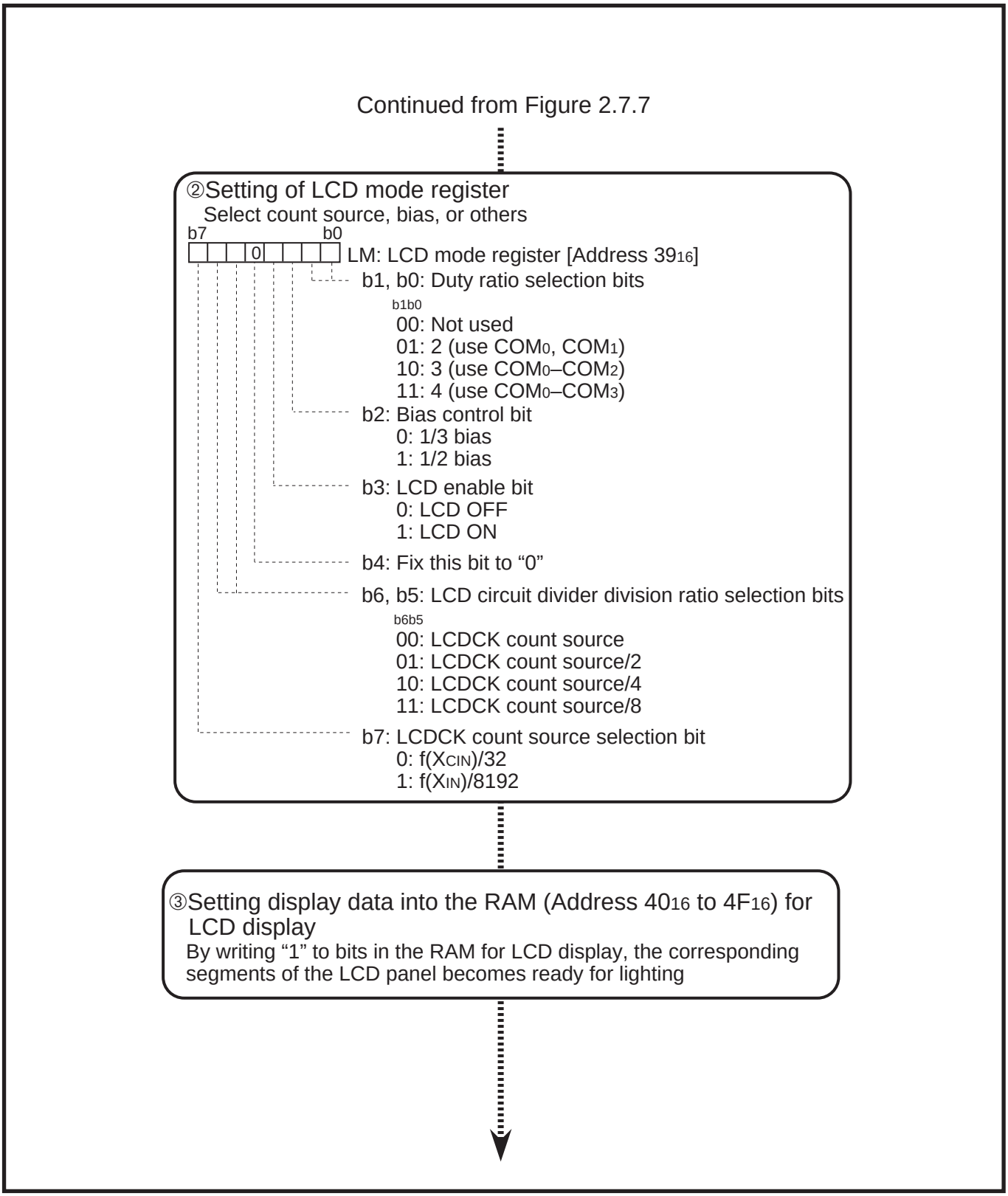


Fig. 2.7.8 Example of setting registers for LCD display (2)

2.7.5 Application examples

(1) LCD panel display pattern example

Figure 2.7.9 shows an 8-segment LCD panel display pattern example when the duty ratio number is 4.

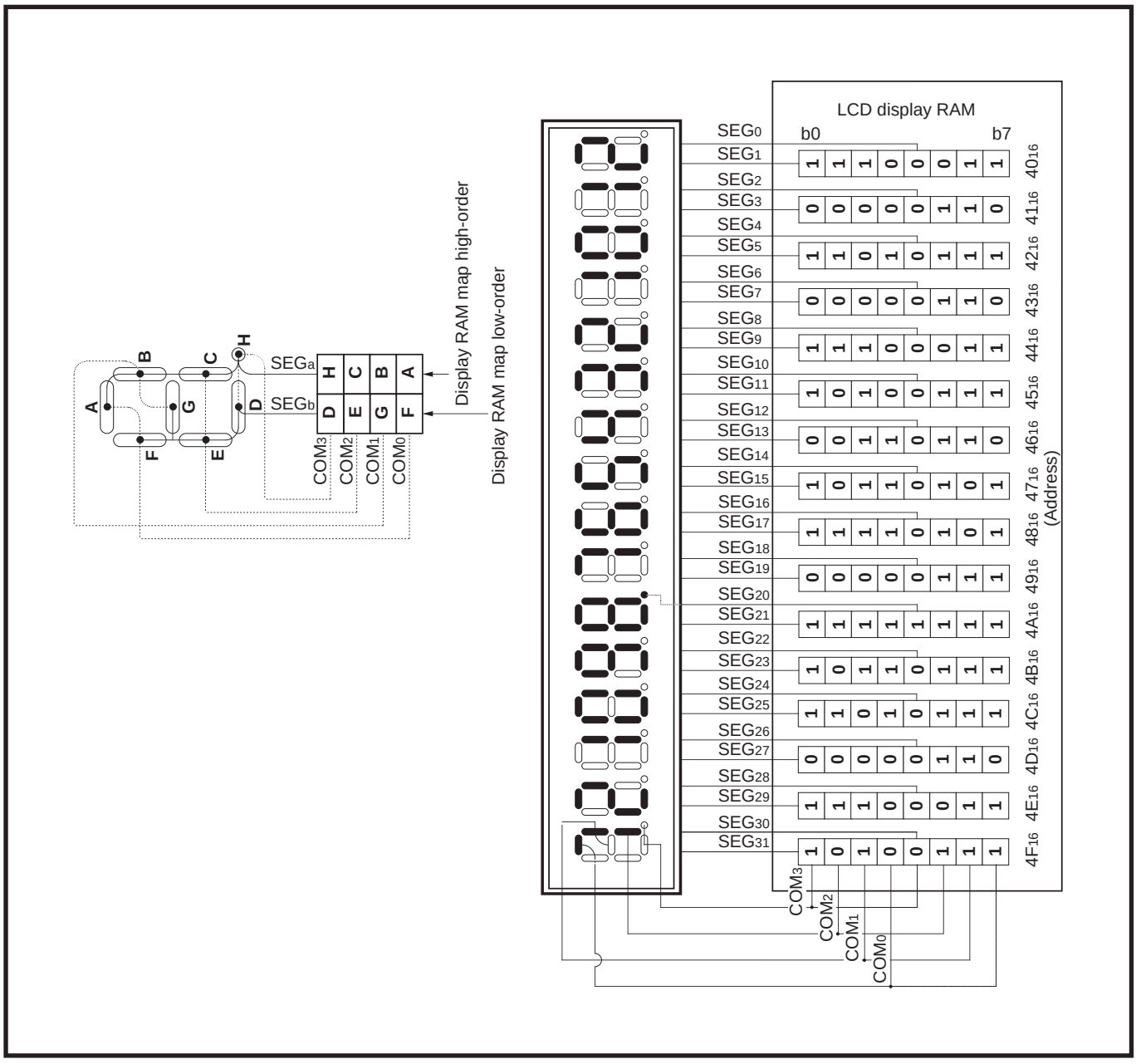


Fig. 2.7.9 8-segment LCD panel display pattern example when duty ratio number is 4

(2) LCD panel example

Figure 2.7.10 to Figure 2.7.12 show an LCD panel example and a segment allocation example for it, and an LCD display RAM setting example.

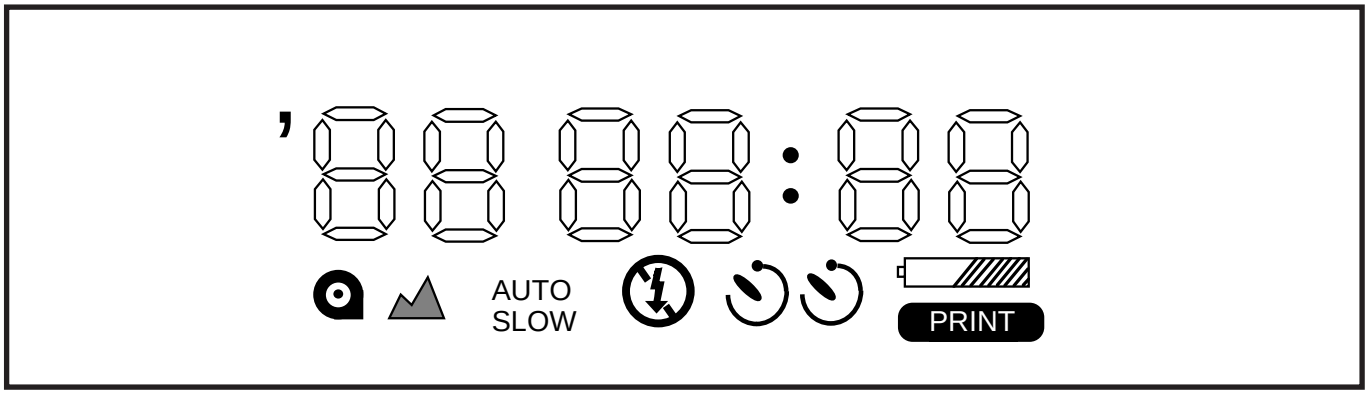


Fig. 2.7.10 LCD panel example

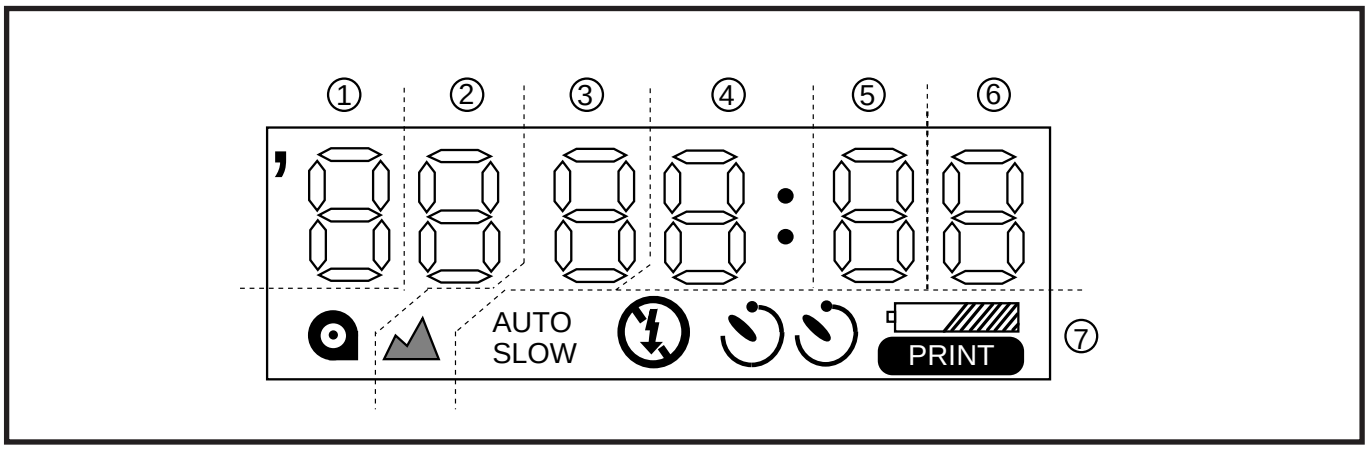


Fig. 2.7.11 Segment allocation example

Bit / Address	7	6	5	4	3	2	1	0
	COM_3	COM_2	COM_1	COM_0	COM_3	COM_2	COM_1	COM_0
0040_{16}	,	g	f	e	d	c	b	a
0041_{16}		g	f	e	d	c	b	a
0042_{16}		g	f	e	d	c	b	a
0043_{16}	:	g	f	e	d	c	b	a
0044_{16}		g	f	e	d	c	b	a
0045_{16}		g	f	e	d	c	b	a
0046_{16}	SLOW	AUTO					PRINT	

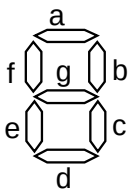
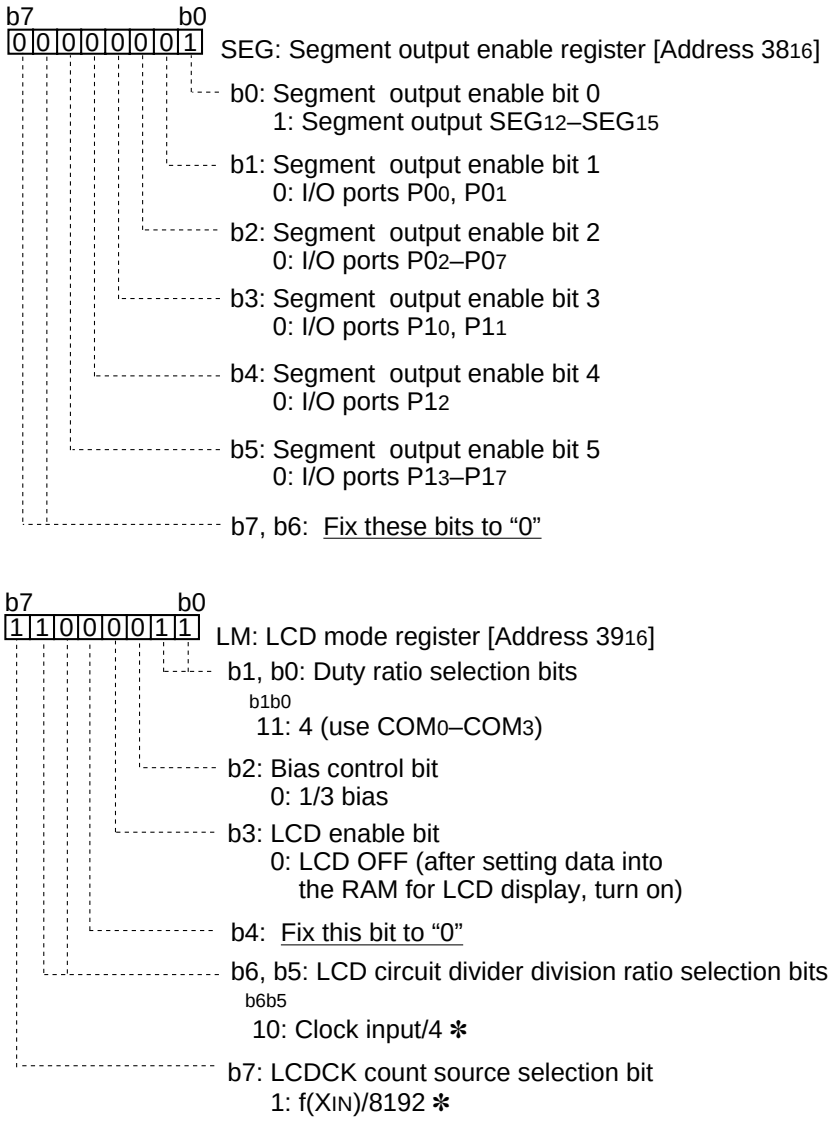


Fig. 2.7.12 LCD display RAM setting example

(3) Control procedure

Figure 2.7.13 shows the setting of related registers to turn on all the LCD display in Figure 2.7.10, and Figure 2.7.14 shows the control procedure.

Specifications: •Frame frequency = 61 Hz
•Duty ratio number = 4, Bias value = 1/3
•Segment output; SEG_0 to SEG_{13} are used.
•Ports P0 and P1 are set as I/O ports.



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

$$\text{Frame frequency } f = \frac{8 \times 10^6}{8192} \times \frac{1}{4} \times \frac{1}{4} \approx 61.035 \text{ Hz}$$

Fig. 2.7.13 Setting of related registers

2.7 LCD drive control circuit

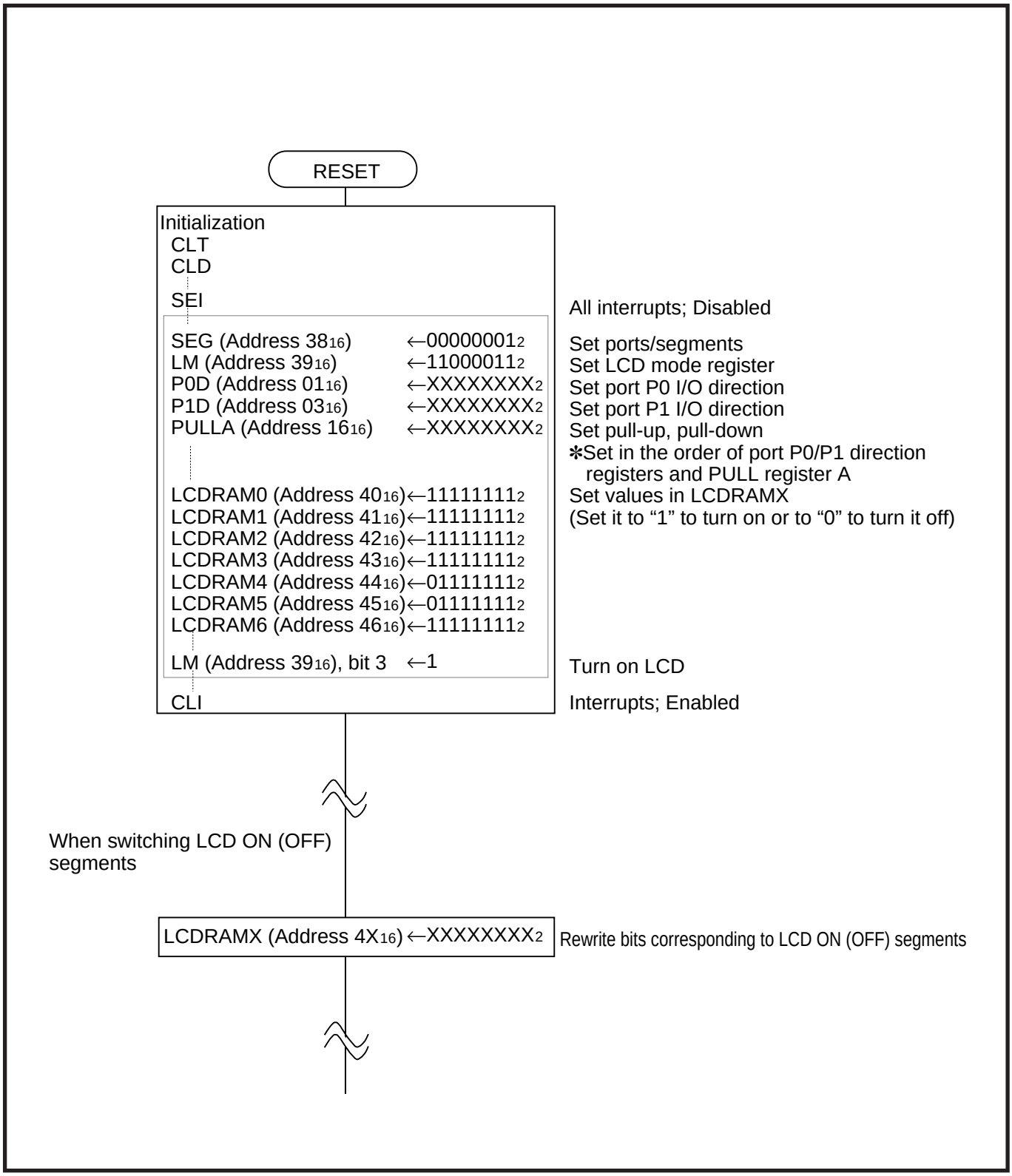


Fig. 2.7.14 Control procedure

2.7.6 Notes on use

(1) For transition from the high-speed or the middle-speed mode to the low-power operation of the low-speed mode:

① Select oscillation at 32 kHz (CM_4 = 1)
② Count source for LCDCK; select f(X_{CIN})/32 (LM_7 = 0)
③ Internal system clock; select X_{CIN}–X_{COUT} (CM_7 = 1)
④ Stop main clock X_{IN}–X_{OUT} (CM_5 = 1)
In the above order, execute transition. Execute the setting ② after the oscillation at 32 kHz (setting ①) becomes completely stable.

(2) If the **STP** instruction is executed while the LCD is turned on by setting bit 3 of the LCD mode register to "1," a DC voltage is applied to the LCD. For this reason, do not execute the **STP** instruction while the LCD is lighting.

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

2.8 Standby function

The 3822 group is provided with a standby function to stop the CPU by software and put the CPU into the low-power operation.
The following two types of standby function are available.
•Stop mode by the **STP** instruction
•Wait mode by the **WIT** instruction

2.8.1 Stop mode

The stop mode is set by executing the **STP** instruction. In the stop mode, the oscillation of both XIN and XCIN stops 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 the stop mode

The stop mode is set by executing the **STP** instruction.*1
In the stop mode, the oscillation of both XIN and XCIN stops, so that all the functions stop, providing a low-power operation.
Table 2.8.1 shows the state in the stop mode.

*1: After setting the LCD enable bit (bit 3) of the LCD mode register to "0," execute the **STP** instruction.

Table 2.8.1 State in the stop mode

Item	State in stop mode
Oscillation	Stop
CPU	Stop
Internal clock ϕ	Stop at "H" level
I/O ports P0–P7	The state where **STP** instruction is executed is held
Timer, serial I/O, LCD display functions	Stop

(2) Release of stop mode

The stop mode is released by reset input or by the occurrence of an interrupt request.
There is a difference in restore processing from the stop mode by reset input and by an interrupt request.

■Restoration by reset input

By holding the "L" input level of the $\overline{\text{RESET}}$ pin in the stop mode for 2 μs or more, the reset state is set, so that the stop mode is released.
At the time when the stop mode is released, oscillation is started. At this time, the inside of the microcomputer is in the reset state. After the input level of the $\overline{\text{RESET}}$ pin is returned to the "H," the reset state is released in approximately 8,000 cycles of the X_{IN} input.
The oscillation is unstable at start of oscillation. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) is required. The time to hold the internal reset state is reserved as the oscillation stabilizing time.
Figure 2.8.1 shows the oscillation stabilizing time at restoration by reset input.
At release of the stop mode, the contents of the internal RAM previous to the reset are held.
However, the contents of the CPU register and SFR are not held.
For resetting, refer to **"2.9 Reset."**

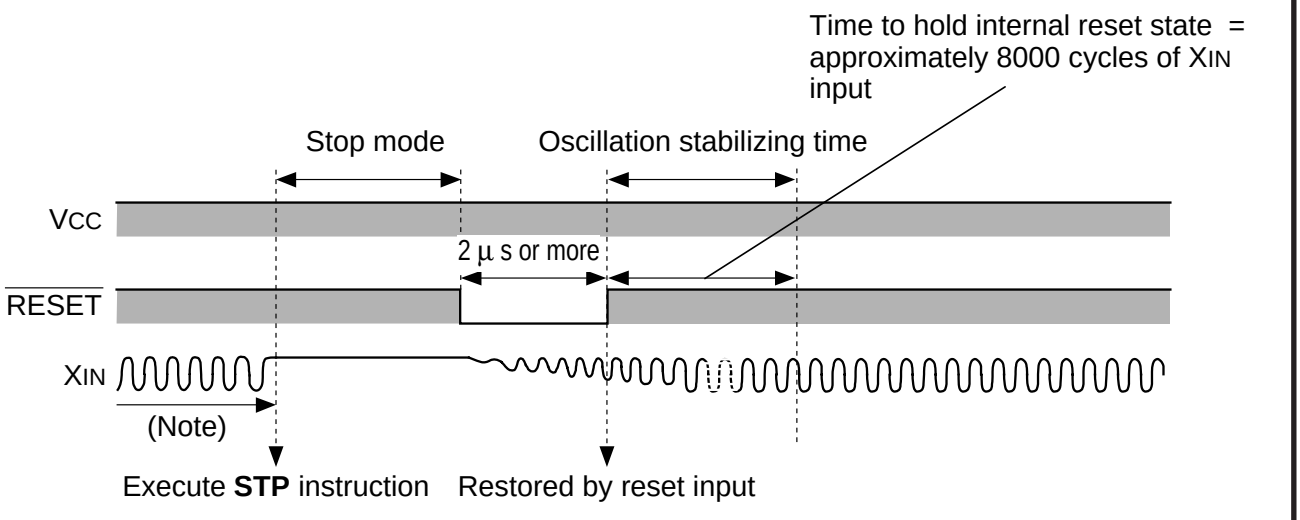


Fig. 2.8.1 Oscillation stabilizing time at restoration by reset input

2.8 Standby function

■Restoration by an interrupt request

The occurrence of an interrupt request in the stop mode releases the stop mode. As a result, oscillation is resumed. The interrupt requests available for restoration are:

- INT_0–INT_3
- Serial I/O transmit/receive using an external clock
- Timer X/Y using an external clock
- Key input (key-on wake up)

However, to use the above interrupt requests for restoration from the stop mode, after setting the following, execute the **STP** instruction in order to enable the interrupt request to be used.

[Necessary register setting]

① Interrupt disable flag I = "0" (interrupts enabled)
② Both timers 1 and 2 interrupt enable bits = "0" (interrupts disabled)
③ Interrupt request bit of the interrupt source to be used for restoration = "0" (no interrupt request issued)
④ Interrupt enable bit of the interrupt source to be used for restoration = "1" (interrupts enabled)

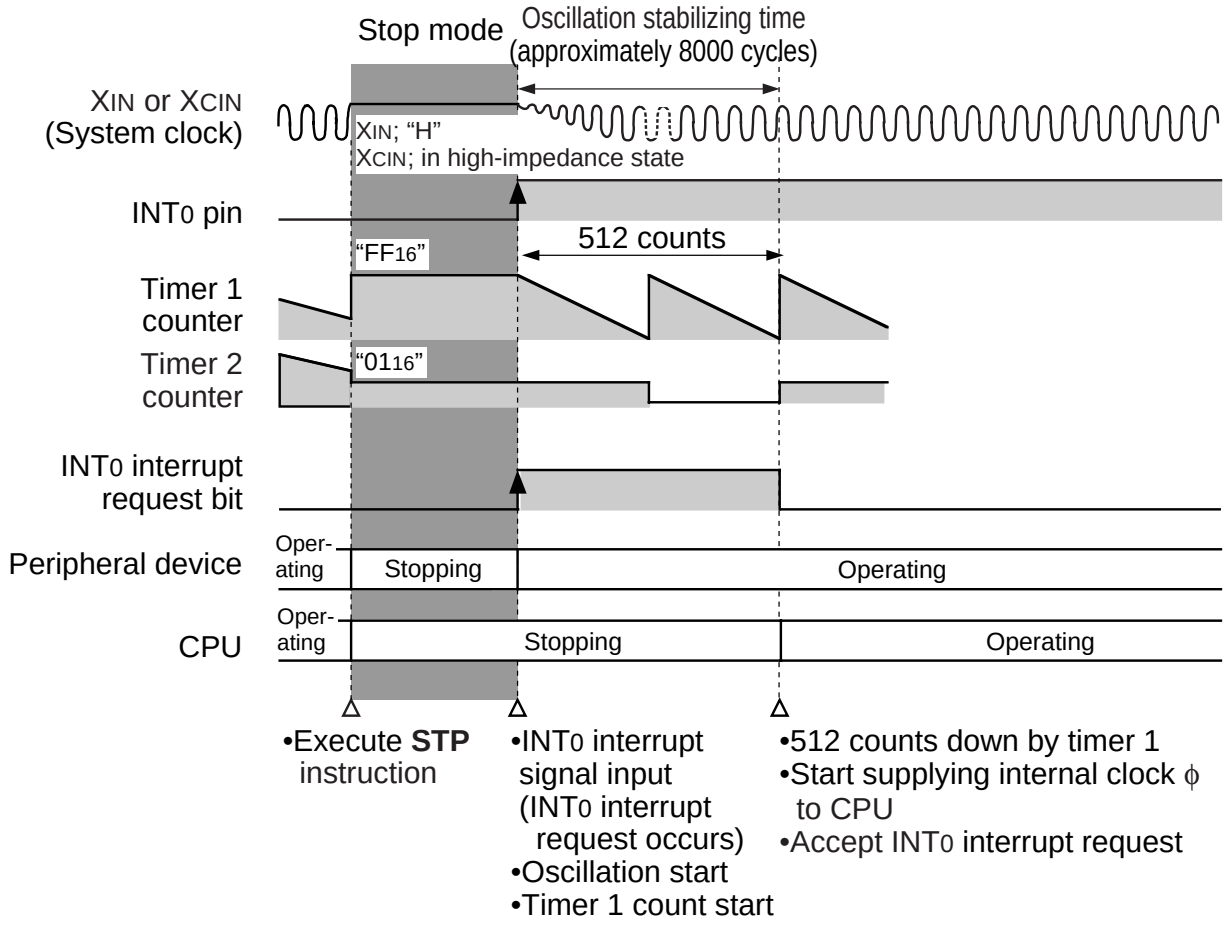
For interrupts, refer to **"2.2 Interrupts."**

The oscillation is unstable at start of oscillation. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) is required. At restoration by an interrupt request, the time to wait for supplying the internal clock ϕ to the CPU is automatically generated*1 by timers 1*2 and 2.*2 This wait time is reserved as the oscillation stabilizing time on the system clock side.
Figure 2.8.2 shows an execution sequence example at restoration by the occurrence of an INT_0 interrupt request.

✻1: At restoration from the stop mode, all bits except bit 4 of the timer 123 mode register (address 0029_{16}) are set to "0."
As a count source of the timer 1, an f(X_{IN})/16 or f(X_{CIN})/16 clock is selected. As a count source of the timer 2, the timer 1 underflow is selected.
Immediately after the oscillation is started, the count source is supplied to the timer 1 counter, so that a count operation is started. The supplying the internal clock ϕ to the CPU is started at the timer 2 underflow.

✻2: When the **STP** instruction is executed, "FF_{16}" and "01_{16}" are automatically set in the timer 1 counter/latch and timer 2 counter/latch respectively.

●When restoring from stop mode by using INT0 interrupt (rising edge selected)



Note: As a count source, f(XIN)/16 or f(XCIN)/16 is input. Either f(XIN)/16 or f(XCIN)/16 is selected by bit 7 of CPU mode register.

Fig. 2.8.2 Execution sequence example at restoration by occurrence of INT0 Interrupt request

(3) Notes on using the stop mode

■Release sources

The release sources of the stop mode are shown below.

•Reset input
•INT_0–INT_3 interrupts
•Serial I/O transmit/receive interrupts using an external clock
•Timers X/Y interrupts using an external clock
•Key input interrupt (key-on wake up)

Each INT pin (INT_0, INT_1, INT_2, INT_3) is also used as ports $P4_2$, $P4_3$, $P5_0$ or $P5_1$. To use INT_0 to INT_3 interrupts, after setting the corresponding bits of the following direction registers to "0" to set them for the input mode, execute the **STP** instruction.

•Port P4 direction register (address 0009_{16})
•Port P5 direction register (address $000B_{16}$)

The pin for a key input interrupt is also used as port P2. To use a key input interrupt, set the corresponding bits of the port P2 direction register (address 0005_{16}) to "0" for setting the input mode. And then, execute the **STP** instruction.

■Register setting

To use the above interrupt requests for restoration from the stop mode, after setting the following, execute the **STP** instruction in order to enable the interrupt request to be used.

[Necessary register setting]

① Interrupt disable flag I = "0" (interrupts enabled)
② Both timers 1 and 2 interrupt enable bits = "0" (interrupts disabled)
③ Interrupt request bit of the interrupt source to be used for restoration = "0" (no interrupt request issued)
④ Interrupt enable bit of the interrupt source to be used for restoration = "1" (interrupts enabled)

•At restoration from the stop mode, the values of the timers 1, 2 and 123 mode registers are automatically rewritten. Accordingly, set each of them again.
•To prevent a DC voltage from being applied to the LCD, after setting the LCD enable bit (bit 3) of the LCD mode register to "0," execute the **STP** instruction.

■Clock after restoration

After restoration from the stop mode by an interrupt request, the contents of the CPU mode register previous to the **STP** instruction execution are held. Accordingly, when both X_{IN} and X_{CIN} were oscillating before execution of the **STP** instruction, the oscillation of both X_{IN} and X_{CIN} is resumed at restoration from the stop mode by an interrupt request.

In the above case, when the X_{IN} 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 X_{CIN} side may not be stabilized even after the lapse of the oscillation stabilizing time (of the X_{IN} side).

2.8.2 Wait mode

The wait mode is set by execution of the **WIT** instruction. In the wait mode, the oscillation is continued, but the internal clock ϕ stops at the "H" level.
Since the oscillation is continued regardless of the CPU stop, the peripheral units operate.

(1) States in the wait mode

By executing the **WIT** instruction, the wait mode is set.
In the wait mode, the internal clock ϕ which is supplied to the CPU stops at the "H" level. The continuation of oscillation permits clock supply to the peripheral units.
Table 2.8.2 shows the state in the wait mode.

Table 2.8.2 State in wait mode

Item	State in wait mode
Oscillation	Operating
CPU	Stop
Internal clock ϕ	Stop at "H" level
I/O ports P0–P7	The state where **WIT** instruction is executed is held.
Timer, serial I/O, LCD display functions	Operating

(2) Release of wait mode

The wait mode is released by reset input or by the occurrence of an interrupt request.

There is a difference in restore processing from the wait mode by use of reset input and by use of an interrupt request.

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 reset state is provided by holding the input level of the $\overline{\text{RESET}}$ pin at "L" for 2 μs or more in the wait mode. As a result, the wait mode is released.

At the time when the wait mode is released, the supplying the internal clock φ to the CPU is started. The reset state is released in approximately 8,000 cycles of the XIN input after the input of the $\overline{\text{RESET}}$ pin is returned to the "H" level.

At release of the wait mode, the contents of the internal RAM previous to the reset are held. However, the contents of the CPU mode register and SFR are not held.

Figure 2.8.3 shows the reset input time.

For reset, refer to **"2.9 Reset."**

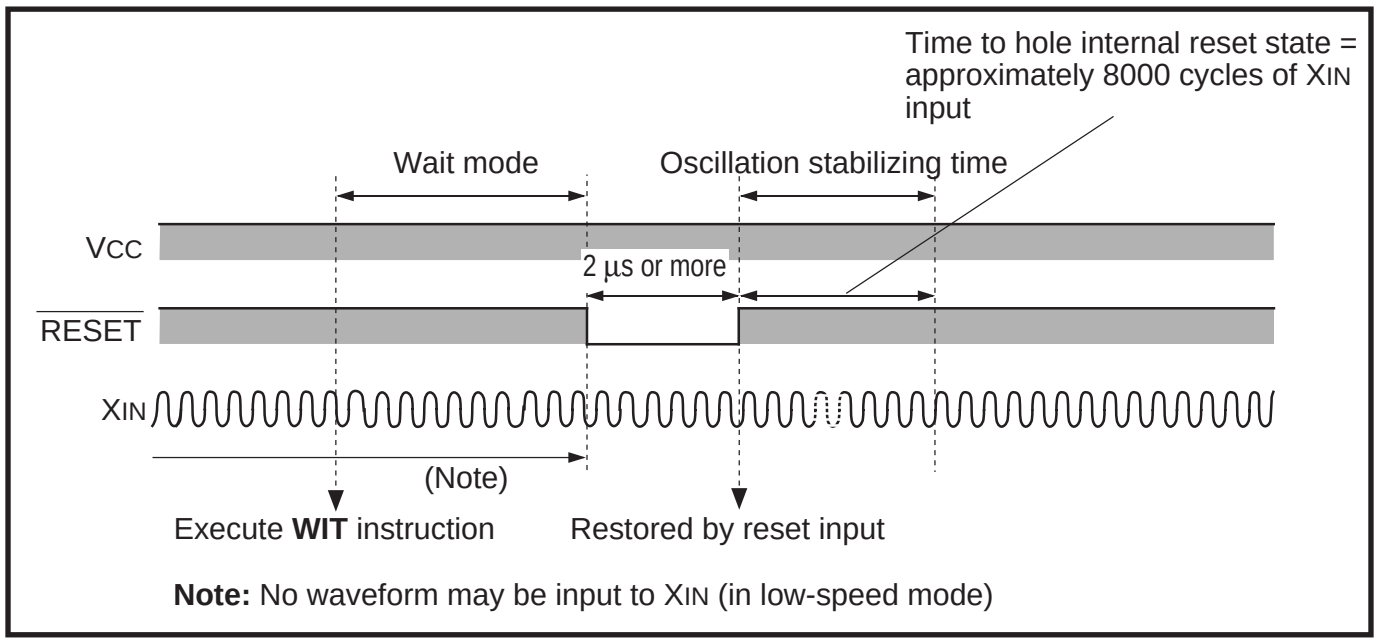


Fig. 2.8.3 Reset input time

■Restoration by an interrupt request

In the wait mode, the occurrence of an interrupt request releases the wait mode and the supplying 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, to use an interrupt for restoration from the wait mode, after setting the following, execute the **WIT** instruction in order to enable the interrupt to be used.

[Necessary register setting]

① Interrupt disable flag I = "0" (interrupts enabled)

② Interrupt request bit of the interrupt source to be used for restoration = "0" (no interrupt request issued)

③ Interrupt enable bit of the interrupt source to be used for restoration = "1" (interrupts enabled)

For interrupts, refer to **"2.2 Interrupts."**

(3) Notes on the wait mode

■Restoration by INT_0 to INT_3 interrupt requests

Each INT pin (INT_0, INT_1, INT_2, INT_3) is also used as ports $P4_2$, $P4_3$, $P5_0$ or $P5_1$. To use INT_0 to INT_3 interrupts, set the corresponding bits of the following direction registers to "0" for setting the input mode. And then, execute the **WIT** instruction.

•Port P4 direction register (address 0009_{16})

•Port P5 direction register (address $000B_{16}$)

■Restoration by key input interrupt request

The pins for a key input interrupt is also used as port P2. To use a key input interrupt, set the corresponding bits of the port P2 direction register (address 0005_{16}) to "0" for setting the input mode. And then, execute the **WIT** instruction.

■Register setting

To use the above interrupt requests for restoration from the stop mode, after setting the following, execute the **WIT** instruction in order to enable the interrupt request to be used.

[Necessary register setting]

① Interrupt disable flag I = "0" (interrupts enabled)

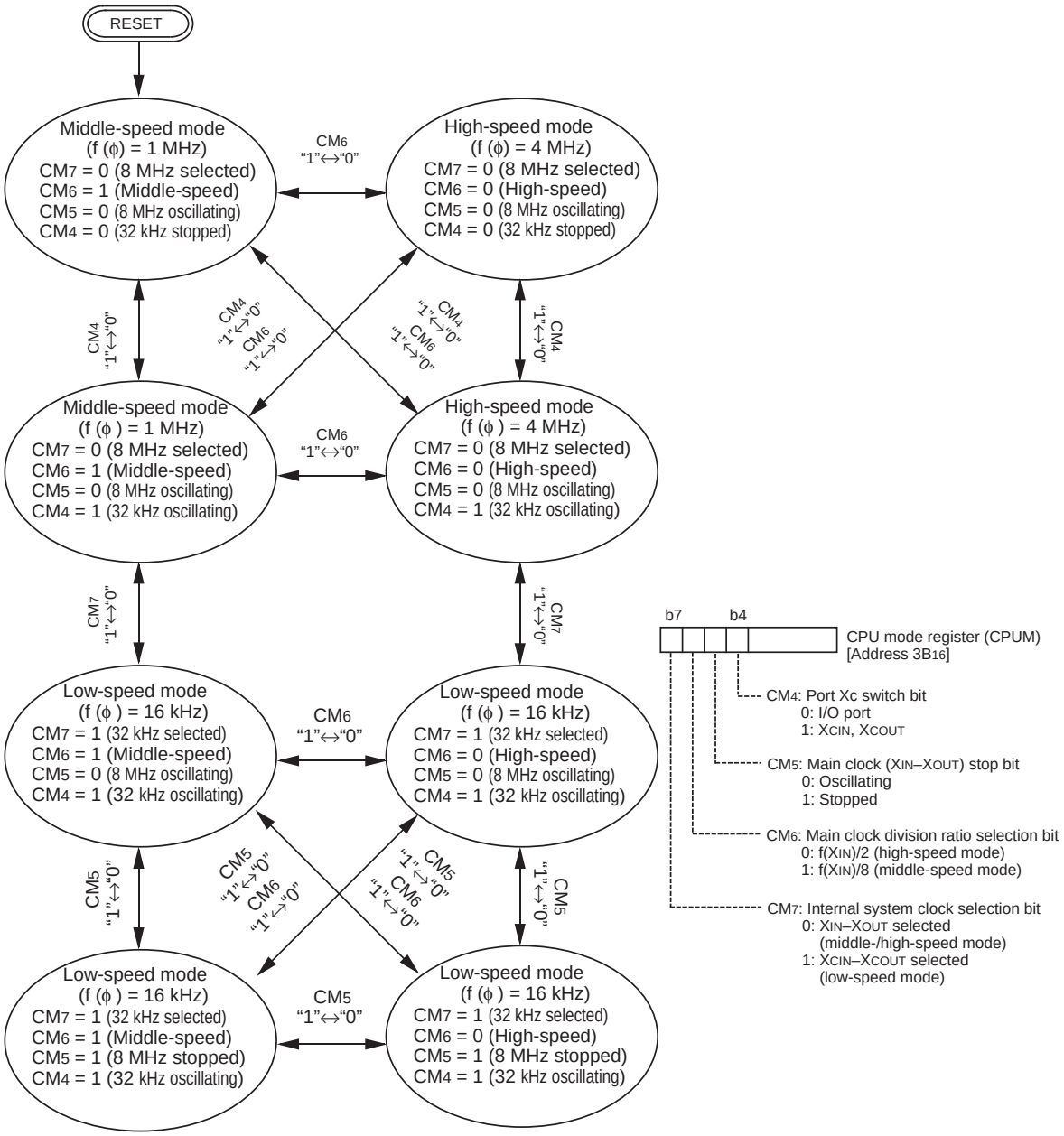
② Interrupt request bit of the interrupt source to be used for restoration = "0" (no interrupts request issued)

③ Interrupt enable bit of the interrupt source to be used for restoration = "1" (interrupts enabled)

2.8 Standby function

2.8.3 State transitions of internal clock φ

Figure 2.8.4 shows the state transitions of the internal clock φ when the standby function is used.



Notes 1: Switch the mode by the allows shown between the mode blocks.(Do not switch between the mode directly without an allow.)

2: The all modes can be switched to the stop mode or the wait mode and return to the source mode when the stop mode or the wait mode is released.

3: Timer and LCD operate in the wait mode.

4: In middle-/high-speed mode, when the stop mode is released, a delay of approximately 1 ms occurs automatically by timer 1 and timer 2.

5: In low-speed mode, when the stop mode is released, a delay of approximately 0.25 s occurs automatically by timer 1 and timer 2.

6: Wait until oscillation stabilizes after oscillating the main clock XIN before the switching from the low-speed mode to the middle-/high-speed mode.

7: The example assumes that 8 MHz is being applied to the XIN pin and 32 kHz to the XCIN pin. φ indicates the internal clock.

Fig. 2.8.4 State transitions of internal clock φ

2.9 Reset

The internal reset state is provided by applying a “L” level to the $\overline{\text{RESET}}$ pin. After that, the reset state is released by applying a “H” level to the $\overline{\text{RESET}}$ pin, so that the program is executed in the middle-speed mode starting from the contents at the reset vector address.

2.9.1 Explanation of operations

Figure 2.9.1 shows the internal reset state hold/release timing.

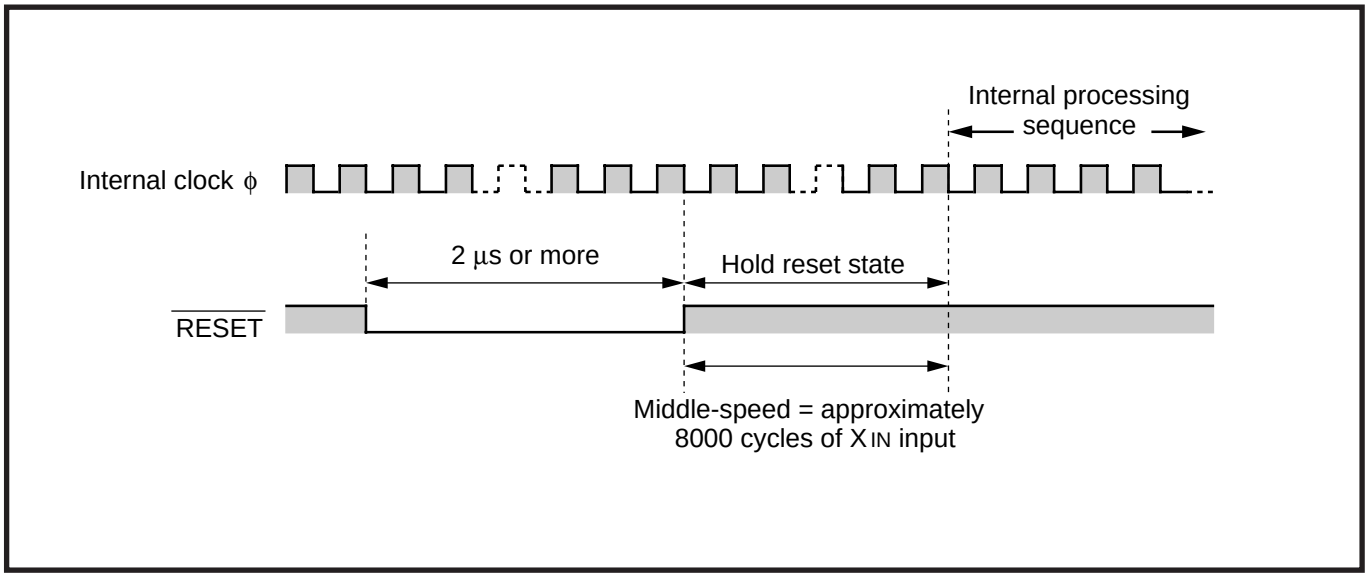


Fig. 2.9.1 Internal reset state hold/release timing

2.9 Reset

The reset state is provided by applying a “L” level to the $\overline{\text{RESET}}$ pin at power source voltage of 2.5 V to 5.5 V. Allow 2 μs or more as “L” level applying time.

By applying a “H” level to the $\overline{\text{RESET}}$ pin in the internal reset state, the timers and their count source shown in Table 2.9.1 is automatically set. After that, the internal reset state is released by the timer 2 underflow.

After applying “H” level, only the main clock oscillates in the middle-speed mode regardless of the oscillation state previous to internal resetting. The X_{CIN} pin on the sub-clock side becomes the input port.

After the internal reset state is released, the program is run from the address determined with the contents (high-order address) at address $FFFD_{16}$ and the contents (low-order address) at address $FFFC_{16}$.

Figure 2.9.2 shows the internal processing sequence immediately after reset release.

Table 2.9.1 Timers 1 and 2 at reset

Item	Timer 1	Timer 2
Value	FF_{16}	01_{16}
Count source	f (X_{IN})/16	Timer 1 underflow

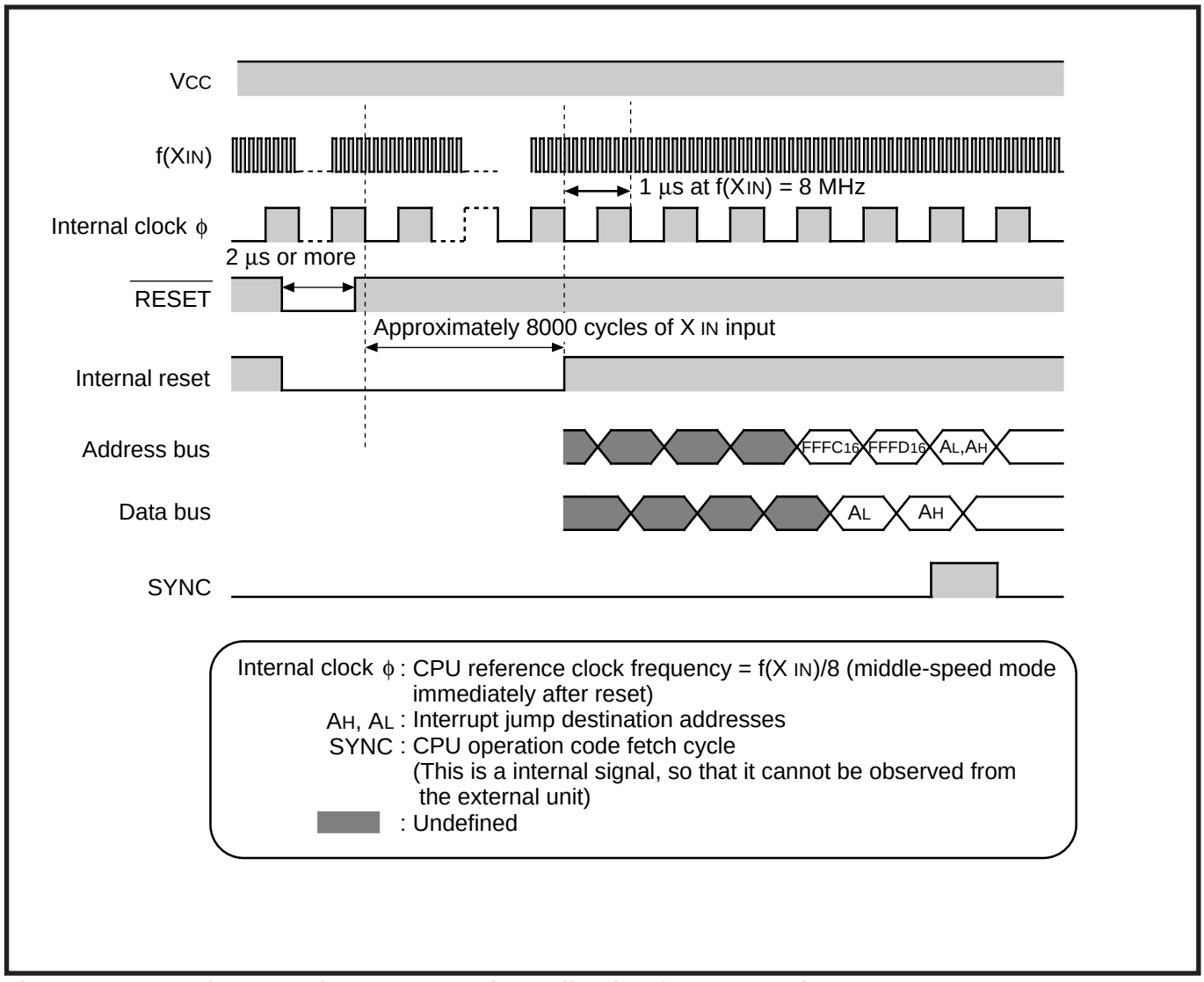


Fig. 2.9.2 Internal processing sequence immediately after reset release

2.9.2 Internal state of the microcomputer immediately after reset release

Figure 2.9.3 shows the internal state of the microcomputer immediately after reset release.
The contents of all other registers except registers in Figure 2.9.3 and internal RAM are undefined at poweron reset.

	Address	b7 Contents of register b1
Port P0 direction register	0001_{16}	— — — — — — — 0
Port P1 direction register	0003_{16}	— — — — — — — 0
Port P2 direction register	0005_{16}	00_{16}
Port P4 direction register	0009_{16}	0 0 0 0 0 0 0 —
Port P5 direction register	$000B_{16}$	00_{16}
Port P6 direction register	$000D_{16}$	00_{16}
Port P7 direction register	$000F_{16}$	— — — — — — 0 0
PULL register A	0016_{16}	— — — 0 1 0 1 1
PULL register B	0017_{16}	— — — 0 0 0 0 0
Serial I/O status register	0019_{16}	1 0 0 0 0 0 0 0
Serial I/O control register	$001A_{16}$	00_{16}
UART control register	$001B_{16}$	1 1 1 0 0 0 0 0
Timer X (low-order)	0020_{16}	FF_{16}
Timer X (high-order)	0021_{16}	FF_{16}
Timer Y (low-order)	0022_{16}	FF_{16}
Timer Y (high-order)	0023_{16}	FF_{16}
Timer 1	0024_{16}	FF_{16}
Timer 2	0025_{16}	01_{16}
Timer 3	0026_{16}	FF_{16}
Timer X mode register	0027_{16}	00_{16}
Timer Y mode register	0028_{16}	00_{16}
Timer 123 mode register	0029_{16}	00_{16}
ϕ output control register	$002A_{16}$	— — — — — — — 0
A-D control register	0034_{16}	0 0 0 0 1 0 0 0
Segment output enable register	0038_{16}	— — 0 0 0 0 0 0
LCD mode register	0039_{16}	0 0 0 — 0 0 0 0
Interrupt edge selection register	$003A_{16}$	00_{16}
CPU mode register	$003B_{16}$	0 1 0 0 1 0 0 0
Interrupt request register 1	$003C_{16}$	00_{16}
Interrupt request register 2	$003D_{16}$	00_{16}
Interrupt control register 1	$003E_{16}$	00_{16}
Interrupt control register 2	$003F_{16}$	00_{16}
Processor status register	(PS)	X X X X X 1 X X
Program counter	(PC_H)	Contents of address $FFFD_{16}$
	(PC_L)	Contents of address $FFFC_{16}$

Notes —: Unused bits
X : Undefined
The contents of all other registers and internal RAM are undefined at poweron reset, so they must be initialized by software.

Fig. 2.9.3 Internal state of microcomputer immediately after reset release

2.9 Reset

2.9.3 Reset circuit

Design a configuration of the reset circuit so that the reset input voltage may be 0.5 V or less at the time when the power sorce voltage passes 2.5 V.

Figure 2.9.4 shows the poweron reset conditions and Figure 2.9.5 shows poweron reset circuit examples.

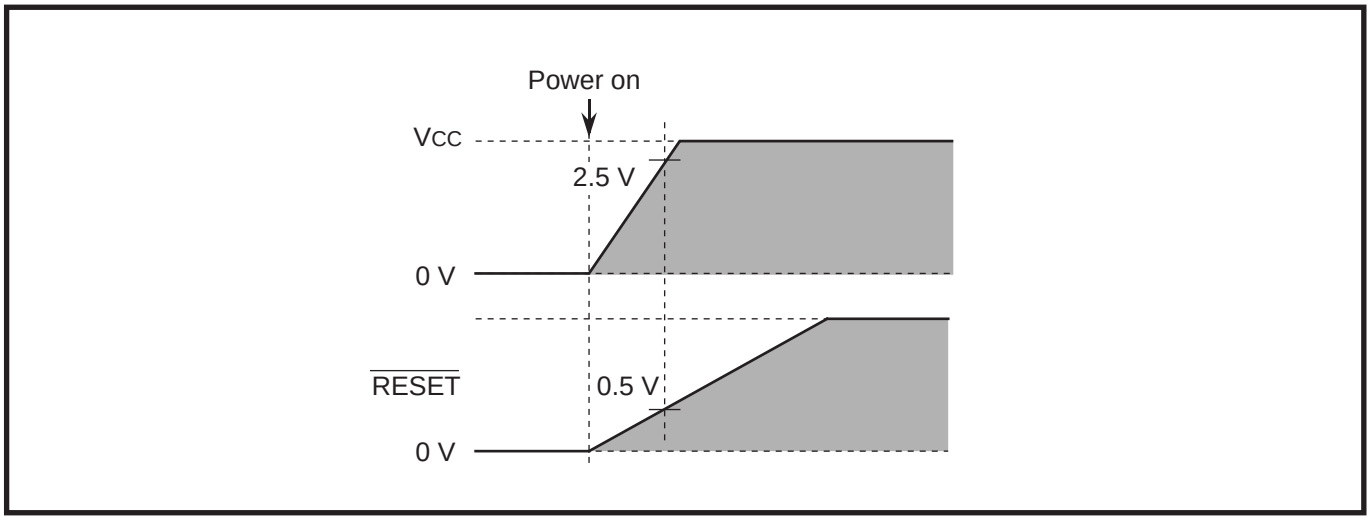


Fig. 2.9.4 Poweron reset conditions

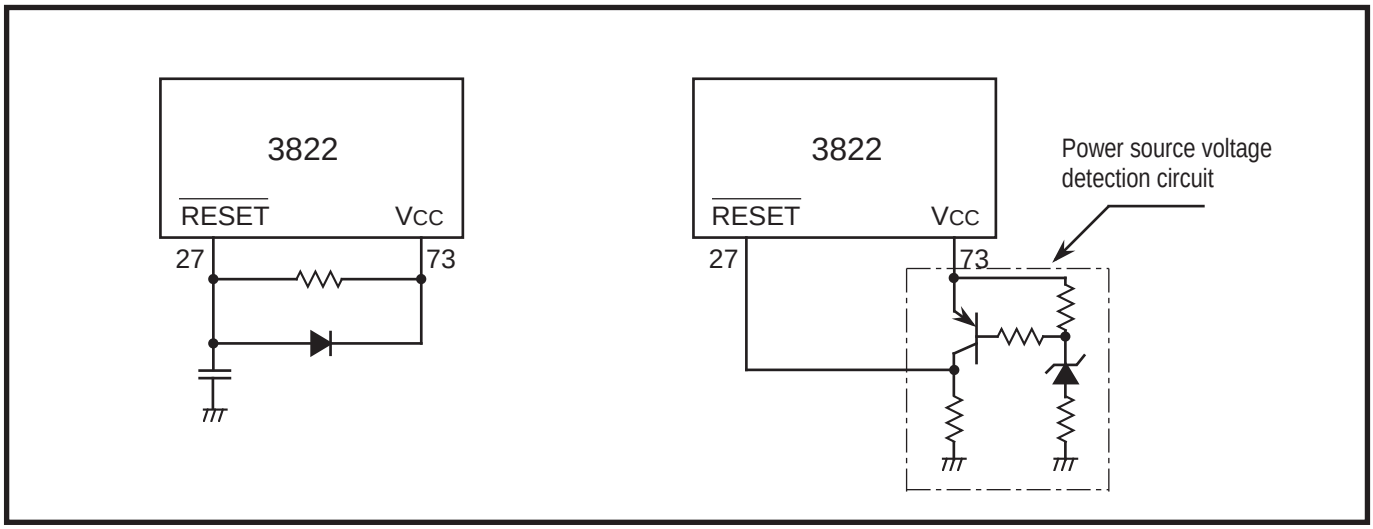


Fig. 2.9.5 Poweron reset circuit examples

2.9.4 Notes on the $\overline{\text{RESET}}$ pin

In case where the reset signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the Vss pin. And use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following:

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to check 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.

2.10 Oscillation circuit

2.10.1 Oscillation circuit

Two oscillation circuits are included to obtain clocks required for operations.

- XIN–XOUT oscillation circuit.....Main clock (XIN input) oscillation circuit
- XCIN–XCOUT oscillation circuit.....Sub-clock (XCIN input) oscillation circuit

A clock*1 obtained by dividing the frequency input to the clock input pins XIN or XCIN is an internal clock ϕ. The internal clock ϕ is used as a standard for operations.

*1: The internal clock ϕ varies with modes.

- •High-speed mode .....Frequency input to the XIN pin/2
- •Middle-speed mode .....Frequency input to the XIN pin/8
- •Low-speed mode .....Frequency input to the XCIN pin/2

(1) Oscillation circuit using ceramic resonators

Figure 2.10.1 shows an oscillation circuit example using ceramic resonators. As shown in the figure, an oscillation circuit can be formed by connecting a ceramic resonator or a quartz-crystal oscillator between the pins XIN and the XOUT and between the pins XCIN and XCOUT. As the XIN–XOUT oscillation circuit includes a feedback resistor, an external resistor is omissible.

As the XCIN–XCOUT oscillation circuit does not include any feedback resistor, connect a feedback resistor externally.

Regarding circuit constants for Rf, Rd, CIN, COUT, CCIN and CCOUT, ask the resonator manufacturer for information, and set the values recommended by the resonator manufacturer.

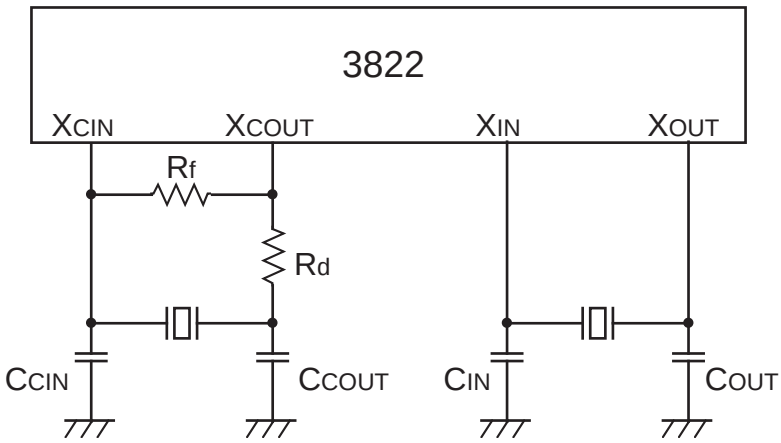


Fig. 2.10.1 Oscillation circuit example using ceramic resonators

(2) External clock input circuit

An external clock can also be supplied to the main clock oscillation circuit.

Figure 2.10.2 shows an external clock input circuit example. As an external clock to be input to the XIN pin, use a pulse signal with a duty ratio of 50%. At this time, open the XOUT pin.

Any clock externally generated cannot be input to the XCIN pin directly. Cause oscillation with an external ceramic resonator.

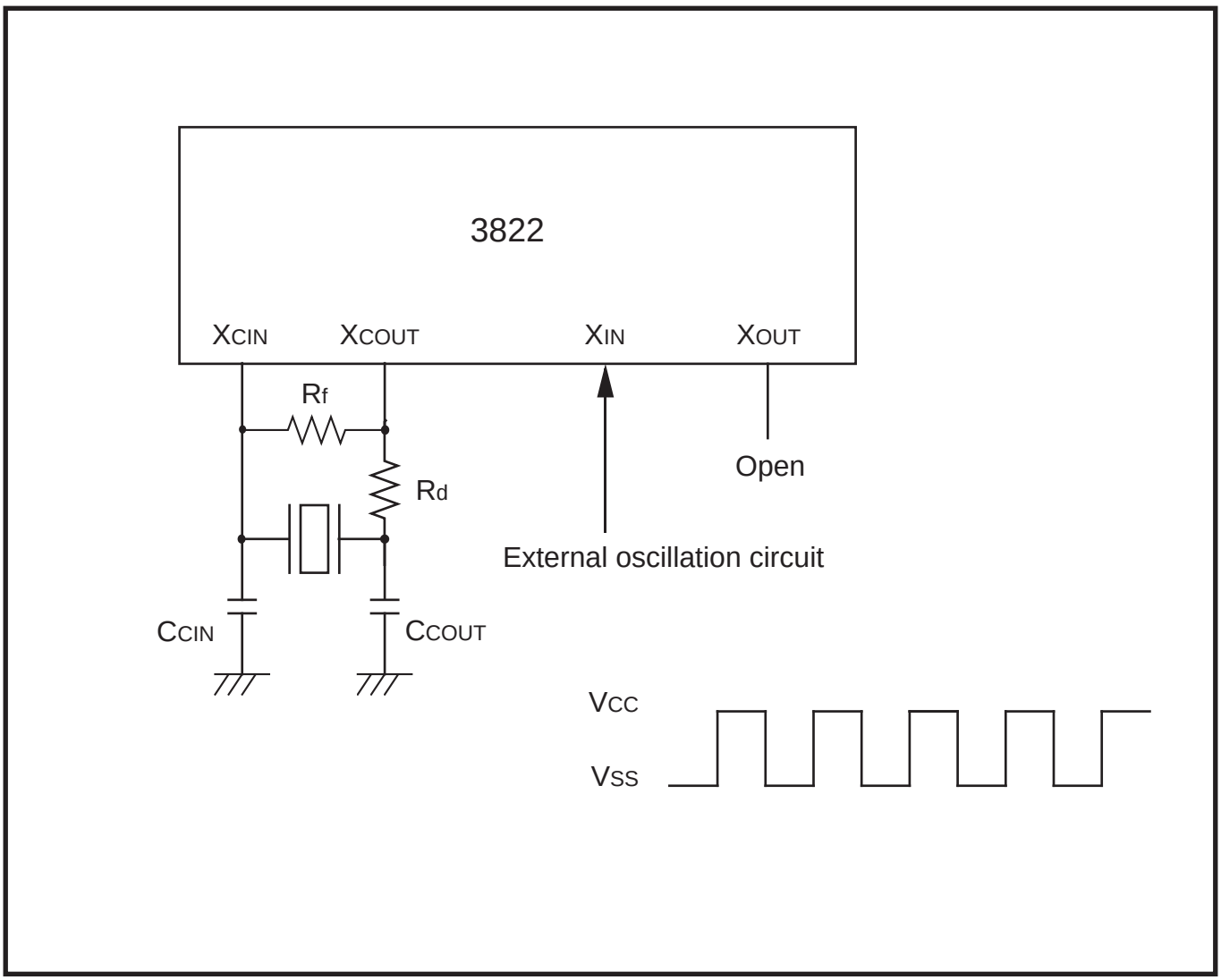


Fig. 2.10.2 External clock input circuit example

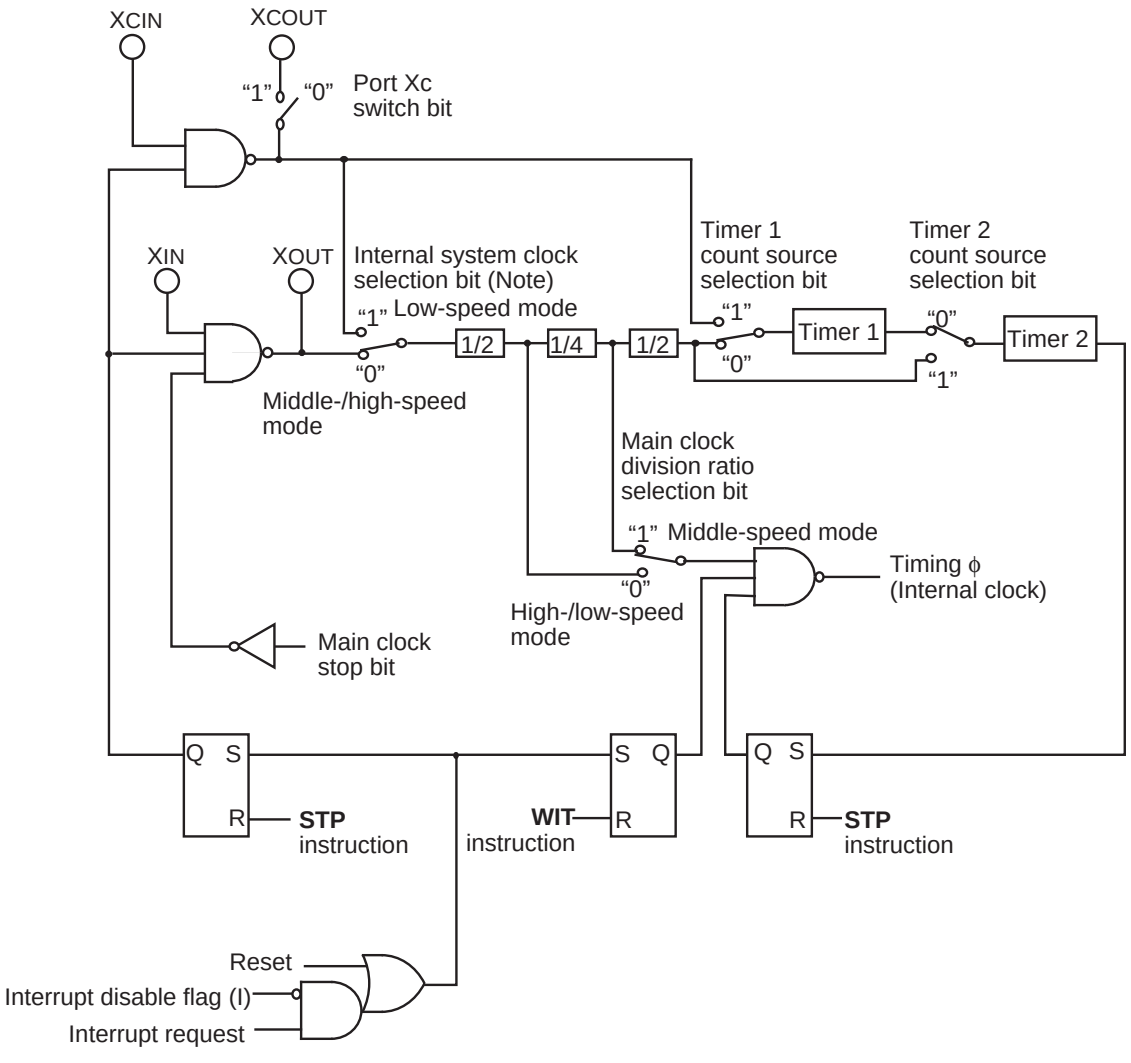
2.10.2 Internal clock ϕ

The internal clock ϕ is the standard for operations.

(1) Clock generating circuit

The clock generating circuit controls the oscillation of the oscillation circuit. The generated clock (internal clock ϕ) is supplied to the CPU and peripheral units.

Figure 2.10.3 shows the clock generating circuit block diagram. Oscillation can be stopped and resumed by the clock generating circuit.



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

Fig. 2.10.3 Clock generating circuit block diagram

(2) Clock output function

The internal clock φ can be output from the φ pin by setting the φ output control bit (bit 0) of the φ output control register (address 002A16) to "1."

The φ pin is also used as port P41. Accordingly, to use it as an φ pin, set bit 1 of the port P4 direction register to "1."

Figure 2.10.4 shows the structure of the φ output control register.

φ output control register

b7 b6 b5 b4 b3 b2 b1 b0

φ output control register (CKOUT) [Address 2A 16]

B	Name	Functions	At reset	R	W
0	φ output control bit	0: Port function 1: φ clock output (Port direction register = "1")	0	○	○
1 to 7	Nothing is allocated. These bits cannot be written to and are fixed to "0" at reading.		0	0	×

Fig. 2.10.4 Structure of φ output control register

2.10 Oscillation circuit

2.10.3 Oscillating operation

The start and stop sources for oscillating operation are described below.

(1) Oscillating operation

At reset release, the middle-speed mode is provided. At this time, only the main clock oscillates and the XCIN and XCOUT pins function as I/O ports.

To use the sub-clock, set the $P7_0$, $P7_1$ pull-up bit (bit 4) of the PULL register A (address 0016_{16}) to "0" and disconnect each pull-up resistor of the XCIN and XCOUT pins.

■Middle-speed mode

The clock obtained by dividing the frequency "f(XIN)" which is input to the XIN pin by 8 is an internal clock ϕ after reset release.

When changing to the high-speed mode:

Set the main clock division ratio selection bit (bit 6) of the CPU mode register (address $003B_{16}$) to "0."

When changing to the low-speed mode:

Change the mode according to the following procedure.

① Set the port Xc switch bit (bit 4) of the CPU mode register to "1."
② Generate the oscillation stabilizing time of XCIN input by software.
③ Set the internal system clock selection bit (bit 7) of the CPU mode register to "1."

■High-speed mode

The clock obtained by dividing f(XIN) by 2 is an internal clock ϕ.

When changing to the middle-speed mode:

Set the main clock division ratio selection bit (bit 6) of the CPU mode register to "1."

When changing to the low-speed mode:

Change the mode according to the following procedure.

① Set the port Xc switch bit (bit 4) of the CPU mode register to "1."
② Generate the oscillation stabilizing time of XCIN input by software.
③ Set the internal system clock selection bit (bit 7) of the CPU mode register to "1."

■Low-speed mode

The clock obtained by dividing the frequency f(XCIN) input to the XCIN pin by 2 is an internal clock ϕ. In the low-speed mode, the oscillation of the main clock is stopped by setting the main clock (XIN–XOUT) stop bit to "1," so that the low-power operation can be attained.

When changing to the middle- or high-speed modes:

Change the mode according to the following procedure.

① Set the main clock (XIN–XOUT) stop bit (bit 5) of the CPU mode register to "0."
② Generate the oscillation stabilizing time of XIN input by software.
③ Set the internal system clock selection bit (bit 7) of the CPU mode register to "0."
④ Specify the main clock division ratio selection bit (bit 6) of the CPU mode register.

Notes1: Make a mode change from the middle- or high-speed modes to the low-speed mode after the oscillation of both the main clock and the sub-clock is stabilized (for oscillation stabilizing time, ask the resonator manufacturer for information).

2: For the sub-clock, the stabilizing of oscillation requires much time. When making a change from the middle- or high-speed modes to the stop mode and then making a return from the stop mode while the sub-clock oscillates, the oscillation of the sub-clock is not yet stabilized even when the main clock has become stable and the CPU has been restored.

3: For a mode change, set to f(XIN) > f(XCIN) × 3.

(2) Oscillating operation in the stop mode

After the stop mode is provided by executing the **STP** instruction, every oscillation stops and the internal clock ϕ stops at the "H" level. At the time when restoration is made from the stop mode by rest input or by the occurrence of an interrupt request for restoration, oscillation starts.

For the details of the stop mode, refer to **"2.8.1 Stop mode."**

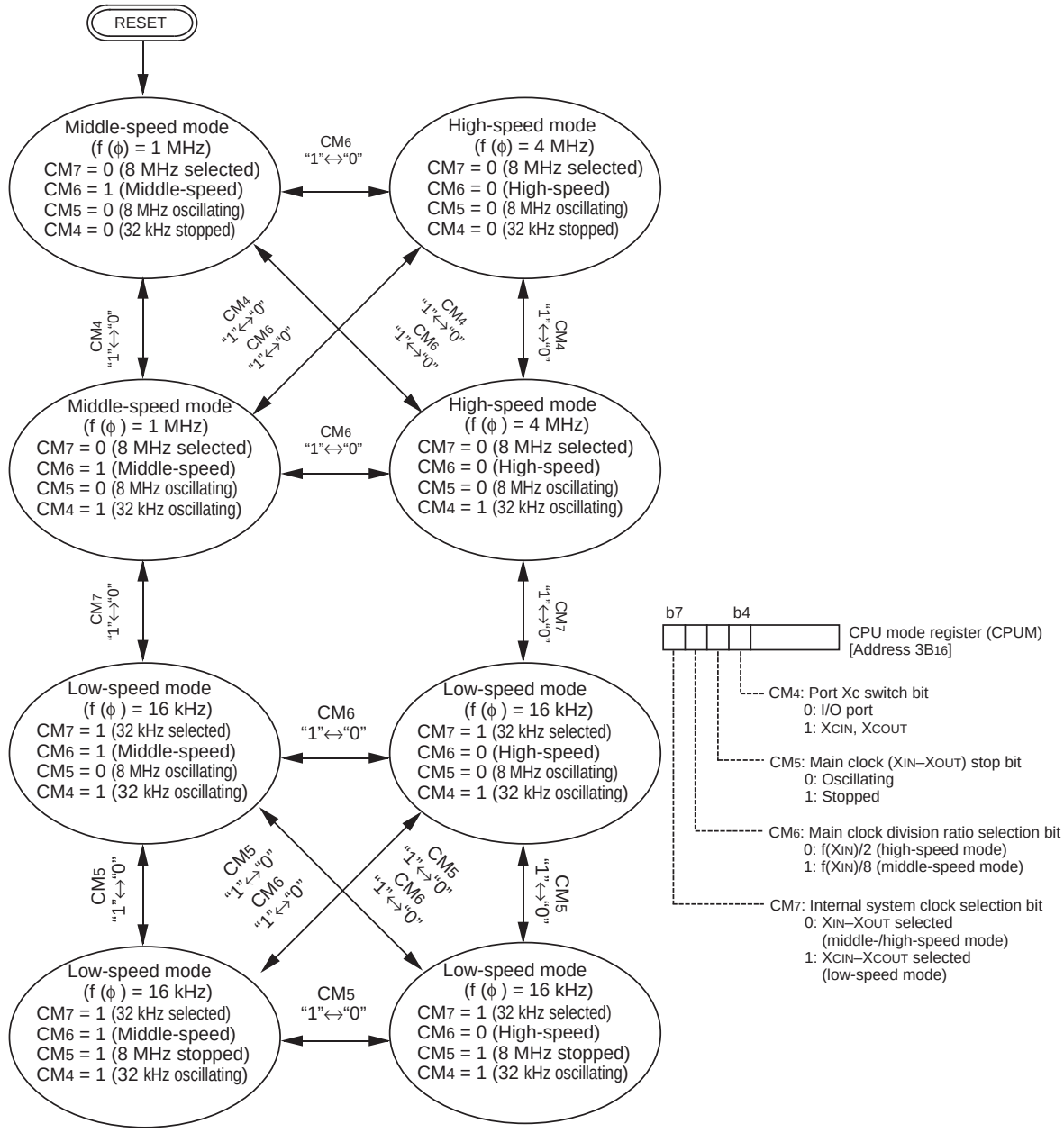
(3) Oscillating operation in the wait mode

After the wait mode is provided by executing the **WIT** instruction, the internal clock ϕ supplied to the CPU stops at the "H" level. As oscillation is continued, the supply of internal clock ϕ to the peripheral units is continued.

At the time when restoration is made from the wait mode by reset input or by the occurrence of an interrupt request for restoration, the supply of internal clock ϕ to the CPU starts. For the details of the wait mode, refer to **"2.8.2 Wait mode."**

(4) State transitions of internal clock ϕ

Figure 2.10.5 shows the state transitions of the internal clock ϕ.



Notes 1: Switch the mode by the allows shown between the mode blocks.(Do not switch between the mode directly without an allow.)

2: The all modes can be switched to the stop mode or the wait mode and return to the source mode when the stop mode or the wait mode is released.

3: Timer and LCD operate in the wait mode.

4: In middle-/high-speed mode, when the stop mode is released, a delay of approximately 1 ms occurs automatically by timer 1 and timer 2.

5: In low-speed mode, when the stop mode is released, a delay of approximately 0.25 s occurs automatically by timer 1 and timer 2.

6: Wait until oscillation stabilizes after oscillating the main clock XIN before the switching from the low-speed mode to the middle-/high-speed mode.

7: The example assumes that 8 MHz is being applied to the XIN pin and 32 kHz to the XCIN pin. ϕ indicates the internal clock.

Fig. 2.10.5 State transitions of internal clock ϕ

2.10.4 Oscillation stabilizing time

In the oscillating circuit using ceramic resonators, the oscillation is unstable for a certain time when the oscillation of the resonators starts.
The time required for stabilizing of oscillation is called oscillation stabilizing time.
An appropriate oscillation stabilizing time is required in accordance with the conditions of the oscillation circuit in use. For oscillation stabilizing time, ask the resonator manufacturer for information.

(1) Oscillation stabilizing time at poweron

In the oscillating circuit using ceramic resonators, oscillation is unstable for a certain time immediately after poweron. At reset release, the oscillation stabilizing time for approximately 8,000 cycles of X_{IN} input is automatically generated.
Figure 2.10.6 shows the oscillation stabilizing time at poweron.

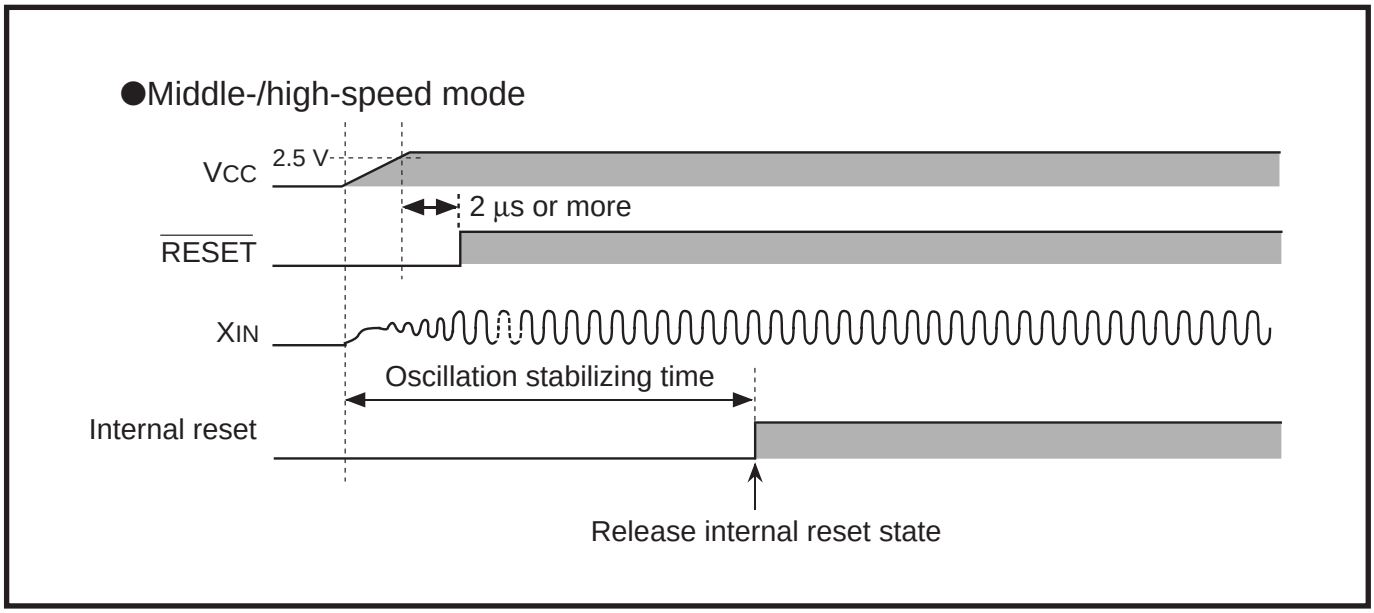


Fig. 2.10.6 Oscillation stabilizing time at poweron

2.10 Oscillation circuit

(2) Oscillation stabilizing time at restoration from the stop mode

In the stop mode, oscillation stops.

When restoration is made from the stop mode by reset input or an interrupt request, the oscillation stabilizing time for approximately 8,000 cycles of XIN input or XCIN input is automatically generated as at poweron.

At restoration made by reset, XIN input is a clock source of oscillation stabilizing time.

At restoration made by an interrupt request, either XIN input or XCIN input set as a system clock immediately before execution of the **STP** instruction becomes a count source of oscillation stabilizing time.

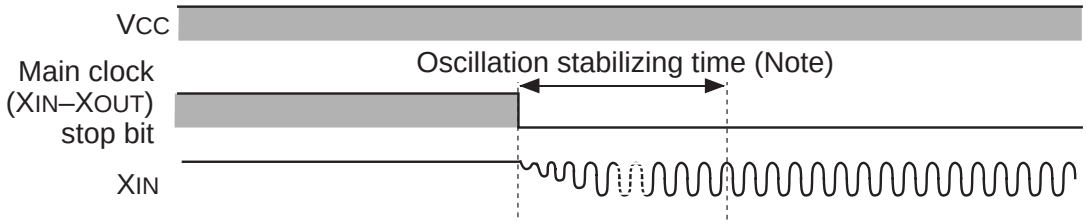
When XIN input is a system clock, the oscillation stabilizing time at restoration becomes approximately 8,000 cycles of XIN input. However, note that the oscillation on the XCIN side may not be stable even after the lapse of this oscillation stabilizing time.

For the details of the stop mode, refer to **"2.8.1 Stop mode."**

(3) Oscillation stabilizing time at reoscillation of XIN

When the oscillation of XIN which was stopped by setting the main clock (XIN–XOUT) stop bit of the CPU mode register to "1" is resumed, set this bit to "0" for reoscillation. At this time, generate oscillation stabilizing time by software.

Figure 2.10.7 shows the oscillation stabilizing time at reoscillation of XIN.



Note: For oscillation stabilizing time, ask the resonator manufacturer for information.

Fig. 2.10.7 Oscillation stabilizing time at reoscillation of XIN

CHAPTER 3
APPENDIX

APPENDIX

3.1 Built-in PROM version

In contrast with the mask ROM version, the microcomputer with a built-in programmable ROM is called the built-in programmable ROM version (referred as “the built-in PROM version”).
The following two types of built-in PROM version are available.

•EPROM version.....The contents of the built-in EPROM version can be written, deleted and rewritten.
•One Time PROM version.....The contents of the built-in PROM can be written only once and cannot be deleted and rewritten.

The EPROM version has the function of the One Time PROM version and also permits deleting and rewriting the contents of the PROM.

3.1.1 Product expansion

Table 3.1.1 shows the product expansion of the built-in PROM version.

Table 3.1.1 Product expansion of built-in PROM version

Product	PROM	RAM	Package	Programming adapter	Remarks
M38223E4-XXXFP	One Time PROM 16384 bytes (16254 bytes)	512 bytes	80P6N-A*1	PCA4738F-80	Shipped after programming and inspection at plant
M38223E4FP					Shipped in blank*5
M38223E4-XXXGP			80P6S-A*2	PCA4738G-80	Shipped after programming and inspection at plant
M38223E4GP					Shipped in blank*5
M38223E4-XXXHP			80P6D-A*3	PCA4738H-80	Shipped after programming and inspection at plant
M38223E4HP					Shipped in blank*5
M38223E4FS	EPROM 16384 bytes (16254 bytes)		80D0*4	PCA4738L-80	EPROM version

✻1 80P6N-A : 0.8 mm-pitch plastic molded QFP
✻2 80P6S-A : 0.65 mm-pitch plastic molded QFP
✻3 80P6D-A : 0.5 mm-pitch plastic molded QFP
✻4 80D0 : 0.8 mm-pitch ceramic QFP
✻5 Shipped in blank: The product is shipped without writing any data in the built-in PROM

Note: The number in parentheses denotes a user ROM capacity.

3.1.2 Performance overview

Table 3.1.2 shows a performance overview of the built-in PROM version.
The performance of the built-in PROM version is the same as that of the mask ROM version with the exception that the PROM is built in.

Table 3.1.2 Performance overview of built-in PROM version

Parameter		Performance	
Basic instructions		71	
Instruction execution time		0.5 μs (minimum instructions at 8MHz oscillation frequency)	
Memory sizes	PROM	M38223E4	16384 bytes (user ROM capacity; 16254 bytes)
	RAM	M38223E4	512 bytes
Programmable I/O ports		49	
Oscillation frequency	Main clock f(XIN)	8 MHz (maximum)	
	Sub-clock f(XCIN)	32 kHz (standard) to 50 kHz (maximum)	
Interrupts		17 sources, 16 vectors (includes key input interrupt)	
Timers		8-bit × 3	
		16-bit × 2	
Serial I/O		8-bit × 1 (operable in clock synchronous mode and UART mode)	
A-D comparator		8-bit × 8 channels	
LCD (Liquid Crystal Display) drive control functions	Bias	Select 1/2 or 1/3	
	Duty ratio	Select duty ratio value of 2, 3, or 4	
	Segment output	32 (maximum)	
	Common output	4 (maximum)	
ϕ clock output function		1-bit output	
Clock generating circuit		2 built-in circuits (connect an external ceramic resonator or an external quartz-crystal oscillator)	
Power source voltage		2.5 V (minimum) to 5.0 V (standard) to 5.5 V (maximum) ✻4.0 V (minimum) in high-speed mode. However, at f(XIN) = (4 × VCC – 8) MHz, 2.5 V to 4.0 V is possible.	
Power dissipation	High-speed mode	32 mW (at 8 MHz oscillation frequency, VCC = 5 V)	
	Low-speed mode	0.045 mW (at 32 MHz oscillation frequency, VCC = 3 V)	
Operating temperature range		–20 to 85 °C	
Device structure		CMOS silicon gate	
Packages	EPROM version	80D0 (0.8 mm-pitch ceramic LCC)	
	One Time PROM version	80P6N-A (0.8 mm-pitch plastic mold QFP)	
		80P6S-A (0.65 mm-pitch plastic mold QFP)	
		80P6D-A (0.5 mm-pitch plastic mold QFP)	

Note: The parts enclosed by thick line denotes performance peculiar to the PROM version.

APPENDIX

3.1 Built-in PROM version

3.1.3 Pin configuration

The pin configuration of the built-in PROM version is the same as that of the mask ROM version. Figure 3.1.1 shows the pin configuration of the EPROM version.

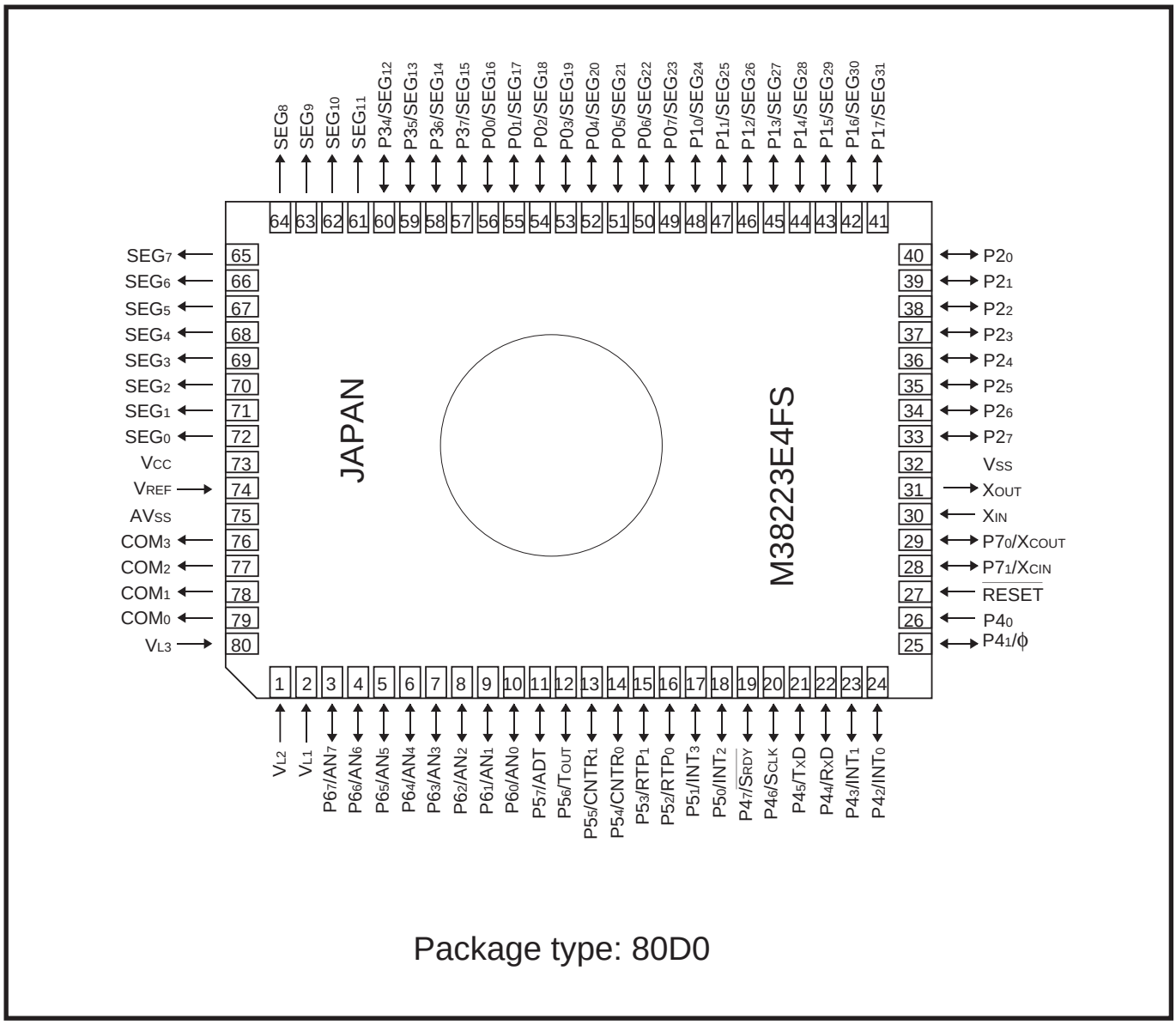
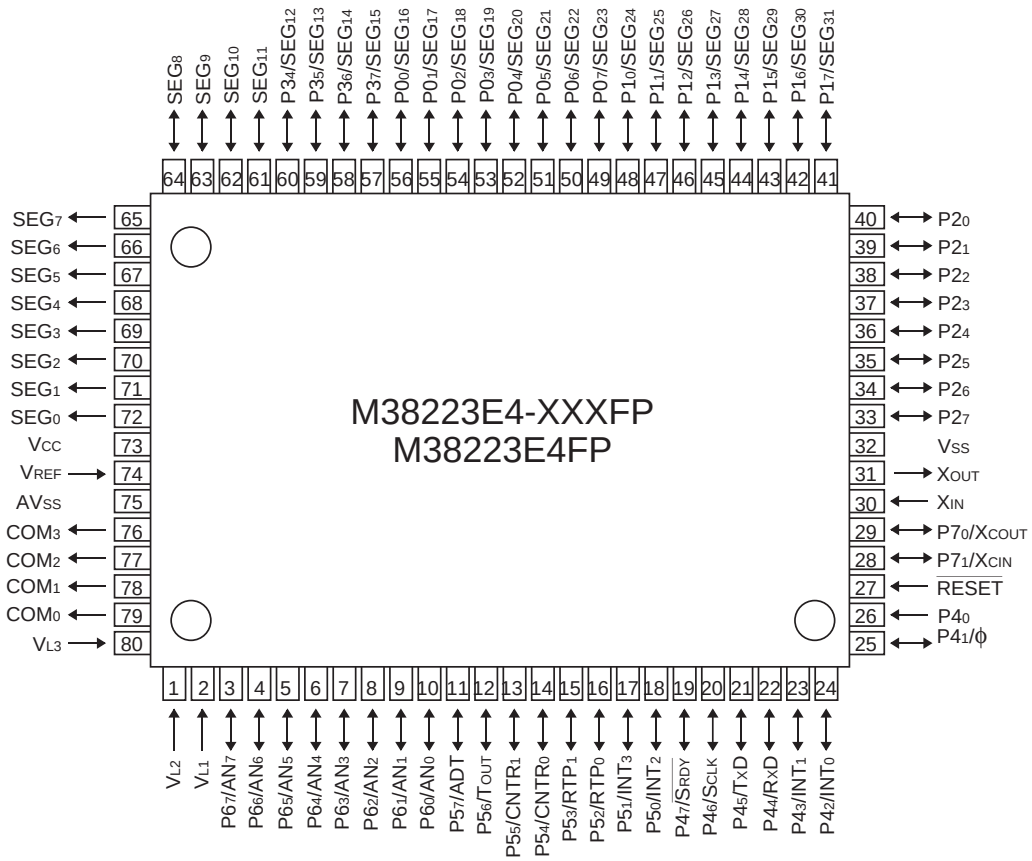


Fig. 3.1.1 Pin configuration of EPROM version (top view)

Figure 3.1.2 and Figure 3.1.3 show the pin configurations of the One Time PROM version.



Package type: 80P6N-A

Fig. 3.1.2 Pin configuration of One Time PROM version (top view) (1)

APPENDIX

3.1 Built-in PROM version

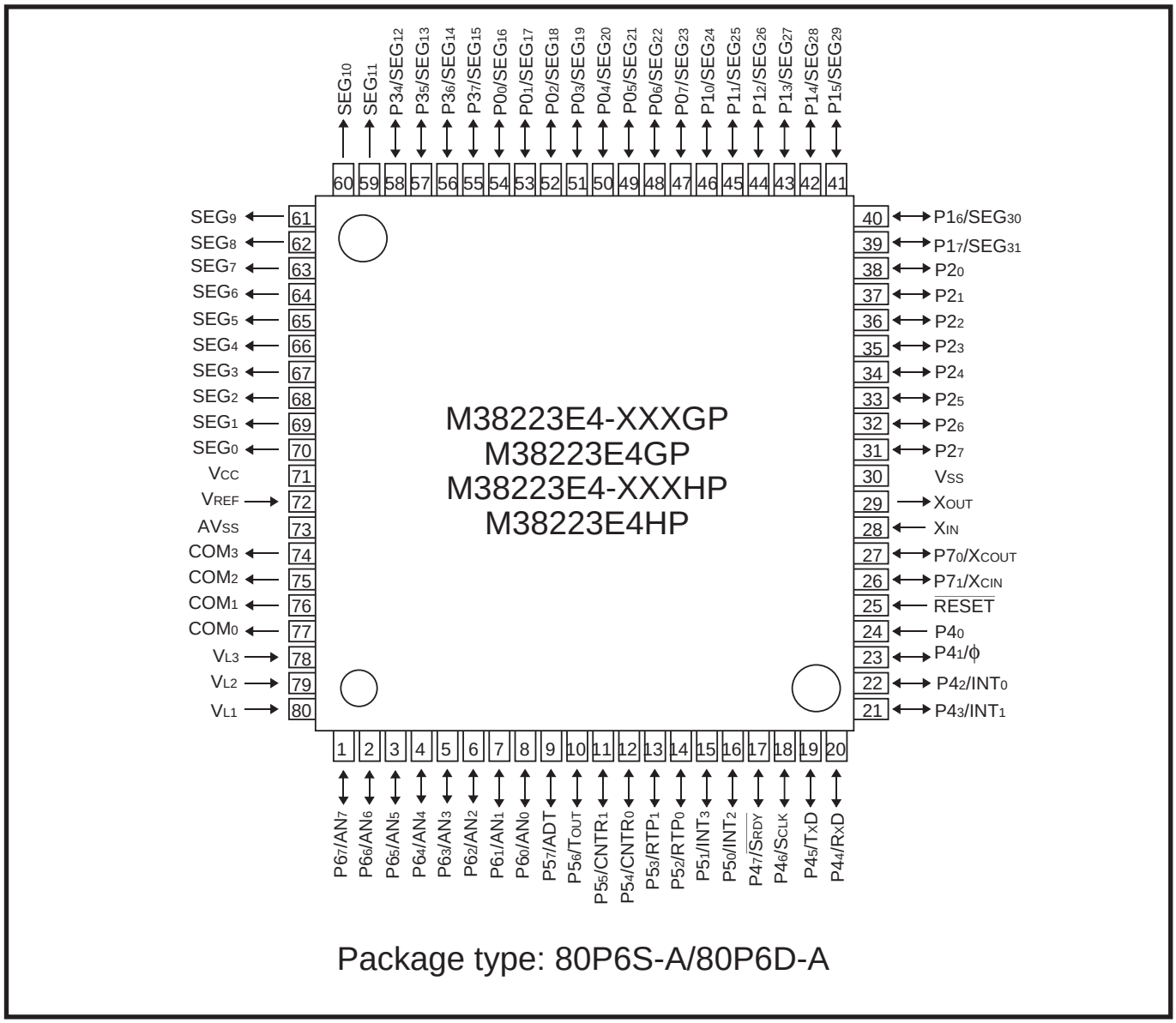
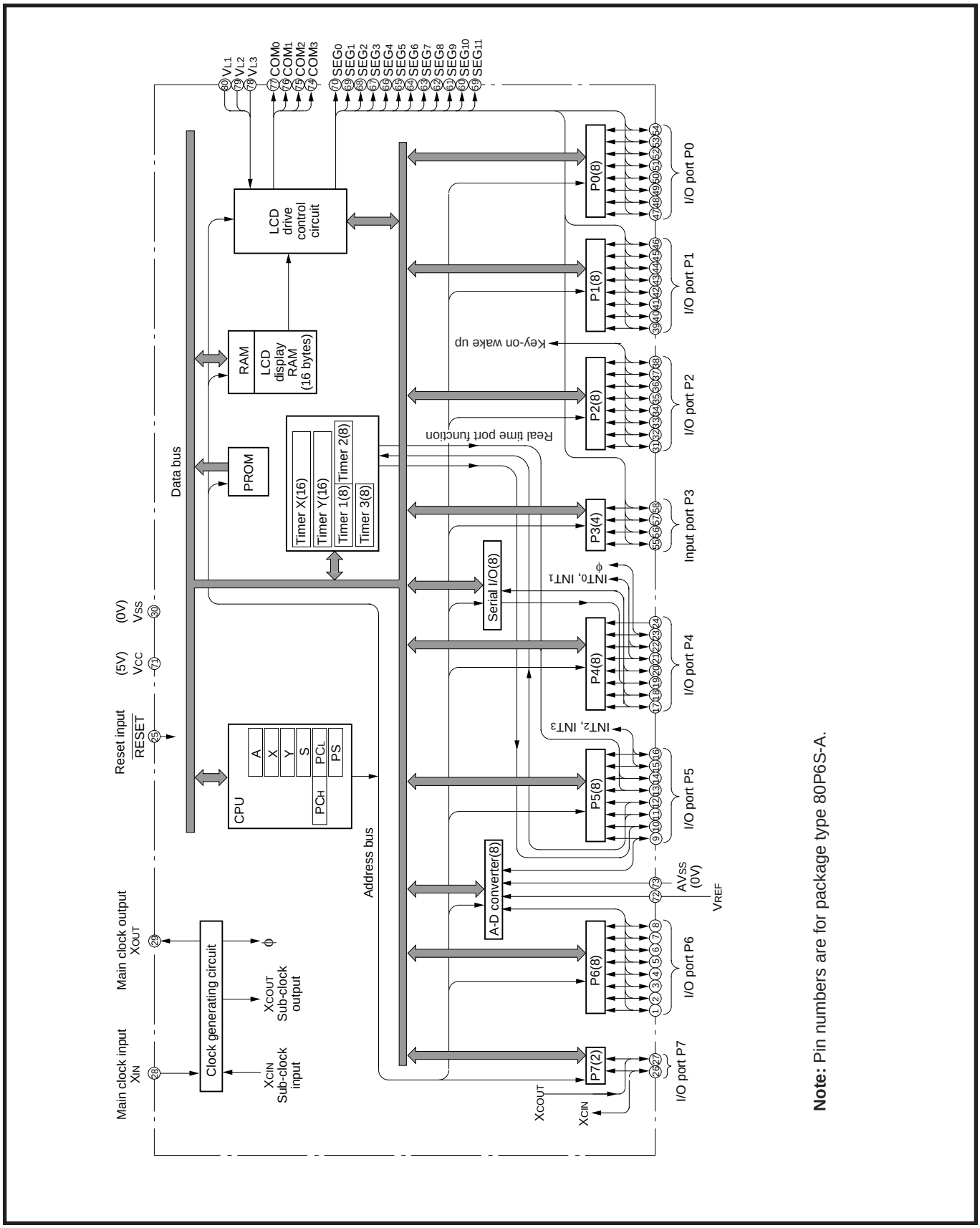


Fig. 3.1.3 Pin configuration of One Time PROM version (top view) (2)

3.1.4 Functional block diagram

Figure 3.1.4 shows the functional block diagram of the built-in PROM version.



Note: Pin numbers are for package type 80P6S-A.

Fig. 3.1.4 Functional block diagram of built-in PROM version

APPENDIX

3.1 Built-in PROM version

3.1.5 Notes on use

Notes on using the built-in PROM version are described below.

(1) All products of built-in PROM version

■Notes on programming

●When programming the contents of the PROM, use the dedicated programming adapter. This permits programming with a general-purpose PROM programmer.
At that time, set all of SW1, SW2 and SW3 in the above programming adapter to "OFF."

●As a high voltage is used for programming, be careful not to apply overvoltage to pins. Special care must be exercised at poweron.

■Notes on reading

When reading out the contents of the PROM, use the dedicated programming adapter as in programming. This permits reading out with a general-purpose PROM programmer.
At that time, set all of SW1, SW2 and SW3 in the programmer to "OFF."

■Notes on using port $P4_0$

When using port $P4_0$ as an input port in the One Time PROM/EPROM version, connect a resistors of several kΩ externally to port $P4_0$ in series. If this pin is not used, connect a resistor of several kΩ externally to V_{SS} in series (for improvement of the value withstand noise operation failure).
For details, refer to **"3.2 Countermeasures against noise, 3.2.1 Shortest wiring length, (3) Wiring to the V_{PP} pin of the One Time PROM version and the EPROM version."**

(2) EPROM Version

■Notes on deleting

●Sunlight and fluorescent lamps include light which may delete programmed information. For use in the read mode, cover the transparent glass part of the delete window with a seal or others.

●The seal to cover the transparent glass part is prepared by us.
This seal is metallic (aluminium) for reasons of prevention of information-deleting light and toughness. Be careful not to bring this seal into contact with lead pins of the microcomputer.

●Before deleting information, clean the transparent glass. Finger marks and seal paste may block ultraviolet rays and effect delete characteristics.

■Notes on mounting

●To mount the EPROM version for a purpose other than evaluation, use a suitable mounting socket. When mounting a ceramic package on the socket, fix it securely with silicone resin.

(3) One Time PROM version

■Notes on setting the PROM programmer area

●For products shipped in blank, access to the first 128 bytes and addresses FFFE16 and FFFF16 in the built-in PROM user area is inhibited.
Note the above point when setting the PROM programmer area.

■Notes before actual use

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 3.1.5 before actual use are recommended.

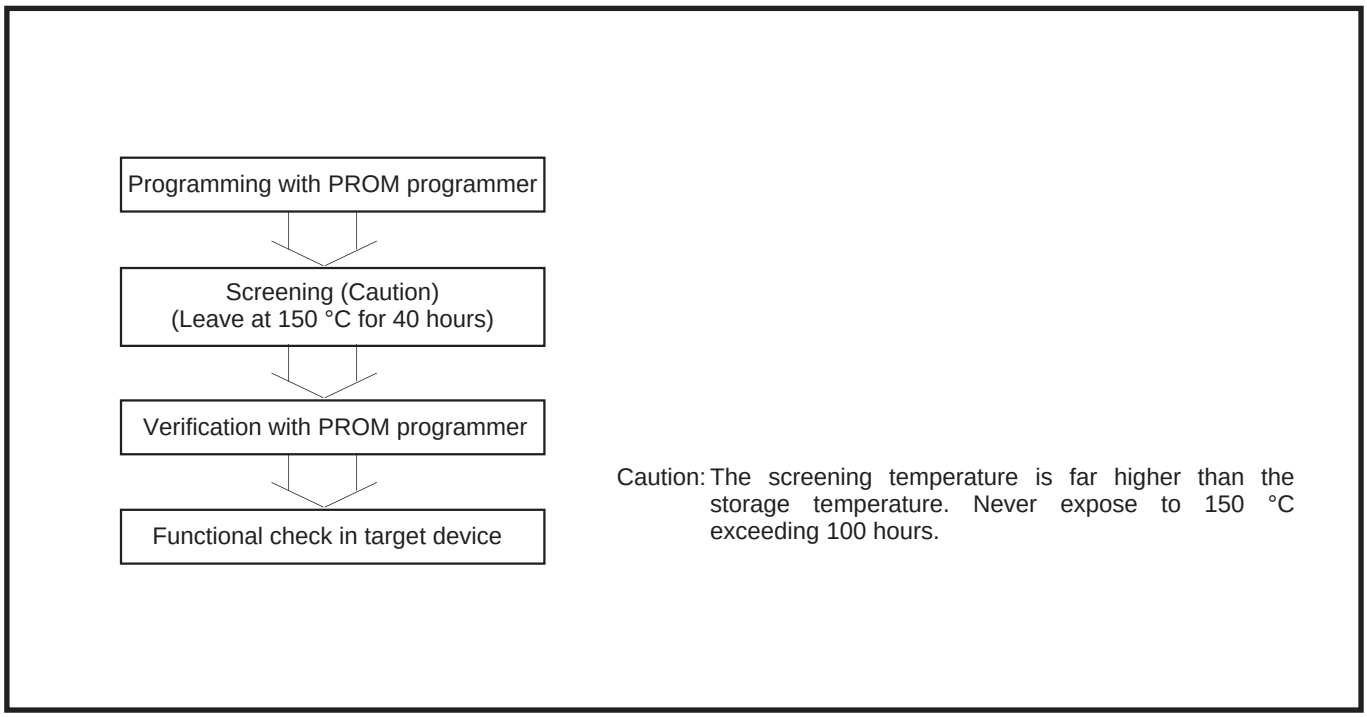


Fig. 3.1.5 Programming and testing of One Time PROM version (shipped in blank)

3.2 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.

3.2.1 Shortest wiring length

The wiring on a printed circuit board can function as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

(1) Wiring for the reset input pin

Make the length of wiring which is connected to the RESET input pin as short as possible. Especially, connect a capacitor across the RESET input pin and the VSS pin with the shortest possible wiring (within 20 mm).

Reason

The reset works to initialize the internal state of a microcomputer.

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

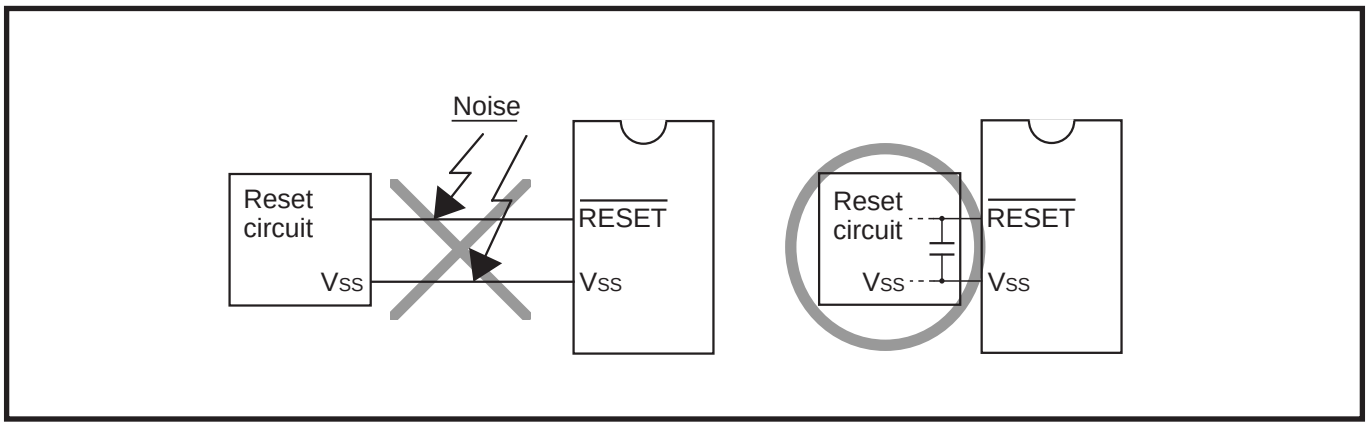


Fig. 3.2.1 Wiring for the RESET input pin

(2) Wiring for clock input/output pins

- Make the length of wiring which is connected to clock I/O pins as short as possible.
- Make the length of wiring (within 20 mm) across the grouding 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.

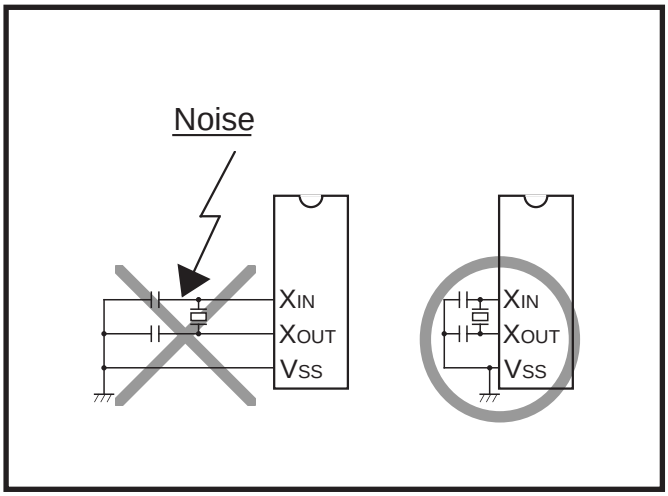


Fig. 3.2.2 Wiring for clock I/O pins

Reason

A microcomputer's operation synchronizes with a clock generated by the oscillator (circuit). 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.

(3) Wiring to the VPP pin of the One Time PROM version and the EPROM version

<When the VSS pin is also used as any other pin than the CNVSS*1 >

- Make the length of wiring which is connected to the VPP pin as short as possible.
- Connect an approximately 5 kW resistor to the VPP pin in serial (refer to **Figure 3.2.3**).

*1 When a microcomputer does not have the CNVSS pin, the VPP pin is also as the input pin adjacent to the RESET input pin.

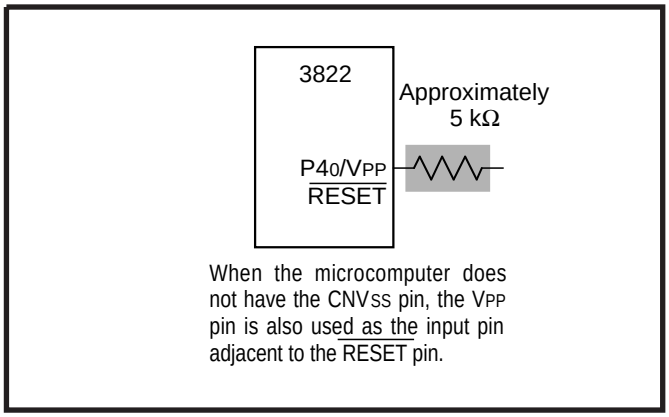


Fig. 3.2.3 Wiring for the VPP pin of the One Time PROM and the EPROM version

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.

3.2.2 Connection of a bypass capacitor across the Vss line and the 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.

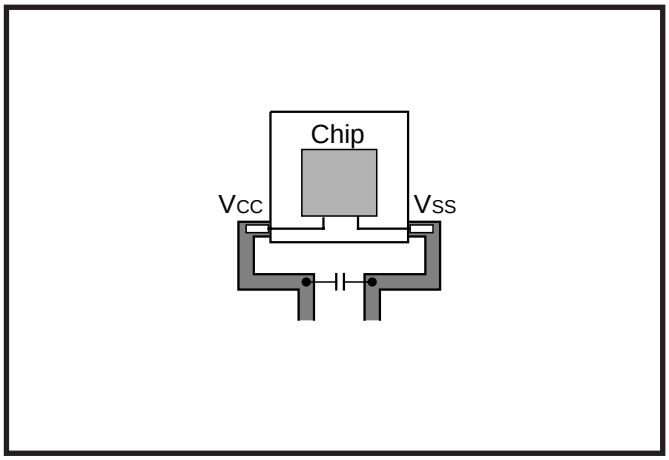


Fig. 3.2.4 Bypass capacitor across the Vss line and the Vcc line

3.2 Countermeasures against noise

3.2.3 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.

Reason

Signals which is input in an analog input pin (such as an A-D converter 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.

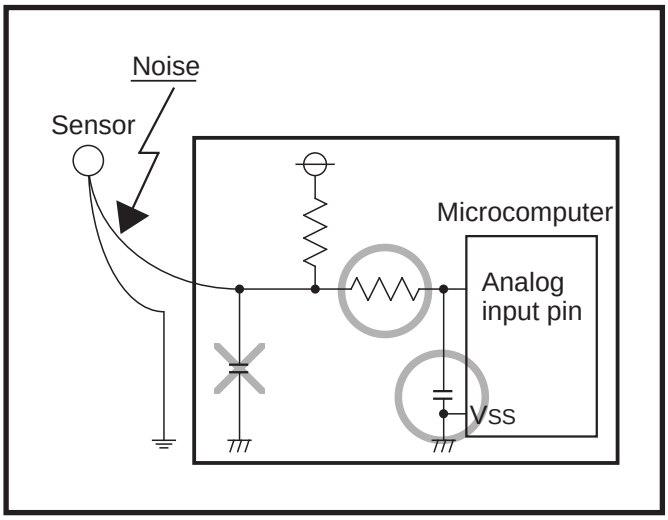


Fig. 3.2.5 Analog signal line and a resistor and a capacitor

3.2.4 Oscillator concerns

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

(1) Installing an 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.

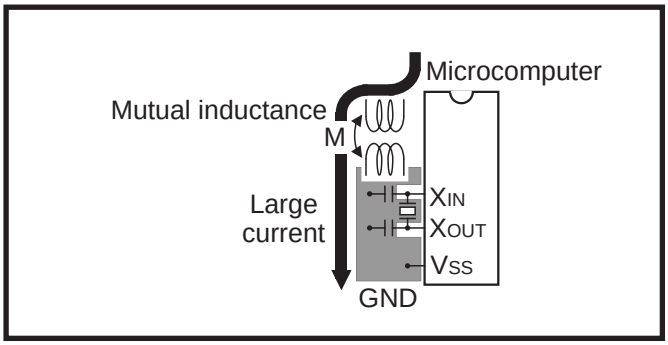


Fig. 3.2.6 Wiring for a large current signal line

3.2.5 Installing an oscillator away from signal lines where potential levels change frequently

Install 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.

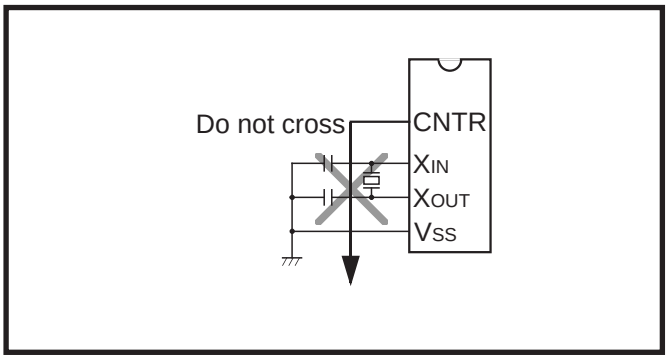


Fig. 3.2.7 Wiring to a signal line where potential levels change frequently

3.2.6 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.

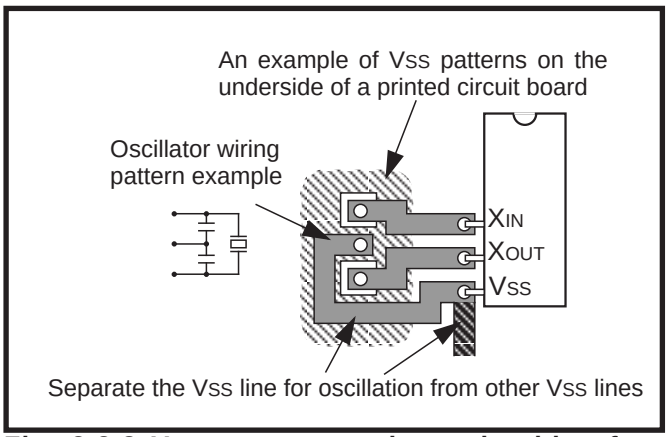


Fig. 3.2.8 Vss pattern on the underside of an oscillator

3.2.7 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 data register at fixed periods.
- Rewirte data to direction registers and pull-up control registers (only the product having it) at fixed periods.

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.

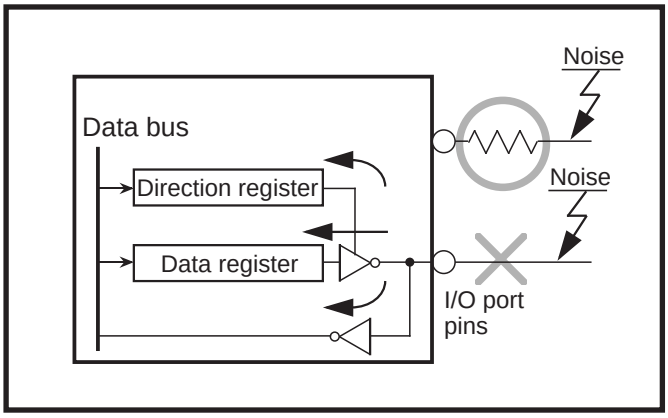


Fig. 3.2.9 Setup for I/O ports

3.2.8 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 cases:
  ①If the SWDT contents do not change after interrupt processing
  ②If the changed SWDT contents are abnormal (In Figure 21, the main routine determines that the interrupt processing routine has failed only if the SWDT contents do not change).

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determins 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:
  ①If the SWDT contents are not initialized to the initial value N but continued to decrement and if they exceed the limit (and reach 0 or less)

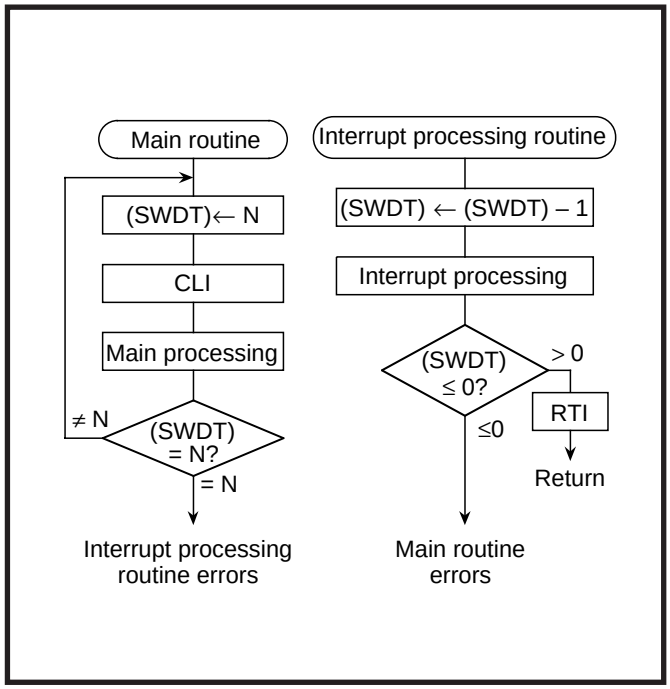


Fig. 3.2.10 Watchdog timer by software

3.3 Control registers

Port P0 direction register, port P1 direction register

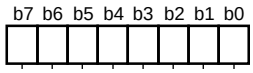
Port P0 direction register (P0D) [Address 01 16]
Port P1 direction register (P1D) [Address 03 16]

B	Name	Functions	At reset	R	W
0	Port P0 direction register / Port P1 direction register	0 : All bits are input mode 1 : All bits are output mode	0	×	○
1 to 7	Nothing is allocated. These bits cannot be written to and be read out.		0	×	×

Note: In ports P0 and P1, input/output switching is performed by a port unit. By setting bit 0 of the corresponding port direction register to "0", the port is set for the input mode. By setting to "1", the port is set for the output mode. Nothing is allocated for bits 1 to 7 of the port P0 direction register, and these bits cannot be written to.

Fig. 3.3.1 Structure of port P0 and P1 direction registers

Port Pi direction register



Port Pi direction register (PiD) (i = 2, 4 to 7)
[Address 05_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, $0F_{16}$]

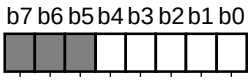
B	Name	Functions	At reset	R	W
0	Port Pi direction register	0 : Port Pi_0 input mode 1 : Port Pi_0 output mode	0	×	○
1		0 : Port Pi_1 input mode 1 : Port Pi_1 output mode	0	×	○
2		0 : Port Pi_2 input mode 1 : Port Pi_2 output mode	0	×	○
3		0 : Port Pi_3 input mode 1 : Port Pi_3 output mode	0	×	○
4		0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

Notes 1: Nothing is allocated bit 0 of port P4 direction register and bit 2 to bit 7 of port P7 direction register. These bits cannot be written to.
2: The contents of the port Pi direction register cannot be read out (refer to **"2.1.4 Notes on use"**)

Fig. 3.3.2 Structure of port Pi (i = 2, 4 to 7) direction registers

3.3 Control registers

PULL register A



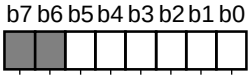
PULL register A (PULLA) [Address 16 16]

B	Name	Functions	At reset	R	W
0	Port $P0_0$–$P0_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
1	Port $P1_0$–$P1_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
2	Port $P2_0$–$P2_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
3	Port $P3_0$–$P3_7$ pull-down bit	0 : No pull-down 1 : Pull-down	1	○	○
4	Port $P7_0$, $P7_1$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
5 to 7	Nothing is allocated. These bits cannot be written to and are fixed to “0” at reading.		0	0	×

Note: For ports set for the output mode, pull-up or pull-down is impossible.

Fig. 3.3.3 Structure of PULL register A

PULL register B



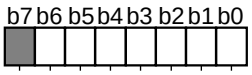
PULL register B (PULLB) [Address 17 16]

B	Name	Functions	At reset	R	W
0	Port $P4_1$-$P4_3$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
1	Port $P4_4$-$P4_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
2	Port $P5_0$-$P5_3$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
3	Port $P5_4$-$P5_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
4	Port $P6_0$-$P6_3$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
5	Port $P6_4$-$P6_7$ pull-up bit	0 : No pull-up 1 : Pull-up	0	○	○
6, 7	Nothing is allocated. These bits cannot be written to and are fixed to “0” at reading.		0	0	×

Note: For ports set for the output mode, pull-up is impossible.

Fig. 3.3.4 Structure of PULL register B

Serial I/O status register



Serial I/O status register (SIOSTS) [Address 19 16]

B	Name	Functions	At reset	R	W
0	Transmit buffer empty flag (TBE)	0: Buffer full 1: Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0: Buffer empty 1: Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0: Transmit shift in progress 1: Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0: No error 1: Overrun error	0	○	×
4	Parity error flag (PE)	0: No error 1: Parity error	0	○	×
5	Framing error flag (FE)	0: No error 1: Framing error	0	○	×
6	Summing error flag (SE)	0: (OE) U (PE) U (FE) = 0 1: (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is allocated. This bit cannot be written to and is fixed to “1” at reading.		1	1	×

Fig. 3.3.5 Structure of serial I/O status register

3.3 Control registers

Serial I/O control register

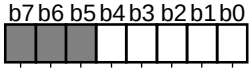
b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O control register (SIOCON) [Address 1A 16]

B	Name	Functions	At reset	R	W
0	BRG count source selection bit (CSS)	0: f(XIN) 1: f(XIN)/4	0	○	○
1	Serial I/O synchronization clock selection bit (SCS)	•In clock synchronous mode 0: BRG output/4 1: External clock input •In UART mode 0: BRG output/16 1: External clock input/16	0	○	○
2	$\overline{\text{SRDY}}$ output enable bit (SRDY)	0: P47/$\overline{\text{SRDY}}$ pin operates as I/O port P47 1: P47/$\overline{\text{SRDY}}$ pin operates as signal output pin $\overline{\text{SRDY}}$ ($\overline{\text{SRDY}}$ signal indicates receive enable state)	0	○	○
3	Transmit interrupt source selection bit (TIC)	0: When transmit buffer has emptied 1: When transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0: Transmit disabled 1: Transmit enabled	0	○	○
5	Receive enable bit (RE)	0: Receive disabled 1: Receive enabled	0	○	○
6	Serial I/O mode selection bit (SIOM)	0: Clock asynchronous serial I/O (UART) mode 1: Clock synchronous serial I/O mode	0	○	○
7	Serial I/O enable bit (SIOE)	0: Serial I/O disabled (pins P44–P47 operate as I/O pins) 1: Serial I/O enabled (pins P44–P47 operate as serial I/O pins)	0	○	○

Fig. 3.3.6 Structure of serial I/O control register

UART control register



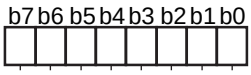
UART control register (UARTCON) [Address 1B 16]

B	Name	Functions	At reset	R	W
0	Character length selection bit (CHAS)	0: 8 bits 1: 7 bits	0	○	○
1	Parity enable bit (PARE)	0: Parity checking disabled 1: Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0: Even parity 1: Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0: 1 stop bit 1: 2 stop bits	0	○	○
4	$P4_5$/TxD P-channel output disable bit (POFF)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5 to 7	Nothing is allocated. These bits cannot be written to and are fixed to “1” at reading.		1	1	×

Fig. 3.3.7 Structure of UART control register

3.3 Control registers

Timer X mode register

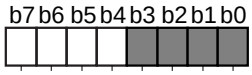


Timer X mode register (TXM) [Address 27 16]

B	Name	Functions	At reset	R	W
0	Timer X write control bit	0 : Write value in latch and counter 1 : Write value in latch only	0	○	○
1	Real time port control bit	0 : Real time port function invalid 1 : Real time port function valid	0	○	○
2	Data storage bit for real time port (RTP0)	0 : "L" level output 1 : "H" level output	0	○	○
3	Data storage bit for real time port (RTP1)	0 : "L" level output 1 : "H" level output	0	○	○
4	Timer X operating mode bits	b5b4 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
5			0	○	○
6	CNTR0 active edge switch bit	•CNTR0 interrupt 0 : Falling edge active 1 : Rising edge active •Pulse output mode 0 : Start at initial level "H" output 1 : Start at initial level "L" output •Event counter mode 0 : Rising edge active 1 : Falling edge active •Pulse width measurement mode 0 : Measure "H" level width 1 : Measure "L" level width	0	○	○
7	Timer X stop control bit	0 : Count start 1 : Count stop	0	○	○

Fig. 3.3.8 Structure of timer X mode register

Timer Y mode register



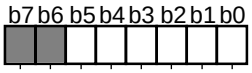
Timer Y mode register (TYM) [Address 28 16]

B	Name	Functions	At reset	R	W
0 to 3	Nothing is allocated. These bits are fixed to "0" at reading.		0	0	×
4	Timer Y operating mode bits	b5b4 0 0 : Timer mode 0 1 : Period measurement mode 1 0 : Event counter mode 1 1 : Pulse width HL continuously measurement mode	0	○	○
5			0	○	○
6	CNTR1 active edge switch bit	•CNTR1 interrupt 0 : Falling edge active 1 : Rising edge active •Period measurement mode 0 : Measure falling edge to falling edge 1 : Measure rising edge to rising edge •Event counter mode 0 : Rising edge active 1 : Falling edge active	0	○	○
7	Timer Y stop control bit	0 : Count start 1 : Count stop	0	○	○

Fig. 3.3.9 Structure of timer Y mode register

3.3 Control registers

Timer 123 mode register



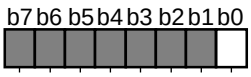
Timer 123 mode register (T123M) [Address 29 16]

B	Name	Functions	At reset	R	W
0	Timer 2 TOUT output active edge switch bit	0 : Start at "H" output 1 : Start at "L" output	0	○	○
1	Timer 2 TOUT output control bit	0 : TOUT output disabled 1 : TOUT output enabled	0	○	○
2	Timer 2 write control bit	0 : Write data in latch and counter 1 : Write data in latch only	0	○	○
3	Timer 2 count source selection bit	0 : Timer 1 underflow 1 : f(XIN)/16 (Middle-/high-speed mode) f(XCIN)/16 (Low-speed mode) (Note)	0	○	○
4	Timer 3 count source selection bit	0 : Timer 1 underflow 1 : f(XIN)/16 (Middle-/high-speed mode) f(XCIN)/16 (Low-speed mode) (Note)	0	○	○
5	Timer 1 count source selection bit	0 : f(XIN)/16 (Middle-/high-speed mode) f(XCIN)/16 (Low-speed mode) (Note) 1 : f(XCIN)	0	○	○
6, 7	Nothing is allocated. These bits cannot be written to and are fixed to "0" at reading.		0	0	×

Note: Internal clock φ is f(XCIN)/2 in the low-speed mode.

Fig. 3.3.10 Structure of timer 123 mode register

φ output control register

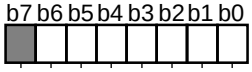


φ output control register (CKOUT) [Address 2A 16]

B	Name	Functions	At reset	R	W
0	φ output control bit	0: Port function 1: φ clock output (Port direction register = "1")	0	○	○
1 to 7	Nothing is allocated. These bits cannot be written to and are fixed to "0" at reading.		0	0	×

Fig. 3.3.11 Structure of φ output control register

A-D control register



A-D control register (ADCON) [Address 34 16]

B	Name	Functions	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0: AN0 0 0 1: AN1 0 1 0: AN2 0 1 1: AN3 1 0 0: AN4 1 0 1: AN5 1 1 0: AN6 1 1 1: AN7	0	○	○
1			0	○	○
2			0	○	○
3	A-D conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
4	VREF input switch bit	0: OFF 1: ON	0	○	○
5	A-D external trigger valid bit	0: A-D external trigger invalid (internal trigger selected) 1: A-D external trigger valid (external trigger selected)	0	○	○
6	Interrupt source selection bit	0: At A-D conversion completed 1: At falling of ADT pin input	0	○	○
7	Nothing is allocated. This bit cannot be written to and is fixed “0” at reading.		0	0	×

Note: When an internal trigger is selected, A-D conversion is started by setting bit 3 to “0.” Only writing “0” to bit 3 is valid. Even if “1” is written to bit 3, it is not set.
Accordingly, to write value to the ADCON without affecting bit 3, set bit 3 to “1.”

Fig. 3.3.12 Structure of A-D control register

3.3 Control registers

Segment output enable register

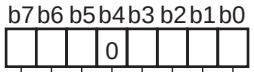
b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Segment output enable register (SEG) [Address 38 16]

B	Name	Functions	At reset	R	W
0	Segment output enable bit 0	0: Input ports P34–P37 1: Segment output SEG12–SEG15	0	○	○
1	Segment output enable bit 1	0: I/O ports P00, P01 1: Segment output SEG16, SEG17	0	○	○
2	Segment output enable bit 2	0: I/O ports P02–P07 1: Segment output SEG18–SEG23	0	○	○
3	Segment output enable bit 3	0: I/O ports P10, P11 1: Segment output SEG24, SEG25	0	○	○
4	Segment output enable bit 4	0: I/O ports P12 1: Segment output SEG26	0	○	○
5	Segment output enable bit 5	0: I/O ports P13–P17 1: Segment output SEG27–SEG31	0	○	○
6,7	Fix these bits to “0.”		0	0	0

Fig. 3.3.13 Structure of segment output register

LCD mode register



B	Name	Functions	At reset	R	W
0	Duty ratio selection bits	b1b0 00: Not available 01: 2 (use COM_0, COM_1) 10: 3 (use COM_0–COM_2) 11: 4 (use COM_0–COM_3)	0	○	○
1			0	○	○
2	Bias control bit	0: 1/3 bias 1: 1/2 bias	0	○	○
3	LCD enable bit	0: LCD OFF 1: LCD ON	0	○	○
4	Fix this bit to “0.”		0	0	0
5	LCD circuit divider division ratio selection bits (Note 1)	b6b5 00: LCDCK count source 01: LCDCK count source/2 10: LCDCK count source/4 11: LCDCK count source/8	0	○	○
6					
7	LCDCK count source selection bit (Note 2)	0: $f(X_{CIN})/32$ 1: $f(X_{IN})/8192$	0	○	○

Notes 1: Reference values at $f(X_{IN})$ = 8 MHz
00: 977 Hz
01: 488 Hz
10: 244 Hz
11: 122 Hz
2: LCDCK is a clock for a LCD timing controller.

Fig. 3.3.14 Structure of LCD mode register

3.3 Control registers

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register (INTEDGE) [Address 3A 16]

B	Name	Functions	At reset	R	W
0	INT_0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT_1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	INT_2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	INT_3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4 to 7	Nothing is allocated. These bits cannot be written to and are fixed to "0" at reading.		0	0	×

Fig. 3.3.15 Structure of interrupt edge selection register

CPU mode register

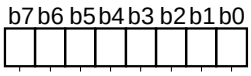
b7 b6 b5 b4 b3 b2 b1 b0

CPU mode register (CPUM) [Address 3B 16]

B	Name	Functions	At reset	R	W
0	Processor mode bits	b1b0 00: Single-chip mode 01: } Not available 10: } Not available 11: } Not available	0	○	○
1			0	○	○
2	Stack page selection bit	0: 0 page 1: 1 page	0	○	○
3	Fix this bit to "1."		1	1	1
4	Port X_C switch bit	0: I/O port 1: X_{CIN}, X_{COUT}	0	○	○
5	Main clock (X_{IN}–X_{OUT}) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bit	0: f(X_{IN})/2 (high-speed mode) 1: f(X_{IN})/8 (middle-speed mode)	1	○	○
7	Internal system clock selection bit	0: X_{IN}–X_{OUT} selected (middle-/high-speed mode) 1: X_{CIN}–X_{COUT} selected (low-speed mode)	0	○	○

Fig. 3.3.16 Structure of CPU mode register

Interrupt request register 1



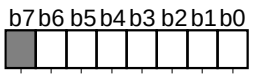
Interrupt request register 1 (IREQ1) [Address 3C 16]

B	Name	Functions	At reset	R	W
0	INT0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

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

Fig. 3.3.17 Structure of interrupt request register 1

Interrupt request register 2



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

B	Name	Functions	At reset	R	W
0	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key input interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	ADT/A-D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated. This bit cannot be written to and is fixed to "0" at reading.		0	0	×

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

Fig. 3.3.18 Structure of interrupt request register 2

3.3 Control registers

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1 (ICON1) [Address 3E 16]

B	Name	Functions	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	INT_1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	Timer 2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Timer 3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○

Fig. 3.3.19 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	Functions	At reset	R	W
0	$CNTR_0$ interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
1	$CNTR_1$ interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
2	Timer 1 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
3	INT_2 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
4	INT_3 interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
5	Key input interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
6	ADT/A-D conversion interrupt enable bit	0 : Interrupts disabled 1 : Interrupts enabled	0	○	○
7	Fix this bit to “0.”		0	0	0

Fig. 3.3.20 Structure of interrupt control register 2

3.4 List of instruction codes

D7 – D4 \ D3 – D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
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 (Note)	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 (Note)	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	SLW (Note) 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	FST (Note) DIV	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

3.5 Machine instructions

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	#
ADC (Note 1) (Note 5)	When T = 0 A ← A + M + C When T = 1 M(X) ← M(X) + M + C	Adds the carry, accumulator and memory contents. The results are entered into the accumulator. Adds the contents of the memory in the address indicated by index register X, the contents of the memory specified by the addressing mode and the carry. The results are entered into the memory at the address indicated by index register X.				69	2	2							65	3	2			
AND (Note 1)	When T = 0 A ← A ∧ M When T = 1 M(X) ← M(X) ∧ M	"AND's" the accumulator and memory contents. The results are entered into the accumulator. "AND's" the contents of the memory of the address indicated by index register X and the contents of the memory specified by the addressing mode. The results are entered into the memory at the address indicated by index register X.				29	2	2							25	3	2			
ASL	7 0 C←□←0	Shifts the contents of accumulator or contents of memory one bit to the left. The low order bit of the accumulator or memory is cleared and the high order bit is shifted into the carry flag.							0A	2	1				06	5	2			
BBC (Note 4)	A_b or M_b = 0?	Branches when the contents of the bit specified in the accumulator or memory is "0".										13+2i	4	2				17+2i	5	3
BBS (Note 4)	A_b or M_b = 1?	Branches when the contents of the bit specified in the accumulator or memory is "1".										03+2i	4	2				07+2i	5	3
BCC (Note 4)	C = 0?	Branches when the contents of carry flag is "0".																		
BCS (Note 4)	C = 1?	Branches when the contents of carry flag is "1".																		
BEQ (Note 4)	Z = 1?	Branches when the contents of zero flag is "1".																		
BIT	A ∧ M	"AND's" the contents of accumulator and memory. The results are not entered anywhere.													24	3	2			
BMI (Note 4)	N = 1?	Branches when the contents of negative flag is "1".																		
BNE (Note 4)	Z = 0?	Branches when the contents of zero flag is "0".																		
BPL (Note 4)	N = 0?	Branches when the contents of negative flag is "0".																		
BRA	PC ← PC ± offset	Jumps to address specified by adding offset to the program counter.																		
BRK	B ← 1 M(S) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 M(S) ← PS S ← S – 1 PC_L ← AD_L PC_H ← AD_H	Executes a software interrupt.	00	7	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
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																									M7	M6	•	•	•	•	Z	•
																											30	2	2				•	•	•	•	•	•	•	•
																											D0	2	2				•	•	•	•	•	•	•	•
																											10	2	2				•	•	•	•	•	•	•	•
																											80	4	2				•	•	•	•	•	•	•	•
																																	•	•	•	1	•	1	•	•

3.5 Machine instructions

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	#
BVC (Note 4)	V = 0?	Branches when the contents of overflow flag is "0."																		
BVS (Note 4)	V = 1?	Branches when the contents of overflow flag is "1."																		
CLB	A_b or $M_b \leftarrow 0$	Clears the contents of the bit specified in the accumulator or memory to "0."										1B + 2i	2	1				1F + 2i	5	2
CLC	C ← 0	Clears the contents of the carry flag to "0."	18	2	1															
CLD	D ← 0	Clears the contents of decimal mode flag to "0."	D8	2	1															
CLI	I ← 0	Clears the contents of interrupt disable flag to "0."	58	2	1															
CLT	T ← 0	Clears the contents of index X mode flag to "0."	12	2	1															
CLV	V ← 0	Clears the contents of overflow flag to "0."	B8	2	1															
CMP (Note 3)	When T = 0 A – M When T = 1 M(X) – M	Compares the contents of accumulator and memory. Compares the contents of the memory specified by the addressing mode with the contents of the address indicated by index register X.				C9	2	2							C5	3	2			
COM	$M \leftarrow \overline{M}$	Forms a one's complement of the contents of memory, and stores it into memory.													44	5	2			
CPX	X – M	Compares the contents of index register X and memory.				E0	2	2							E4	3	2			
CPY	Y – M	Compares the contents of index register Y and memory.				C0	2	2							C4	3	2			
DEC	A ← A – 1 or M ← M – 1	Decrements the contents of the accumulator or memory by 1.							1A	2	1				C6	5	2			
DEX	X ← X – 1	Decrements the contents of index register X by 1.	CA	2	1															
DEY	Y ← Y – 1	Decrements the contents of index register Y by 1.	88	2	1															
DIV	A ← (M(zz + X + 1), M(zz + X)) / A M(S) ← 1's complememt of Remainder S ← S – 1	Divides the 16-bit data that is the contents of M (zz + x + 1) for high byte and the contents of M (zz + x) for low byte by the accumulator. Stores the quotient in the accumulator and the 1's complement of the remainder on the stack.																		
EOR (Note 1)	When T = 0 A ← A ∀ M When T = 1 M(X) ← M(X) ∀ M	"Exclusive-ORs" the contents of accumulator and memory. The results are stored in the accumulator. "Exclusive-ORs" the contents of the memory specified by the addressing mode and the contents of the memory at the address indicated by index register X. The results are stored into the memory at the address indicated by index register X.				49	2	2							45	3	2			
FST		Connects oscillator output to the XOUT pin.	E2	2	1															
INC	A ← A + 1 or M ← M + 1	Increments the contents of accumulator or memory by 1.							3A	2	1				E6	5	2			
INX	X ← X + 1	Increments the contents of index register X by 1.	E8	2	1															
INY	Y ← Y + 1	Increments the contents of index register Y by 1.	C8	2	1															

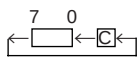
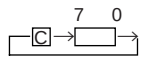
Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																											50	2	2				•	•	•	•	•	•	•	•
																											70	2	2				•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	0
																																	•	•	•	•	0	•	•	•
																																	•	•	•	•	•	0	•	•
																																	•	•	0	•	•	•	•	•
																																	•	0	•	•	•	•	•	•
D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
																																	N	•	•	•	•	•	Z	•
						EC	4	3																									N	•	•	•	•	•	Z	C
						CC	4	3																									N	•	•	•	•	•	Z	C
D6	6	2				CE	6	3	DE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
E2	16	2																															•	•	•	•	•	•	•	•
55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
JMP	If addressing mode is ABS $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is IND $PC_L \leftarrow M(AD_H, AD_L)$ $PC_H \leftarrow M(AD_H, AD_L + 1)$ If addressing mode is ZP, IND $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	Jumps to the specified address.																		
JSR	$M(S) \leftarrow PC_H$ $S \leftarrow S - 1$ $M(S) \leftarrow PC_L$ $S \leftarrow S - 1$ After executing the above, if addressing mode is ABS, $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is SP, $PC_L \leftarrow AD_L$ $PC_H \leftarrow FF$ If addressing mode is ZP, IND, $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	After storing contents of program counter in stack, and jumps to the specified address.																		
LDA (Note 2)	When T = 0 $A \leftarrow M$ When T = 1 $M(X) \leftarrow M$	Load accumulator with contents of memory. Load memory indicated by index register X with contents of memory specified by the addressing mode.				A9	2	2							A5	3	2			
LDM	$M \leftarrow nn$	Load memory with immediate value.													3C	4	3			
LDX	$X \leftarrow M$	Load index register X with contents of memory.				A2	2	2							A6	3	2			
LDY	$Y \leftarrow M$	Load index register Y with contents of memory.				A0	2	2							A4	3	2			
LSR	7 0 0→□→C	Shift the contents of accumulator or memory to the right by one bit. The low order bit of accumulator or memory is stored in carry, 7th bit is cleared.							4A	2	1				46	5	2			
MUL (Note 5)	$M(S) \cdot A \leftarrow A \times M(zz + X)$ $S \leftarrow S - 1$	Multiplies the accumulator with the contents of memory specified by the zero page X addressing mode and stores the high byte of the result on the stack and the low byte in the accumulator.																		
NOP	$PC \leftarrow PC + 1$	No operation.	EA	2	1															
ORA (Note 1)	When T = 0 $A \leftarrow A \vee M$ When T = 1 $M(X) \leftarrow M(X) \vee M$	"Logical OR's" the contents of memory and accumulator. The result is stored in the accumulator. "Logical OR's" the contents of memory indicated by index register X and contents of memory specified by the addressing mode. The result is stored in the memory specified by index register X.				09	2	2							05	3	2			

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
						4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
						20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
			B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•
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	•

APPENDIX

3.5 Machine instructions

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	#
PHA	M(S) ← A S ← S – 1	Saves the contents of the accumulator in memory at the address indicated by the stack pointer and decrements the contents of stack pointer by 1.	48	3	1															
PHP	M(S) ← PS S ← S – 1	Saves the contents of the processor status register in memory at the address indicated by the stack pointer and decrements the contents of the stack pointer by 1.	08	3	1															
PLA	S ← S + 1 A ← M(S)	Increments the contents of the stack pointer by 1 and restores the accumulator from the memory at the address indicated by the stack pointer.	68	4	1															
PLP	S ← S + 1 PS ← M(S)	Increments the contents of stack pointer by 1 and restores the processor status register from the memory at the address indicated by the stack pointer.	28	4	1															
ROL		Shifts the contents of the memory or accumulator to the left by one bit. The high order bit is shifted into the carry flag and the carry flag is shifted into the low order bit.							2A	2	1				26	5	2			
ROR		Shifts the contents of the memory or accumulator to the right by one bit. The low order bit is shifted into the carry flag and the carry flag is shifted into the high order bit.							6A	2	1				66	5	2			
RRF		Rotates the contents of memory to the right by 4 bits.													82	8	2			
RTI	S ← S + 1 PS ← M(S) S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	Returns from an interrupt routine to the main routine.	40	6	1															
RTS	S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	Returns from a subroutine to the main routine.	60	6	1															
SBC (Note 1) (Note 5)	When T = 0 A ← A – M – $\overline{C}$ When T = 1 M(X) ← M(X) – M – $\overline{C}$	Subtracts the contents of memory and complement of carry flag from the contents of accumulator. The results are stored into the accumulator. Subtracts contents of complement of carry flag and contents of the memory indicated by the addressing mode from the memory at the address indicated by index register X. The results are stored into the memory of the address indicated by index register X.				E9	2	2							E5	3	2			
SEB	Ab or Mb ← 1	Sets the specified bit in the accumulator or memory to “1.”										0B + 2i	2	1				0F + 2i	5	2
SEC	C ← 1	Sets the contents of the carry flag to “1.”	38	2	1															
SED	D ← 1	Sets the contents of the decimal mode flag to “1.”	F8	2	1															
SEI	I ← 1	Sets the contents of the interrupt disable flag to “1.”	78	2	1															
SET	T ← 1	Sets the contents of the index X mode flag to “1.”	32	2	1															
SLW		Disconnects the oscillator output from the XOUT pin.	C2	2	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
																															•	•	•	•	•	•	•	•
																															•	•	•	•	•	•	•	•
																															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)							
																															•	•	•	•	•	•	•	•
F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
																															•	•	•	•	•	•	•	•
																															•	•	•	•	•	•	•	1
																															•	•	•	•	1	•	•	•
																															•	•	•	•	•	1	•	•
																															•	•	1	•	•	•	•	•
																															•	•	•	•	•	•	•	•

3.5 Machine instructions

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	#
STA	M ← A	Stores the contents of accumulator in memory.													85	4	2			
STP		Stops the oscillator.	42	2	1															
STX	M ← X	Stores the contents of index register X in memory.													86	4	2			
STY	M ← Y	Stores the contents of index register Y in memory.													84	4	2			
TAX	X ← A	Transfers the contents of the accumulator to index register X.	AA	2	1															
TAY	Y ← A	Transfers the contents of the accumulator to index register Y.	A8	2	1															
TST	M = 0?	Tests whether the contents of memory are “0” or not.													64	3	2			
TSX	X ← S	Transfers the contents of the stack pointer to index register X.	BA	2	1															
TXA	A ← X	Transfers the contents of index register X to the accumulator.	8A	2	1															
TXS	S ← X	Transfers the contents of index register X to the stack pointer.	9A	2	1															
TYA	A ← Y	Transfers the contents of index register Y to the accumulator.	98	2	1															
WIT		Stops the internal clock.	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.

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
95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
			96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	–	Subtraction
A	Accumulator or Accumulator addressing mode	∧	Logical OR
		∨	Logical AND
BIT, A	Accumulator bit relative addressing mode	∀	Logical exclusive OR
		—	Negation
ZP	Zero page addressing mode	←	Shows direction of data flow
BIT, ZP	Zero page bit relative addressing mode	X	Index register X
		Y	Index register Y
ZP, X	Zero page X addressing mode	S	Stack pointer
ZP, Y	Zero page Y addressing mode	PC	Program counter
ABS	Absolute addressing mode	PS	Processor status register
ABS, X	Absolute X addressing mode	PC_H	8 high-order bits of program counter
ABS, Y	Absolute Y addressing mode	PC_L	8 low-order bits of program counter
IND	Indirect absolute addressing mode	AD_H	8 high-order bits of address
		AD_L	8 low-order bits of address
ZP, IND	Zero page indirect absolute addressing mode	FF	FF in Hexadecimal notation
		nn	Immediate value
IND, X	Indirect X addressing mode	M	Memory specified by address designation of any addressing mode
IND, Y	Indirect Y addressing mode		
REL	Relative addressing mode	M(X)	Memory of address indicated by contents of index register X
SP	Special page addressing mode		
C	Carry flag	M(S)	Memory of address indicated by contents of stack pointer
Z	Zero flag		
I	Interrupt disable 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.
D	Decimal mode flag		
B	Break flag		
T	X-modified arithmetic mode flag	$M(00, AD_L)$	Contents of address indicated by zero page AD_L
V	Overflow flag	A_b	1 bit of accumulator
N	Negative flag	M_b	1 bit of memory
		OP	Opcode
		n	Number of cycles
		#	Number of bytes

3.6 Mask ROM ordering method

GZZ–SH06–60B < 2XB0 >

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38222M2-XXXFP/GP/HP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✻.

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

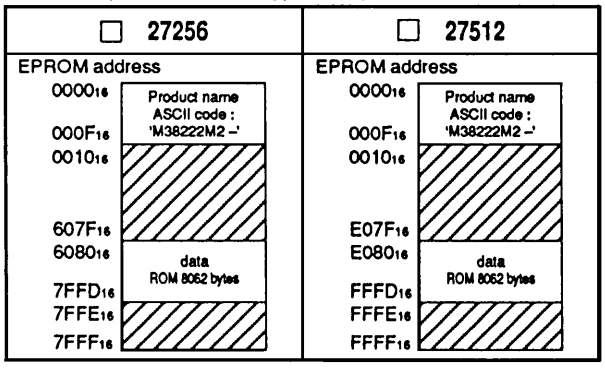
✻ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M38222M2-XXXFP ☐ M38222M2-XXXGP ☐ M38222M2-XXXHP

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

EPROM type (indicate the type used)



In the address space of the microcomputer, the internal ROM area is from address $E080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38222M2–" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = 4D 16	0008_{16}	'–' = 2D 16
0001_{16}	'3' = 33 16	0009_{16}	FF 16
0002_{16}	'8' = 38 16	$000A_{16}$	FF 16
0003_{16}	'2' = 32 16	$000B_{16}$	FF 16
0004_{16}	'2' = 32 16	$000C_{16}$	FF 16
0005_{16}	'2' = 32 16	$000D_{16}$	FF 16
0006_{16}	'M' = 4D 16	$000E_{16}$	FF 16
0007_{16}	'2' = 32 16	$000F_{16}$	FF 16

GZZ–SH06–60B < 2XB0 >

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38222M2-XXXFP/GP/HP
MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27256	27512
The pseudo-command	*=△$8000 .BYTE△ 'M38222M2–'	*=△$0000 .BYTE△ 'M38222M2–'

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

✻ 2. Mark specification
Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (80P6N for M38222M2-XXXFP, 80P6S for M38222M2-XXXGP, 80P6D for M38222M2-XXXHP) and attach it to the mask ROM confirmation form.

✻ 3. Usage conditions
Please answer the following questions about usage for use in our product inspection :
(1) How will you use the XIN-XOUT oscillator?

☐ Ceramic resonator ☐ Quartz crystal
☐ External clock input ☐ Other ( )

At what frequency? f(XIN) = ☐ MHz

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

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

✻ 4. Comments

(2/2)

GZZ-SH06-16B < 25B0 >

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38223M4-XXXFP/GP/HP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✻.

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

✻ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M38223M4-XXXFP ☐ M38223M4-XXXGP ☐ M38223M4-XXXHP

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

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Product name ASCII code : 'M38223M4 –'	0000_{16} Product name ASCII code : 'M38223M4 –'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$407F_{16}$	$C07F_{16}$
4080_{16} data ROM 16254 bytes	$C080_{16}$ data ROM 16254 bytes
$7FFD_{16}$	$FFFD_{16}$
$7FFE_{16}$	$FFFE_{16}$
$7FFF_{16}$	$FFFF_{16}$

In the address space of the microcomputer, the internal ROM area is from address $C080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38223M4–" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$.
The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	'–' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'2' = 32_{16}	$000B_{16}$	FF_{16}
0004_{16}	'2' = 32_{16}	$000C_{16}$	FF_{16}
0005_{16}	'3' = 33_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

GZZ–SH06–16B < 25B0 >

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38223M4-XXXFP/GP/HP
MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27256	27512
The pseudo-command	*=△$8000 .BYTE△ 'M38223M4–'	*=△$0000 .BYTE△ 'M38223M4–'

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

✻ 2. Mark specification
Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (80P6N for M38223M4-XXXFP, 80P6S for M38223M4-XXXGP, 80P6D for M38223M4-XXXHP) and attach it to the mask ROM confirmation form.

✻ 3. Usage conditions
Please answer the following questions about usage for use in our product inspection :
(1) How will you use the XIN-XOUT oscillator?

☐ Ceramic resonator ☐ Quartz crystal
☐ External clock input ☐ Other ( )

At what frequency? f(XIN) = ☐ MHz

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

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

✻ 4. Comments

(2/2)

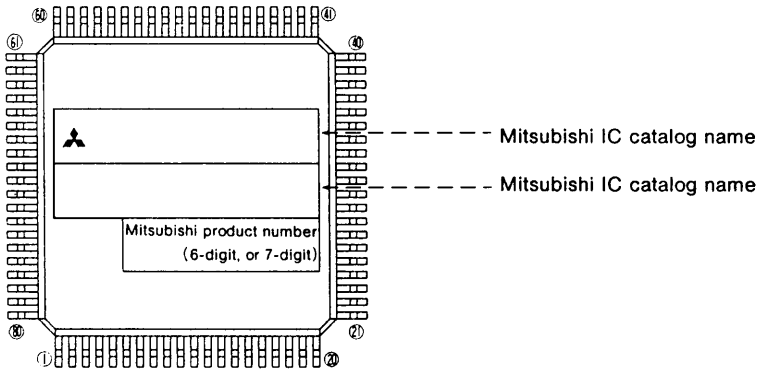
3.7 Mark specification form

80P6S (80-PIN QFP) MARK SPECIFICATION FORM
80P6D (80-PIN Fine-pitch QFP)

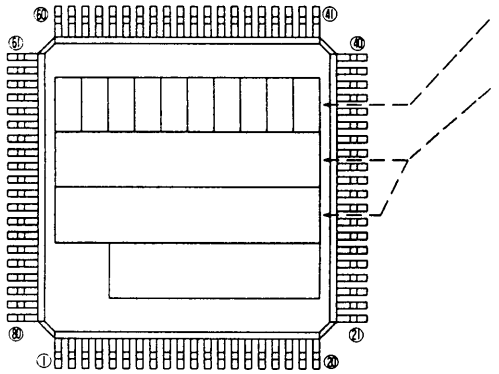
Mitsubishi IC catalog name

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

A. Standard Mitsubishi Mark



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

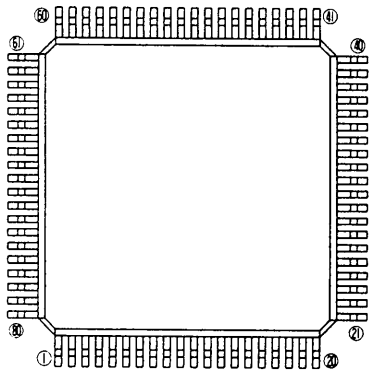


Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 10 alphanumeric characters for capital letters, hyphens, commas, periods and so on.
4 : If the Mitsubishi logo is not required, check the box below.

Mitsubishi logo is not required ☐

5 : The allocation of Mitsubishi IC catalog name and Mitsubishi product number is different on the package owing to the number of Mitsubishi IC catalog name's characters, and the requiring Mitsubishi logo or not.

C. Special Mark Required



Note1 : If special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated technically as close as possible. Mitsubishi product number (6-digit, or 7-digit) and Mask ROM number (3-digit) are always marked for sorting the products.
2 : If special character fonts (e.g., customer's trade mark logo) must be used in Special Mark, check the box below.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

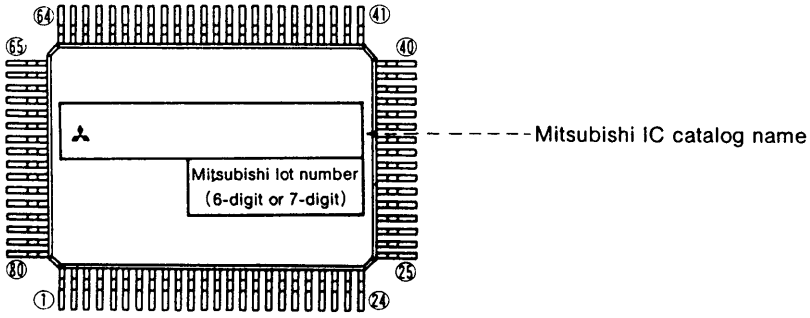
Special character fonts required ☐

80P6N (80-PIN QFP) MARK SPECIFICATION FORM

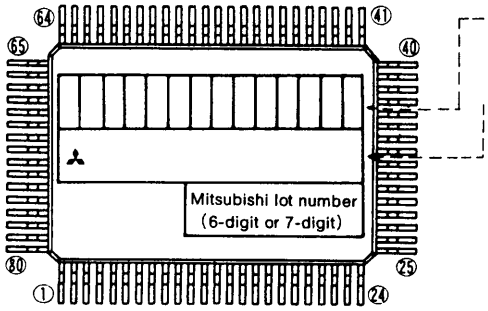
Mitsubishi IC catalog name ☐

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

A. Standard Mitsubishi Mark



B. Customer's Parts Number + Mitsubishi Catalog Name

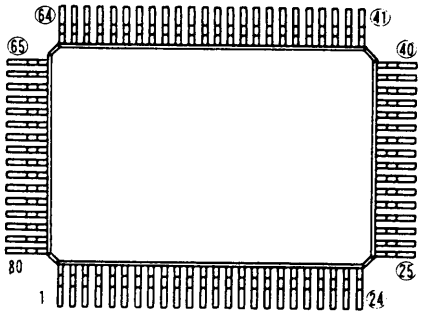


Customer's parts number
Note : The fonts and size of characters are standard Mitsubishi type.
Mitsubishi IC catalog name

Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.

Note3 : Customer's parts number can be up to 14 characters :
Only 0~9, A~Z, +, −, /, (,), &, ©, . (period), and , (comma) are usable.
4 : If the Mitsubishi logo is not required, check the box below.
Mitsubishi logo is not required ☐

C. Special Mark Required



Note1 : If the special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit or 7-digit) and mask ROM number (3-digit) are always marked.
2 : If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.
Special logo required ☐

The standard Mitsubishi font is used for all characters except for a logo.

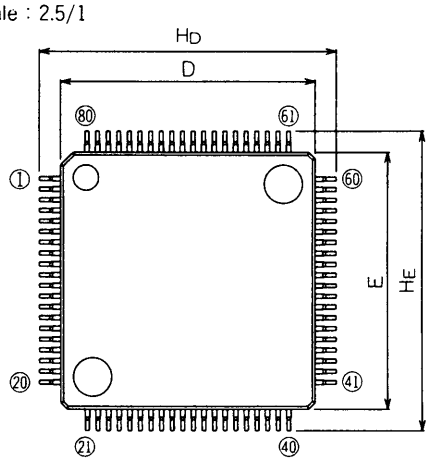
3.8 Package outlines

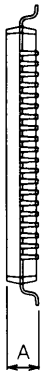
80P6D−A

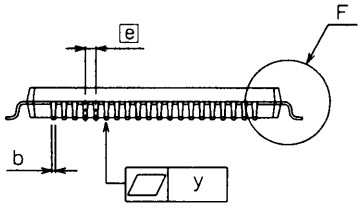
Plastic 80pin 12×12mm body QFP(FP)

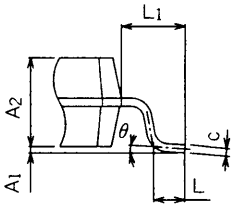
EIAJ Package Code	JEDEC Code	Weight(g)
*QFP80-P-1212	–	0.44

Scale : 2.5/1

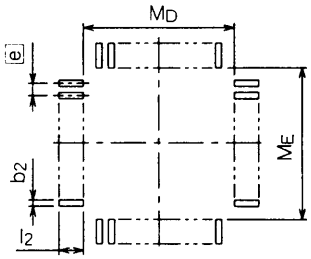








Detail F



Recommended Mount Pad

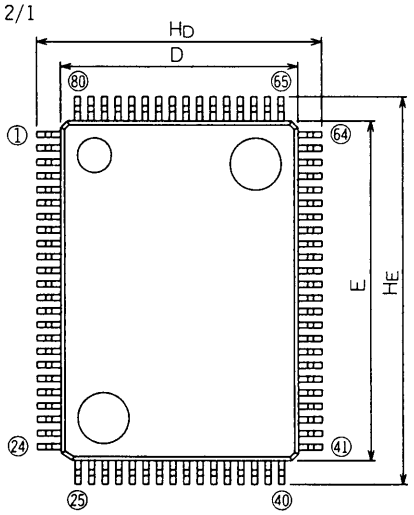
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A_1	0	0.1	0.2
A_2	–	1.4	–
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	11.9	12.0	12.1
E	11.9	12.0	12.1
e	–	0.5	–
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
L	0.3	0.5	0.7
L_1	–	1.0	–
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.225	–
l_2	1.0	–	–
M_D	–	12.4	–
M_E	–	12.4	–

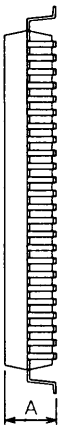
80P6N−A

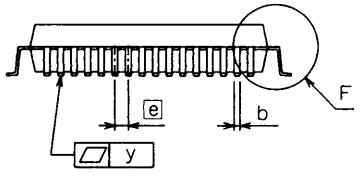
Plastic 80pin 14×20mm body QFP

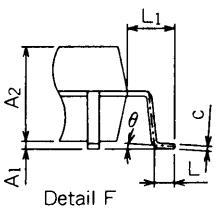
EIAJ Package Code	JEDEC Code	Weight(g)
*QFP080-P-1420	–	1.58

Scale : 2/1

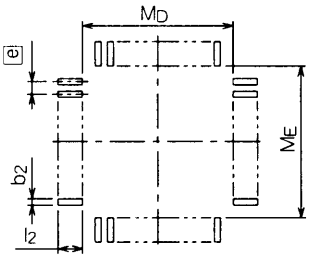








Detail F



Recommended Mount Pad

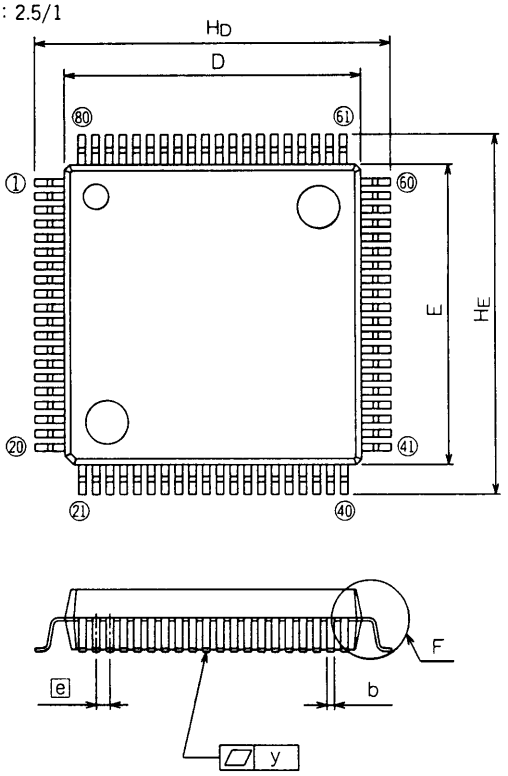
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A_1	0	0.1	0.2
A_2	–	2.8	–
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	–	0.8	–
H_D	16.5	16.8	17.1
H_E	22.5	22.8	23.1
L	0.4	0.6	0.8
L_1	–	1.4	–
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.5	–
l_2	1.3	–	–
M_D	–	14.6	–
M_E	–	20.6	–

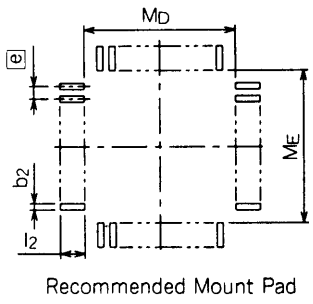
80P6S−A

Plastic 80pin 14×14mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)
＊QFP080-P-1414		1.11

Scale : 2.5/1





Recommended Mount Pad

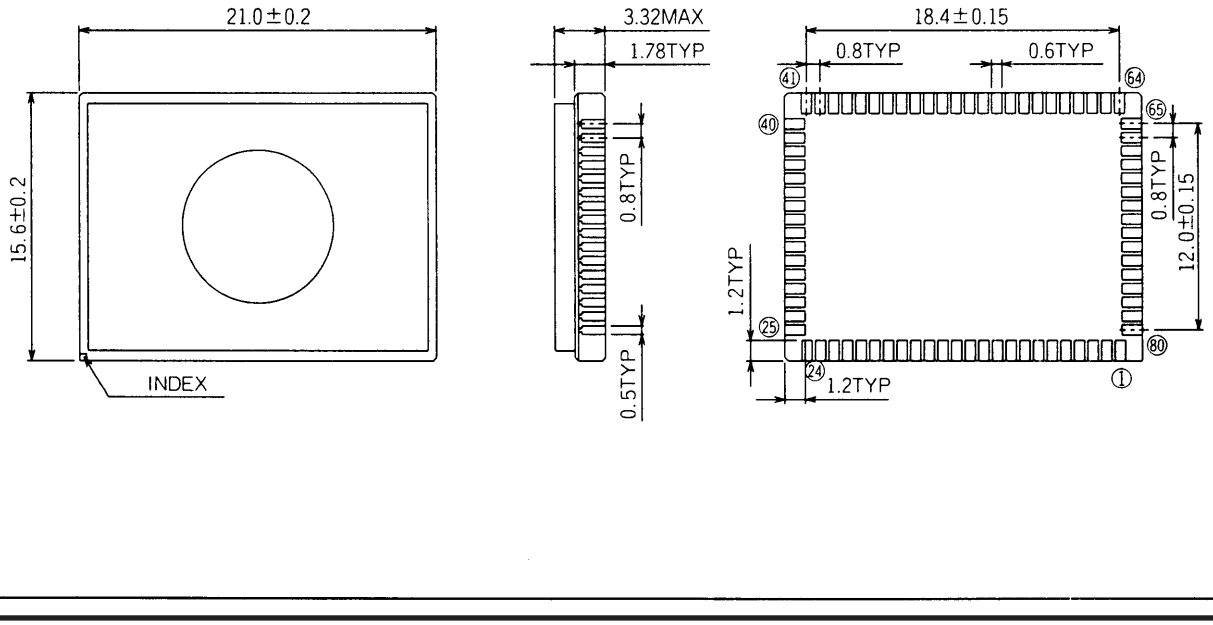
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	3.05
A_1	0	0.1	0.2
A_2	—	2.8	—
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	13.8	14.0	14.2
e	—	0.65	—
H_D	16.5	16.8	17.1
H_E	16.5	16.8	17.1
L	0.4	0.6	0.8
L_1	—	1.4	—
y	—	—	0.1
θ	0°	—	10°
b_2	—	0.35	—
l_2	1.3	—	—
M_D	—	14.6	—
M_E	—	14.6	—

80D0

Glass seal 80pin QFN(LCC)

EIAJ Package Code	JEDEC Code	Weight(g)
WQFN080-G	—	

Scale : 2/1



3.9 SFR allocation

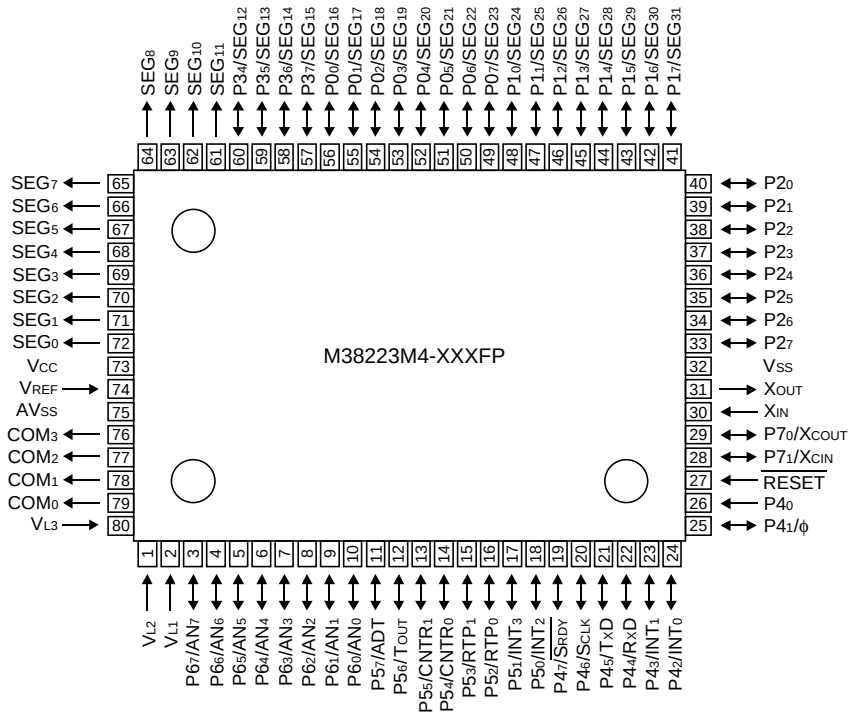
0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	Port P5 (P5)
$000B_{16}$	Port P5 direction register (P5D)
$000C_{16}$	Port P6 (P6)
$000D_{16}$	Port P6 direction register (P6D)
$000E_{16}$	Port P7 (P7)
$000F_{16}$	Port P7 direction register (P7D)
0010_{16}	
0011_{16}	
0012_{16}	
0013_{16}	
0014_{16}	
0015_{16}	
0016_{16}	PULL register A (PULLA)
0017_{16}	PULL register B (PULLB)
0018_{16}	Transmit/Receive buffer register(TB/RB)
0019_{16}	Serial I/O status register (SIOSTS)
$001A_{16}$	Serial I/O control register (SIOCON)
$001B_{16}$	UART control register (UARTCON)
$001C_{16}$	Baud rate generator (BRG)
$001D_{16}$	
$001E_{16}$	
$001F_{16}$	

0020_{16}	Timer X (low-order) (TXL)
0021_{16}	Timer X (high-order) (TXH)
0022_{16}	Timer Y (low-order) (TYL)
0023_{16}	Timer Y (high-order) (TYH)
0024_{16}	Timer 1 (T1)
0025_{16}	Timer 2 (T2)
0026_{16}	Timer 3 (T3)
0027_{16}	Timer X mode register (TXM)
0028_{16}	Timer Y mode register (TYM)
0029_{16}	Timer 123 mode register (T123M)
$002A_{16}$	ϕ output control register (CKOUT)
$002B_{16}$	
$002C_{16}$	
$002D_{16}$	
$002E_{16}$	
$002F_{16}$	
0030_{16}	
0031_{16}	
0032_{16}	
0033_{16}	
0034_{16}	A-D control register (ADCON)
0035_{16}	A-D conversion register (AD)
0036_{16}	
0037_{16}	
0038_{16}	Segment output enable register (SEG)
0039_{16}	LCD mode register (LM)
$003A_{16}$	Interrupt edge selection register (INTEDGE)
$003B_{16}$	CPU mode register (CPUM)
$003C_{16}$	Interrupt request register 1(IREQ1)
$003D_{16}$	Interrupt request register 2(IREQ2)
$003E_{16}$	Interrupt control register 1(ICON1)
$003F_{16}$	Interrupt control register 2(ICON2)

Memory map of special function register (SFR)

3.10 Pin configuration

PIN CONFIGURATION (TOP VIEW)



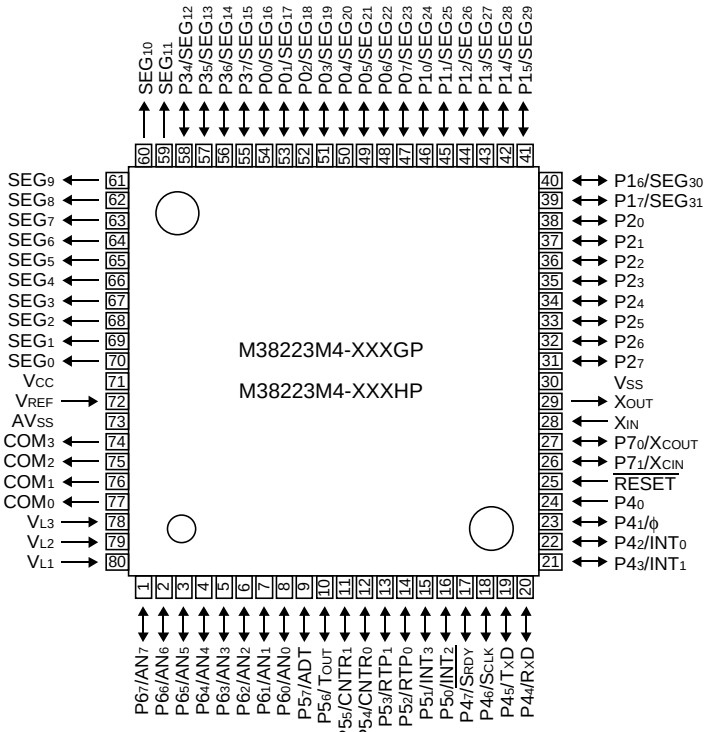
Package type : 80P6N-A
80-pin plastic molded QFP

Pin configuration of M38223M4-XXXFP

APPENDIX

3.10 Pin configuration

PIN CONFIGURATION (TOP VIEW)



Package type : 80P6S-A/80P6D-A
80-pin plastic-molded QFP

Pin configuration of M38223M4-XXXGP/HP

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
3822 Group

Mar. First Edition 1995

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

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

3822 Group
User's Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REVISION DESCRIPTION LIST	3822 Group User's Manual

Rev. No.	Revision Description	Rev. date
1.0	First Edition	980220

GRADE A

MESC TECHNICAL NEWS

No. M380-17-9910

Corrections and Supplementary Explanation for "3822 Group User's Manual"

This news includes all corrections and supplementary explanation for the 3822 Group User's Manual (1995.3 issued, document number: H-ED347-A) issued before.

Please refer to the corrected information as shown below.

- Corrections and supplementary explanation for the 3822 Group User's Manual issued before

  (A) M380-11-9507 Correction of "3822 Group User's Manual"
  (B) M380-14-9907 Corrections and Supplementary Explanation for "3820/3822/3825 Group User's Manuals

  Note: Alphabets in parentheses are corresponding to the alphabets in the parameter "REV." of the following corrections and supplementary explanation list. "Rev.C" is the new information added in this time.

Corrections and Supplementary Explanation for “3822 Group User’s Manuals” No.1

Rev.	Page	Contents
C	P1-20 (Left columun)	Error A key input interrupt request is generated by applying “L” level ... Correct A key input interrupt request is generated by detecting falling edge..
A	P1-22 line 7 (Right column)	Error (However, if the real time port control bit is changed from “0” to “1”, data are output without the timer X.) Correct (However, after rewriting a data storage bit for real time port, if the real time port control bit is changed from “0” to “1”, data is output without the timer X.)
A	P2-40 line 8	Error A data output from the real time port is started at setting the real time port control bit to “1”. Correct A data output from the real time port is started at setting the real time port control bit to “1” (when setting “1” to the real time port control bit of the timer X mode register, use the SEB instruction).
B	P2-65 (1) Timer X ■ Timer mode Fig. 2.3.22	Previous change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before setting ① below, clear the timer X interrupt enable bit and the timer X interrupt request bit to “0”. •After setting ④ below, set the timer X interrupt enable bit to “1” (interrupts enabled). After change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before timer X stops counting (before setting ① below), clear the timer X interrupt enable bit to “0”. •After setting ③ below, clear the timer X interrupt request bit to “0” and next set the timer X interrupt enable bit to “1” (interrupt enabled). •Set ④ last.
B	P2-66 (1) Timer X ■ Pulse output mode Fig. 2.3.23	Previous change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before setting ① below, clear the interrupt enable bits (timer X or $CNTR_0$) and the interrupt request bits (timer X or $CNTR_0$) to “0”. •After setting ⑤ below, set the interrupt enable bits (timer X or $CNTR_0$) to “1” (interrupts enabled). After change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before timer X stops counting (before setting ② below), clear the interrupt enable bit (timer X or $CNTR_0$) to “0”. •After setting ④ below, clear the interrupt request bit (timer X or $CNTR_0$) to “0” and next set the interrupt enable bit (timer X or $CNTR_0$) to “1” (interrupt enabled). •Set ⑤ last.

Corrections and Supplementary Explanation for “3822 Group User’s Manuals” No.2

Rev.	Page	Contents
B	P2-67 (1) Timer X ■ Event counter mode Fig. 2.3.24	Previous change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before setting ① below, clear the interrupt enable bits (timer X or CNTR0) and the interrupt request bits (timer X or CNTR0) to “0”. •After setting ⑤ below, set the interrupt enable bits (timer X or CNTR0) to “1” (interrupts enabled). After change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before timer X stops counting (before setting ② below), clear the interrupt enable bit (timer X or CNTR0) to “0”. •After setting ④ below, clear the interrupt request bit (timer X or CNTR0) to “0” and next set the interrupt enable bit (timer X or CNTR0) to “1” (interrupt enabled). •Set ⑤ last.
B	P2-68 (1) Timer X ■ Pulse width measurement mode Fig. 2.3.25	Previous change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before setting ① below, clear the interrupt enable bits (timer X or CNTR0) and the interrupt request bits (timer X or CNTR0) to “0”. •After setting ⑤ below, set the interrupt enable bits (timer X or CNTR0) to “1” (interrupts enabled). After change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before timer X stops counting (before setting ② below), clear the interrupt enable bit (timer X or CNTR0) to “0”. •After setting ④ below, clear the interrupt request bit (timer X or CNTR0) to “0” and next set the interrupt enable bit (timer X or CNTR0) to “1” (interrupt enabled). •Set ⑤ last.
B	P2-70 (2) Timer Y ■ Timer mode Fig. 2.3.27	Previous change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before setting ① below, clear the timer Y interrupt enable bit and the timer Y interrupt request bit to “0”. •After setting ④ below, set the timer Y interrupt enable bit to “1” (interrupts enabled). After change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before timer Y stops counting (before setting ① below), clear the timer Y interrupt enable bit to “0”. •After setting ③ below, clear the timer Y interrupt request bit to “0” and next set the timer Y interrupt enable bit to “1” (interrupt enabled). •Set ④ last.

Corrections and Supplementary Explanation for “3822 Group User’s Manuals” No.3

Rev.	Page	Contents
B	P2-71 (2) Timer Y ■ Period measurement mode Fig. 2.3.28	Previous change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before setting ① below, clear the interrupt enable bits (timer Y or CNTR1) and the interrupt request bits (timer Y or CNTR1) to “0”. •After setting ⑤ below, set the interrupt enable bits (timer Y or CNTR1) to “1” (interrupts enabled). After change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before timer Y stops counting (before setting ② below), clear the interrupt enable bit (timer Y or CNTR1) to “0”. •After setting ④ below, clear the interrupt request bit (timer Y or CNTR1) to “0” and next set the interrupt enable bit (timer Y or CNTR1) to “1” (interrupt enabled). •Set ⑤ last.
B	P2-72 (2) Timer Y ■ Event counter mode Fig. 2.3.29	Previous change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before setting ① below, clear the interrupt enable bits (timer Y or CNTR1) and the interrupt request bits (timer Y or CNTR1) to “0”. •After setting ⑤ below, set the interrupt enable bits (timer Y or CNTR1) to “1” (interrupts enabled). After change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before timer Y stops counting (before setting ② below), clear the interrupt enable bit (timer Y or CNTR1) to “0”. •After setting ④ below, clear the interrupt request bit (timer Y or CNTR1) to “0” and next set the interrupt enable bit (timer Y or CNTR1) to “1” (interrupt enabled). •Set ⑤ last.
B	P2-73 (2) Timer Y ■ Pulse width HL continuously measurement mode Fig. 2.3.30	Previous change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before setting ① below, clear the interrupt enable bits (timer Y or CNTR1) and the interrupt request bits (timer Y or CNTR1) to “0”. •After setting ⑤ below, set the interrupt enable bits (timer Y or CNTR1) to “1” (interrupts enabled). After change *[Notes on use]* **Notes 1:** For using interrupt processing, set the following: •Before timer Y stops counting (before setting ② below), clear the interrupt enable bit (timer Y or CNTR1) to “0”. •After setting ④ below, clear the interrupt request bit (timer Y or CNTR1) to “0” and next set the interrupt enable bit (timer Y or CNTR1) to “1” (interrupt enabled). •Set ⑤ last.

Corrections and Supplementary Explanation for “3822 Group User’s Manuals” No.4

Rev.	Page	Contents
A	P2-148 line 18	Error ■ Interrupt source selection bit ... **Note:** ... Correct (addition) **Note 2:** When an external trigger is selected, an ADT/A-D conversion interrupt may occur by switching the interrupt source selection bit from “1” to “0” or “0” to “1”. Before accepting an interrupt, set the interrupt request bit to “0” after disabling interrupts and setting the interrupt source selection bit.