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

RENESAS
Renesas Technology Corp.

MITSUBISHI 16-BIT SINGLE-CHIP MICROCOMPUTER

7700 FAMILY / 7900 SERIES

7905

Group

User's Manual

http://www.infomicom.maec.co.jp/indexe.htm

Before using this material, please visit the above website to confirm that this is the most current document available.

RENESAS
Renesas Technology Corp.

Rev. 1.0
Revision date: Nov. 28, 2001

Keep safety first in your circuit designs!

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

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
  The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
  Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (http://www.mitsubishichips.com).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

REVISION HISTORY	7905 GROUP USER’S MANUAL

Rev.	Date	Description	
		Page	Summary
1.0	11/28/01	—	First Edition

Preface

This manual describes the hardware of the Mitsubishi CMOS 16-bit microcomputers 7905 Group. After reading this manual, the user will be able to understand the functions, so that they can utilize their capabilities fully.

For details of software, refer to the "7900 Series Software Manual."
For details of development support tools, refer to the "Mitsubishi Microcomputer Development Support Tools" Homepage (http://www.tool-spt.maec.co.jp/index_e.htm).

BEFORE USING THIS MANUAL

1. Constitution

This user's manual consists of the following chapters. Refer to the chapters relevant to the products and processor mode.

In this manual, "M37905" means all of or one of the 7905 Group products, unless otherwise noted. Each chapter, except for Chapter 19, describes functions of the 7905 Group product at MD0 and MD1 = Vss level.

- **Chapter 1. DESCRIPTION to Chapter 17. DEBUG FUNCTION**
  Functions which are common to all products is described.

- **Chapter 18. APPLICATIONS**
  Example of application are described.

- **Chapter 19. FLASH MEMORY VERSION**
  Characteristics information for the flash memory version is described.

- **Appendix**
  Practical information for using the 7906 Group is described.

2. Remark

- **Product expansion**
  Refer to the latest datasheets or catalogs.

- **Electrical characteristics**
  Refer to the latest datasheets.

- **Software**
  Refer to the **"7900 Series Software Manual."**

- **Development support tools**
  Refer to the latest datasheets or catalogs.

Please Visit Our Web Site.

- Mitsubishi MCU Technical Information (http://www.infomicom.maec.co.jp/indexe.htm)
- Mitsubishi Microcomputer Development Support Tools (http://www.tool-spt.maec.co.jp/index_e.htm)

3. Signal levels in Figure

As a rule, signal levels in each operation example and timing diagram are as follows.

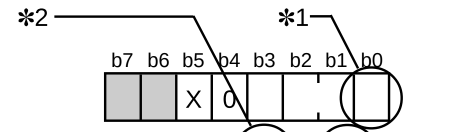
- Signal levels
  The upper line indicates "1," and the lower line indicates "0."
- Input/Output levels of pin
  The upper line indicates "H," and the lower line indicates "L."

Foe the exception, the level is shown on the left side of a signal.

4. Register structure

The view of the register structure is described below:

XXX register (address XX_{16})



Bit	Bit name	Function	At reset	R/W
0	• • • select bit	0 : … 1 : … The value is “0” at reading.	Undefined	WO
1	• • • select bit	b2 b1 0 0 : … 0 1 : … 1 0 : … 1 1 : …	0	RW
2			0	RW
3	• • • flag	0 : … 1 : …	0	RO
4	Fix this bit to “0.”		0	RW
5	This bit is invalid in … mode.		0	RW
6	Nothing is assigned.		Undefined	—
7	The value is “0” at reading.		0	—

✻6 ✻4

✻1

Blank : Set to “0” or “1” according to the usage.
0 : Set to “0” at writing.
1 : Set to “1” at writing.
× : Invalid depending on the mode or state. It may be “0” or “1.”
(shaded box) : Nothing is assigned.

✻2

0 : “0” immediately after reset.
1 : “1” immediately after reset.
Undefined : Undefined immediately after reset.

✻3

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid. Accordingly, the written value may be “0” or “1.”
WO : The written value becomes valid. It is impossible to read the bit state. The value is undefined at reading. However, when [“0” at reading”] is indicated in the “Function” or “Note” column, the bit is always “0” at reading. (See] 5 above.)
— : It is impossible to read the bit state. The value is undefined at reading. However, when [“0” at reading”] is indicated in the “Function” or “Note” column, the bit is always “0” at reading. (See] 6 above.) The written value becomes invalid. Accordingly, the written value may be “0” or “1.”

✻4 Invalid for that function or mode.

Table of contents

CHAPTER 5. INPUT/OUTPUT PINS

CHAPTER 6. INTERRUPTS

CHAPTER 7. TIMER A

CHAPTER 8. TIMER B

CHAPTER 9. PULSE OUTPUT PORT MODE

CHAPTER 10. THERR-PHASE WAVEFORM MODE

CHAPTER 11. SERIAL I/O

CHAPTER 12. A-D CONVERTER

CHAPTER 13. D-A CONVERTER

CHAPTER 14. WATCHDOG TIMER

CHAPTER 15. STOP AND WAIT MODES

CHAPTER 16. POWER SAVING FUNCTIONS

CHAPTER 17. DEBUG FUNCTION

CHAPTER 18. APPLICATIONS

CHAPTER 19. FLASH MEMORY VERSION

APPENDIX

Blank Page

CHAPTER 1
DESCRIPTION

1.1 Performance overview

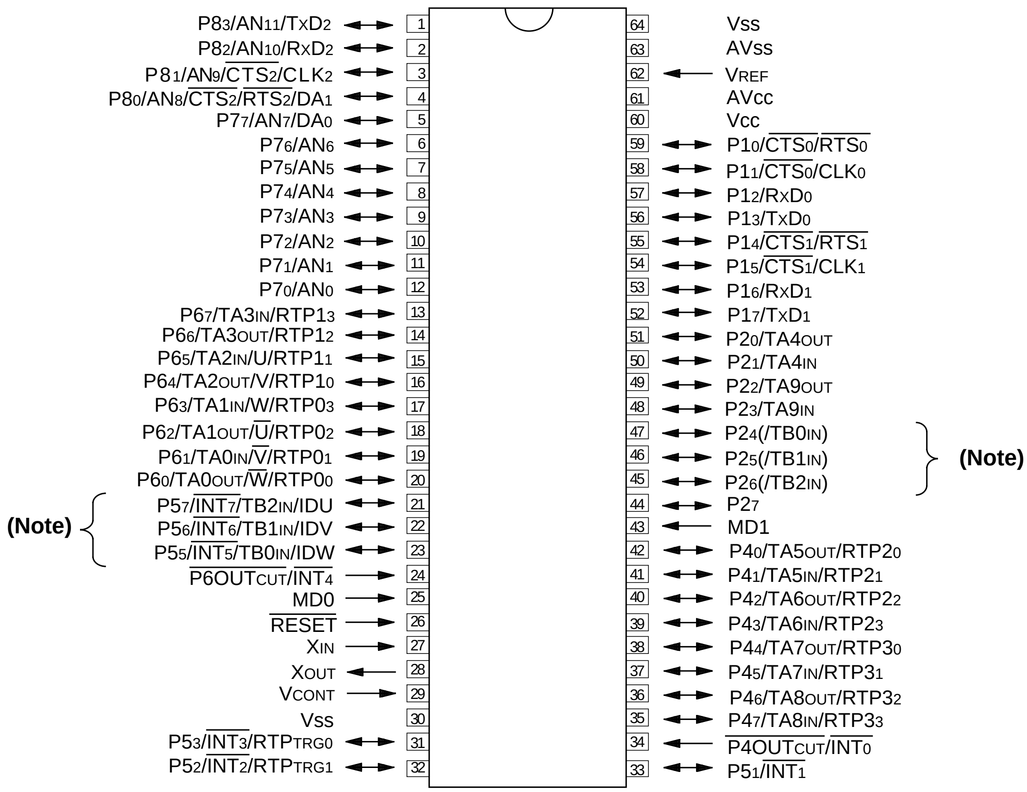
Table 1.1.1 lists the performance overview of the M37905M4C-XXXFP/SP.

Table 1.1.1 M37905M4C-XXXFP/SP performance overview

Items		Performance
Number of basic instructions		203
Instruction execution time		50 ns (the minimum instruction at $f(f_{sys})$ = 20 MHz)
External clock input frequency $f(X_{IN})$		20 MHz (maximum)
System clock frequency $f(f_{sys})$		20 MHz (maximum)
Memory sizes	ROM	32 Kbyte
	RAM	1024 bytes
Programmable Input/Output ports	P1, P2, P4, P6, P7	8 bits × 5
	P5	6 bits × 1
	P8	4 bits × 1
Multifunctional timer	TA0–TA9	16 bits × 10
	TB0–TB2	16 bits × 3
Serial I/O	UART0, UART1, UART2	(UART or clock synchronous serial I/O) × 3
A-D converter		10-bit successive approximation method × 1 (12 channels)
D-A converter		8 bits × 2
Watchdog timer		12 bits × 1
Interrupt	Maskable	8 external, 20 internal (Any of priority levels 0 through 7 can be set for each interrupt, by software.)
	Non-maskable	3 internal
Clock generating circuit		Built-in (externally connected to a ceramic resonator or a quartz-crystal oscillator)
PLL frequency multiplier		Double, Triple, or Quadruple
Power source voltage		5 V ± 0.5 V
Power dissipation		125 mW (at $f(f_{sys})$ = 20 MHz
Port Input/Output characteristics	Input/Output withstand voltage	5 V
	Output current	5 mA
Memory expansion		Not available. (Single-chip mode only)
Operating ambient temperature range		–20 °C to 85 °C
Device structure		CMOS high-performance silicon gate process
Package	M37905M4C-XXXFP	64-pin plastic molded QFP (64P6N-A)
	M37905M4C-XXXSP	64-pin shrink plastic molded SDIP (64P4B)

1.2 Pin configuration

Figure 1.2.1 shows the M37905M4C-XXXSP pin configuration, and Figure 1.2.2 shows the M37905M4C-XXXFP pin configuration.



Note: Allocation of pins TB0IN to TB2IN can be switched by software.

Outline 64P4B

Fig. 1.2.1 M37905M4C-XXXSP pin configuration (outline 64P4B, top view)

1.2 Pin configuration

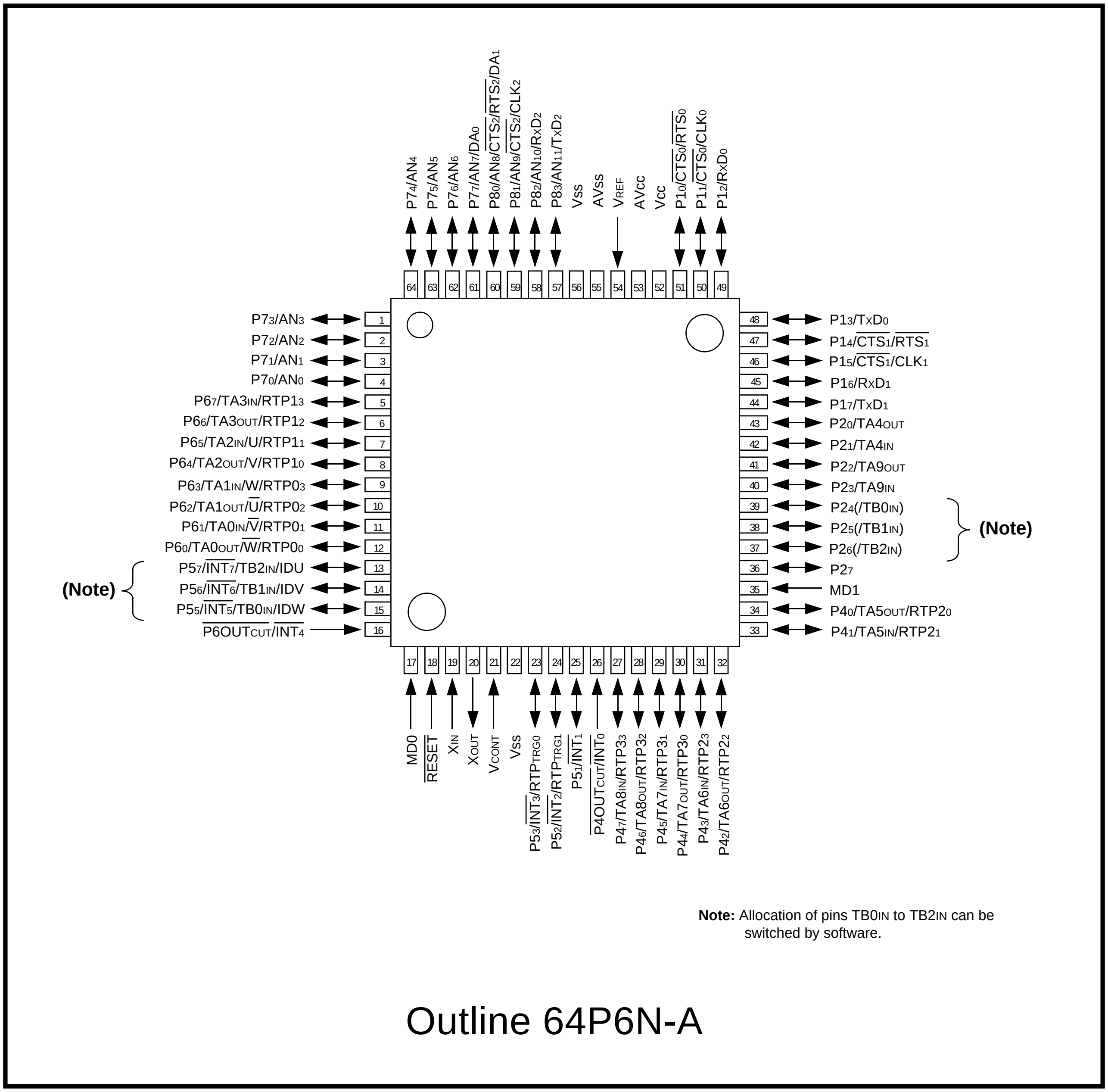


Fig. 1.2.2 M37905M4C-XXXFP pin configuration (outline 64P6N-A, top view)

1.3 Pin description

Tables 1.3.1 and 1.3.2 list the pin description.

Table 1.3.1 Pin description (1)

Pin	Name	Input/Output	Function
Vcc, Vss	Power source input	—	Apply 5 V ± 0.5 V to pin Vcc and 0 V to pin Vss.
MD0	MD0	Input	This pin switches the operating mode. This is only for the single-chip mode, so connect this pin to V_{SS}.
MD1	MD1		
$\overline{RESET}$	Reset input	Input	The microcomputer is reset when “L” level is input to this pin.
X_{IN}	Clock input	Input	Pins X_{IN} and X_{OUT} are the input and output pins of the clock generating circuit, respectively. Connect these pins via a ceramic resonator or a quartz-crystal oscillator. When an external clock is input, this clock should be input to pin X_{IN}, and pin X_{OUT} should be left open.
X_{OUT}	Clock output	Output	
V_{CONT}	Filter circuit connection	—	To use the PLL frequency multiplier, be sure to connect this pin to the filter circuit.
AVcc	Analog power source input	—	The power source input pin for the A-D converter. Connect this pin to Vcc.
AVss			The power source input pin for the A-D and D-A converters. Connect this pin to Vss.
V_{REF}	Reference voltage input	Input	This is the reference voltage input pin for the A-D and D-A converters.
$P1_0$–$P1_7$	I/O port P1	I/O	P0 is an 8-bit CMOS I/O port and has an I/O direction register. Each pin can function as an input or output port pin. By software, these pins can function as I/O pins for serial I/O.
$P2_0$–$P2_7$	I/O port P2	I/O	P2 is an 8-bit I/O port with the same function as port P1. By software, these pins can function as I/O pins for timers A4 and A9. Also, these pins can function as input pins for timers B0 to B2.
$P4_0$–$P4_7$	I/O port P4	I/O	P4 is an 8-bit I/O port with the same function as port P1. By software, these pins can function as I/O pins for timers A5 to A8, or as motor drive waveform output pins.
$P5_1$–$P5_3$, $P5_5$–$P5_7$	I/O port P5	I/O	P5 is a 6-bit I/O port with the same function as port P1. By software, these pins can function as input pins for timers B0 to B2, input pins for external interrupts, position data input pins in the three-phrase waveform mode, or trigger input pins in the pulse output port mode.
$P6_0$–$P6_7$	I/O port P6	I/O	P6 is an 8-bit I/O port with the same function as port P1. By software, these pins can function as I/O pins for timers A0 to A3, or as motor drive waveform output pins.

1.3 Pin description

Table 1.3.2 Pin description (2)

Pin	Name	Input/Output	Function
$P7_0$–$P7_7$	I/O port P7	I/O	P7 is an 8-bit I/O port with the same function as port P1. By software, these pins can function as input pins for the A-D converter, or output pins for the D-A converter.
$P8_0$–$P8_3$	I/O port P8	I/O	P8 is a 4-bit I/O port with the same function as port P1. By software, these pins can function as input pins for the A-D converter, output pins for the D-A converter, or as I/O pins for serial I/O.
$\overline{P4OUT_{CUT}}$	$\overline{P4OUT_{CUT}}$ input	Input	This pin has the function to forcibly place port P4 pins in the input mode (port-output-cutoff function). Also, this pin functions as an input pin for $\overline{INT_0}$, and as an input pin for the port-output-cutoff function in the motor drive waveform output mode.
$\overline{P6OUT_{CUT}}$	$\overline{P6OUT_{CUT}}$ input	Input	This pin has the function to forcibly place port P6 pins in the input mode (port-output-cutoff function). Also, this pin functions as an input pin for $\overline{INT_4}$, and as an input pin for the port-output-cutoff function in the motor drive waveform output mode.

1.4 Block diagram

Figure 1.4.1 shows the M37905 block diagram.

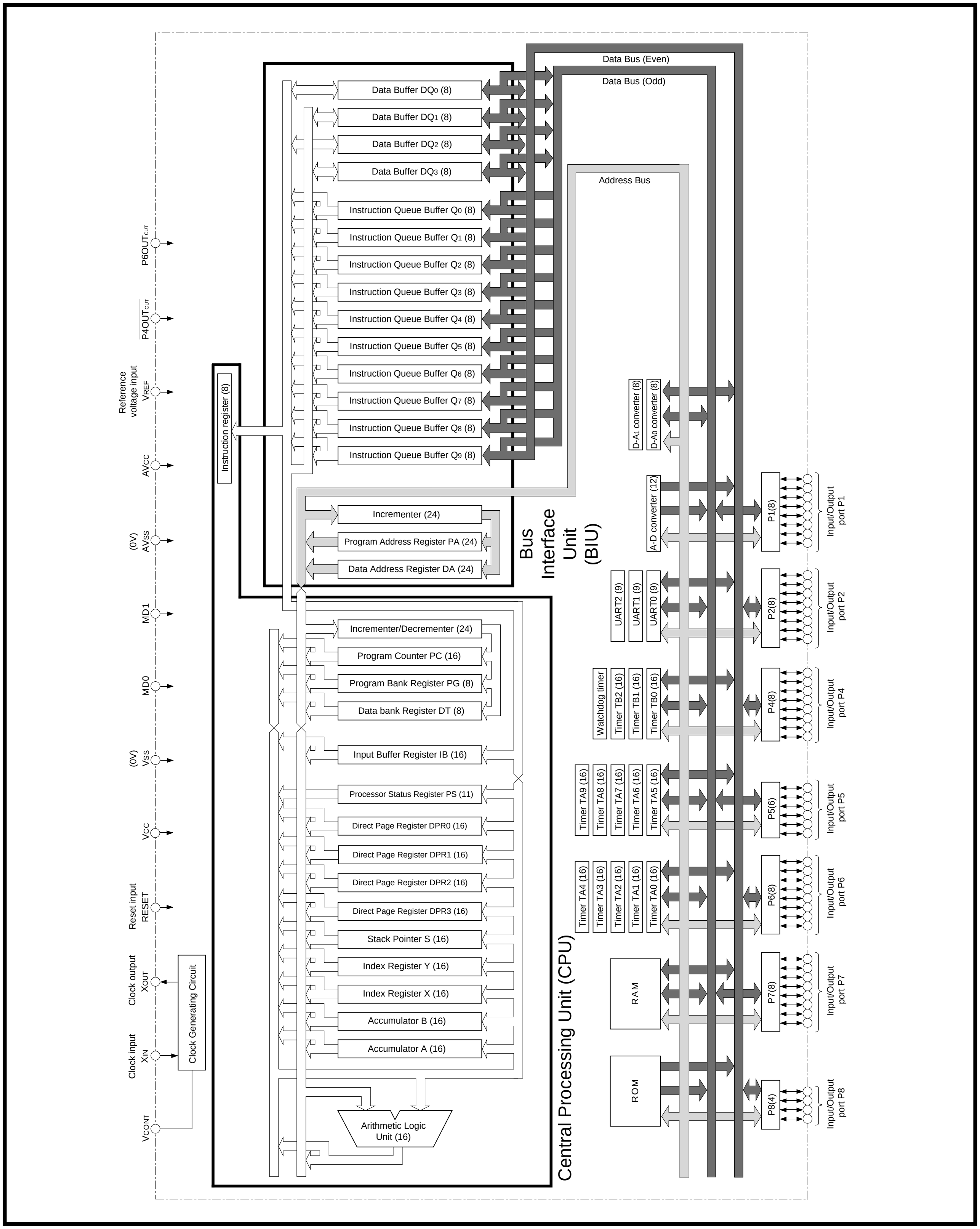


Fig. 1.4.1 M37905 block diagram

MEMORANDUM

CHAPTER 2

CENTRAL PROCESSING UNIT (CPU)

2.1 Central processing unit (CPU)

The CPU (Central Processing Unit) has 13 registers shown in Figure 2.1.1.

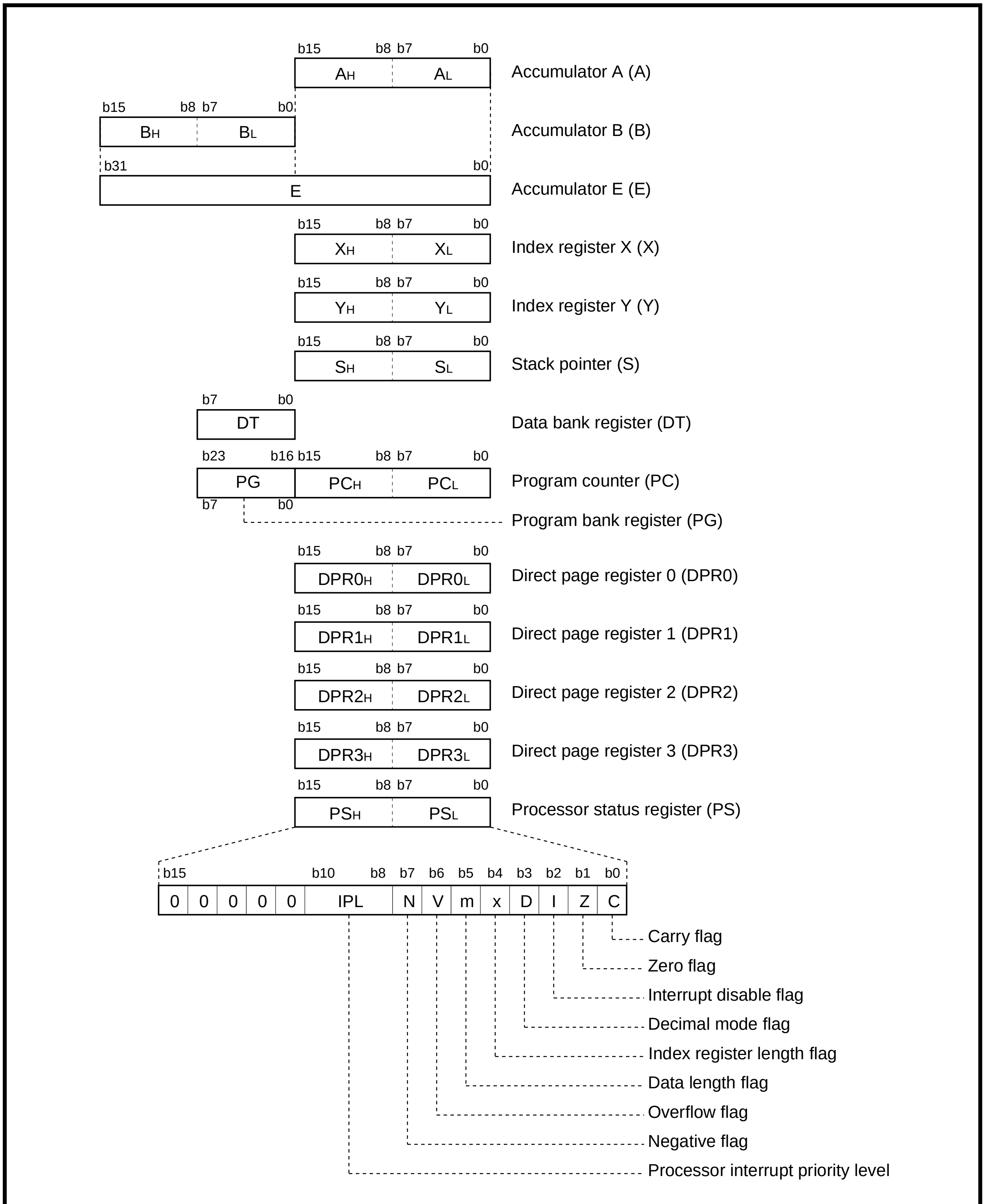


Fig. 2.1.1 CPU registers

2.1.1 Accumulator (Acc)

Accumulators A and B are available. Also, accumulators A and B can be connected in series in order to be used as a 32-bit accumulator (accumulator E).

(1) Accumulator A (A)

Accumulator A is the main register of the microcomputer. The transaction of data such as calculation, data transfer, and input/output are performed mainly through accumulator A. It consists of 16 bits, and the low-order 8 bits can also be used separately. The data length flag (m) determines whether the register is used as a 16-bit register or as an 8-bit register. Flag m is a part of the processor status register, which is described later. When an 8-bit register is selected, only the low-order 8 bits of accumulator A are used, and the contents of the high-order 8 bits is unchanged.

(2) Accumulator B (B)

Accumulator B is a 16-bit register with the same function as accumulator A. Accumulator B can be used instead of accumulator A. The use of accumulator B, however except for some instructions, requires more instruction bytes and execution cycles than those of accumulator A. Accumulator B is also affected by flag m just as in accumulator A.

(3) Accumulator E (E)

This 32-bit accumulator consists of accumulator A located in the low-order 16 bits and accumulator B located in the high-order 16 bits. This accumulator is used by an instruction that handles 32-bit data. It is not affected by flag m.

2.1.2 Index register X (X)

Index register X consists of 16 bits and the low-order 8 bits can also be used separately. The index register length flag (x) determines whether the register is used as a 16-bit register or as an 8-bit register. Flag x is a part of the processor status register, which is described later. When an 8-bit register is selected, only the low-order 8 bits of index register X are used, and the contents of the high-order 8 bits are not unchanged.

In an addressing mode in which index register X is used as an index register, the address obtained by adding the contents of this register to the operand's contents is accessed.

Also, each of the **MVP**, **MVN** and **RMPA** instructions uses index register X.

❋ Refer to **"7900 Series Software Manual"** for addressing modes and instructions.

2.1.3 Index register Y (Y)

Index register Y is a 16-bit register with the same function as index register X. Just as in index register X, this register is affected by flag X.

2.1.4 Stack pointer (S)

The stack pointer (S) is a 16-bit register. It is used for a subroutine call or an interrupt. It is also used when addressing modes using the stack are executed. The contents of S indicate an address (stack area) for storing registers during subroutine calls and interrupts. Bank 0_{16} is specified for the stack area. (Refer to section **"2.3 Access space."**)

When an interrupt request is accepted, the microcomputer stores the contents of the program bank register (PG) at the address indicated by the contents of S and decrements the contents of S by 1. Then the contents of the program counter (PC) and the processor status register (PS) are stored. The contents of S after accepting an interrupt request is equal to the contents of S decremented by 5 before accepting of the interrupt request. (See Figure 2.1.2.)

When completing the process in the interrupt routine and returning to the original routine, the contents of registers stored in the stack area are restored into the original registers in the reverse sequence (PS→PC→PG) by executing the **RTI** instruction. The contents of S is returned to the state before accepting an interrupt request.

The same operation is performed during a subroutine call, however, the contents of PS is not automatically stored. (The contents of PG may not be stored. This depends on the addressing mode.)

During interrupts or subroutine calls, the other registers are not automatically stored. Therefore, if the contents of these registers need to be held on, be sure to store them by software.

Additionally, the S's contents become "$0FFF_{16}$" at reset. The stack area changes when subroutines are nested or when multiple interrupt requests are accepted. Therefore, make sure of the subroutine's nesting depth not to destroy the necessary data.

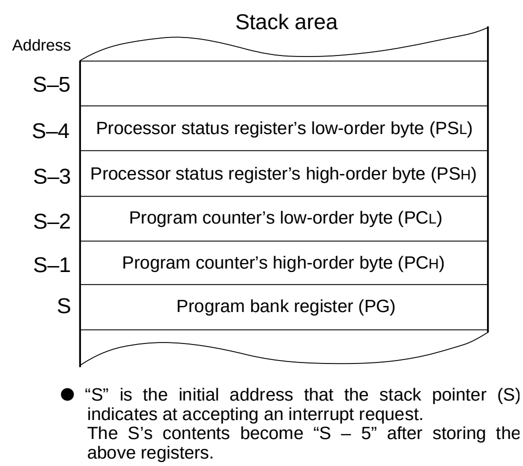


Fig. 2.1.2 Contents of stack area after accepting interrupt request

❋ Refer to "**7900 Series Software Manual**" for addressing modes and instructions.

2.1.5 Program counter (PC)

The program counter is a 16-bit counter that indicates the low-order 16 bits of the address (24 bits) at which an instruction to be executed next (in other words, an instruction to be read out from an instruction queue buffer next) is stored. The contents of the high-order program counter (PC_H) become "FF_{16}," and the low-order program counter (PC_L) becomes "FE_{16}" at reset. The contents of the program counter becomes the contents of the reset's vector address (addresses $FFFE_{16}$, $FFFF_{16}$) just after reset.
Figure 2.1.3 shows the program counter and the program bank register.

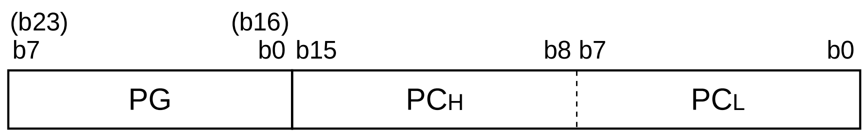


Fig. 2.1.3 Program counter and Program bank register

2.1.6 Program bank register (PG)

The memory space is divided into units of 64 Kbytes. This unit is called "bank." (Refer to section **"2.3 Access space."**)
The program bank register is an 8-bit register that indicates the high-order 8 bits of the address (24 bits) at which an instruction to be executed next (in other words, an instruction to be read out from an instruction queue buffer next) is stored. These 8 bits indicate a bank.
When a carry occurs after adding the contents of the program counter or adding the offset value to the contents of the program counter in the branch instruction and others, the contents of the program bank register is automatically incremented by 1. When a borrow occurs after subtracting the contents of the program counter, the contents of the program bank register is automatically decremented by 1. Therefore, there is no need to consider bank boundaries during programming, usually.
This register is cleared to "00_{16}" at reset.

2.1.7 Data bank register (DT)

The data bank register is an 8-bit register. In the following addressing modes using the data bank register, the contents of this register is used as the high-order 8 bits (bank) of a 24-bit address to be accessed.

Use the **LDT** instruction when setting a value to this register.
This register is cleared to "00_{16}" at reset.

● Addressing modes using data bank register
- •Direct indirect
- •Direct indexed X indirect
- •Direct indirect indexed Y
- •Absolute
- •Absolute indexed X
- •Absolute indexed Y
- •Absolute bit relative
- •Stack pointer relative indirect indexed Y
- •Multiplied accumulation

❋ Refer to **"7900 Series Software Manual"** for addressing modes.

2.1.8 Direct page register 0 to 3 (DPR0 to DPR3)

Each of direct page registers 0 to 3 (hereafter called the "DPRi") is a 16-bit register. The contents of this register specify the direct page area in bank 0_{16} or in the space across banks 0_{16} and 1_{16}. The following addressing modes use DPRi.

The contents of the DPRi indicate the base address (the lowest address) of the direct page area. The direct page area is specified in the space above this address.

After reset, whether to use DPR0 only or DPR0 to DPR3 can be selected by the direct page register switch bit. (See Figure 2.1.5). This selection specifies the direct page area. Table 2.1.1 lists the selection of the direct page register. Figure 2.1.4 shows setting examples of the direct page area.

At reset, DPR0 = "0000_{16}," and each of DPR1 to DPR3 becomes undefined.

● Addressing modes using direct page register
- Direct
- Direct indexed X
- Direct indexed Y
- Direct indirect
- Direct indexed X indirect
- Direct indirect indexed Y
- Direct indirect long
- Direct indirect long indexed Y
- Direct bit relative

Table 2.1.1 Selection of direct page register

	Direct page register switch bit	
	0	1
Usable DPRi	DPR0	DPR0 to DPR3
Direct page area	256 bytes	64 bytes at each DPRi

❋ Refer to **"7900 Series Software Manual"** for addressing modes and instructions.

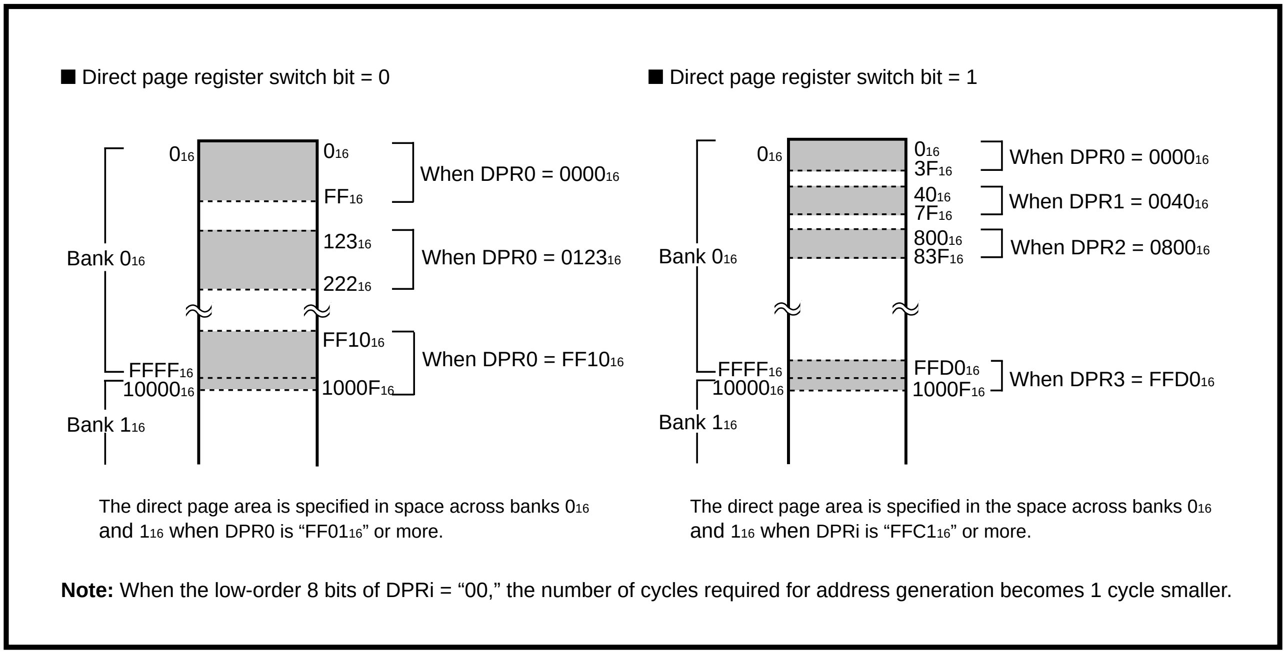


Fig. 2.1.4 Setting examples of direct page area

Processor mode register 1 (Address $5F_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	0		X

Bit	Bit name	Function	At reset	R/W
0	This bit may be either “0” or “1.”		1	RW
1	Direct page register switch bit	0 : Only DPR0 is used. 1 : DPR0 through DPR3 are used.	0	RW **(Note 1)**
6 to 2	Fix these bits to “00000.”		0	RW
7	Internal ROM bus cycle select bit **(Note 2)**	0 : 3ϕ 1 : 2ϕ	0	RW

X : It may be either “0” or “1.”

Notes 1: After reset, this bit is allowed to be changed only once. (During the software execution, be sure not to change this bit’s content.)

2: To reprogram the internal flash memory by using the CPU reprogramming mode, clear this bit to “0.” (Refer to section **“19.2 Flash memory CPU reprogramming mode.”**)

Fig. 2.1.5 Structure of processor mode register 1

2.1.9 Processor status register (PS)

PS is an 11-bit register.

Figure 2.1.6 shows the structure of PS. Refer to “**7900 Series Software Manual**” for detale about the change of each bit.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
0	0	0	0	0	IPL			N	V	m	x	D	I	Z	C	Processor status register (PS)

Note: Be sure to fix bits 15 through 11 to “0.”

Fig. 2.1.6 Structure of PS

(1) Bit 0: Carry flag (C)

This flag retains a carry or a borrow generated in the arithmetic and logic unit (ALU) during an arithmetic operation. This flag is also affected by shift and rotate instructions.

Be sure to use the **SEC** or **SEP** instruction to set this flag to “1”; and be sure to use the **CLC** or **CLP** instruction to clear it to “0”.

The contents of this flag is undefined at reset.

(2) Bit 1: Zero flag (Z)

This flag is set to “1” when the result of an arithmetic operation or data transfer is “0,” and cleared to “0” when otherwise. This flag is invalid in the decimal arithmetic operation.

Be sure to use the **SEP** instruction to set this flag to “1”; and be sure to use the **CLP** instruction to clear it to “0.”

The contents of this flag is undefined at reset.

(3) Bit 2: Interrupt disable flag (I)

This flag disables all maskable interrupts except the following: the address matching detection, watchdog timer, and 0 division interrupts. Interrupts are disabled when this flag is “1.” When an interrupt request has been accepted, this flag is automatically set to “1,” and multiple interrupts become disabled. Be sure to use the **SEI** or **SEP** instruction to set this flag to “1”; and be sure to use the **CLI** or **CLP** instruction to clear this flag to “0.”

This flag is set to “1” at reset.

(4) Bit 3: Decimal mode flag (D)

This flag determines whether addition and subtraction are performed in binary or decimal. Binary arithmetic operation is performed when this flag is “0.” When it is “1,” decimal arithmetic operation is performed with each 8 bits treated as 2-digit decimal (at m = 1) or each 16 bits treated as 4-digit decimal (at m = 0). Decimal adjust is automatically performed. Decimal operation is possible only with the **ADC, ADCB, SBC** and **SBCB** instructions. Be sure to use the **SEP** instruction to set this flag to “1”; and be sure to use the **CLP** instruction to clear it to “0.”

This flag is cleared to “0” at reset.

(5) Bit 4: Index register length flag (x)

This flag determines whether each of index register X and index register Y is used as a 16-bit register or an 8-bit register. That register is used as a 16-bit register when this flag is “0,” and as an 8-bit register when it is “1” **(Note)**. Be sure to use the **SEP** instruction to set this flag to “1”; and be sure to use the **CLP** instruction to clear it to “0.”

This flag is cleared to “0” at reset.

(6) Bit 5: Data length flag (m)

This flag determines whether to use data as a 16-bit unit or as an 8-bit unit. Each data is treated as a 16-bit unit when this flag is "0," and as an 8-bit unit when it is "1" **(Note)**.
Be sure to use the **SEM** or **SEP** instruction to set this flag to "1," and be sure to use the **CLM** or **CLP** instruction to clear it to "0."
This flag is cleared to "0" at reset.

Note: When transferring data between registers which are different in bit length, this data is transferred with the length of the transfer destination register, except for the case where the **TXA**, **TYA**, **TXB**, **TYB**, and **TXS** instructions used. Refer to "**7900 series software manual**" for detail.

(7) Bit 6: Overflow flag (V)

This flag is used when addition or subtraction is performed with a word regarded as signed binary. The overflow flag is set to "1" when the result of addition or subtraction exceeds the range between –2147483648 and +2147483647 (when 32-bit length operation), the range between –32768 and +32767 (when 16-bit length operation), or the range between –128 and +127 (when 8-bit length operation).
The overflow flag is also set to "1" when the operation result of the **DIV** or **DIVS** instruction exceeds the length of the register which will store that result. This flag is invalid in the decimal mode. Be sure to use the **SEP** instruction to set this flag to "1," and be sure to use the **CLV** or **CLP** instruction to clear it to "0."
The contents of this flag is undefined at reset.

(8) Bit 7: Negative flag (N)

This flag is set to "1" when the result of arithmetic operation or data transfer is negative. (The most significant bit of the result is "1.") It is cleared to "0" in all other cases. This flag is invalid in the decimal mode. Be sure to use the **SEP** instruction to set this flag to "1," and be sure to use the **CLP** instruction to clear it to "0."
The contents of this flag is undefined at reset.

(9) Bits 10 to 8: Processor interrupt priority level (IPL)

These 3 bits can determine the processor interrupt priority level to one of levels 0 through 7. When the interrupt priority level of a requested interrupt, which has been set in the corresponding interrupt control register, is higher than IPL, that interrupt becomes enabled. When an interrupt request is accepted, IPL is stored in the stack area, and IPL is replaced by the interrupt priority level of the accepted interrupt request.
There are no instruction to directly set or clear the bits of IPL. IPL can be changed by storing the new IPL into the stack area and updating PS with the **PUL** or **PLP** instruction.
The contents of IPL is cleared to "000_2" at reset.

2.2 Bus interface unit (BIU)

The bus interface unit (hereafter called "BIU") performs the following two operations:

- Instruction prefetch
- Data transfer (read and write)

Figure 2.2.1 shows the bus and BIU.
BIU is structured with four kinds of registers shown in Figure 2.2.2. Table 2.2.1 lists the function of the BIU registers.

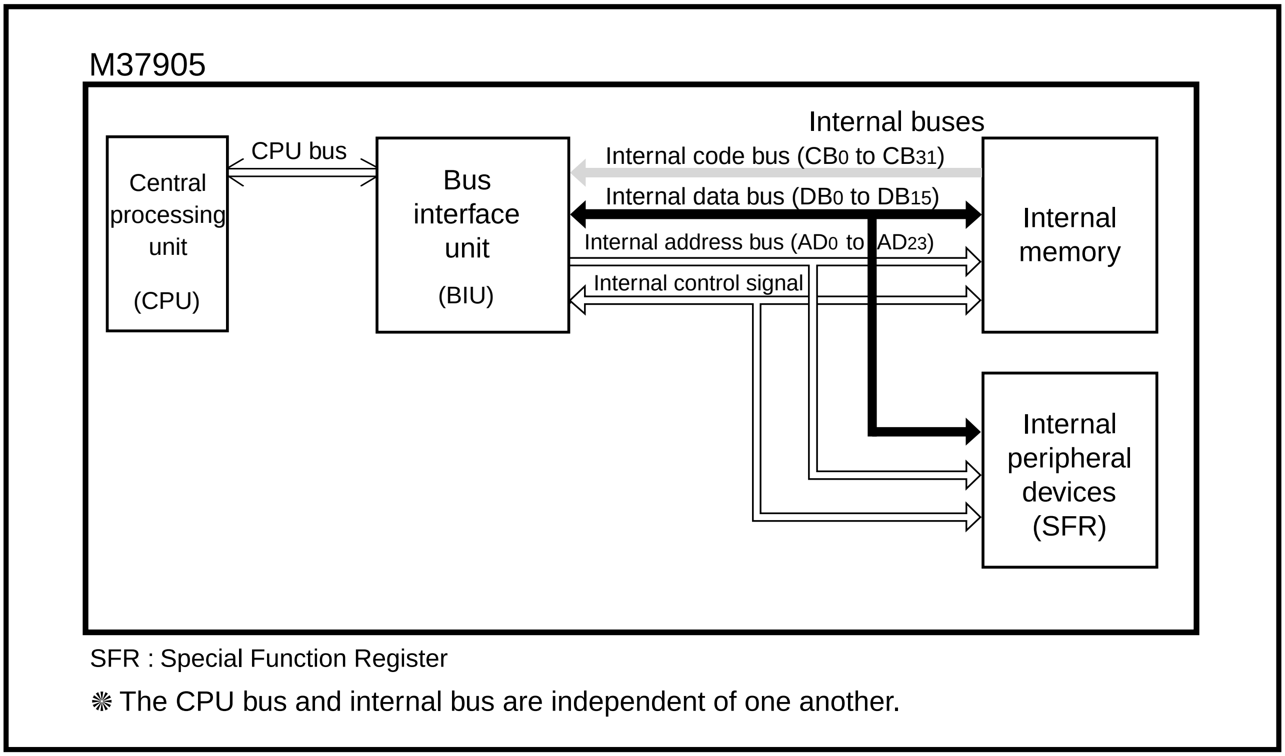


Fig. 2.2.1 Bus and BIU

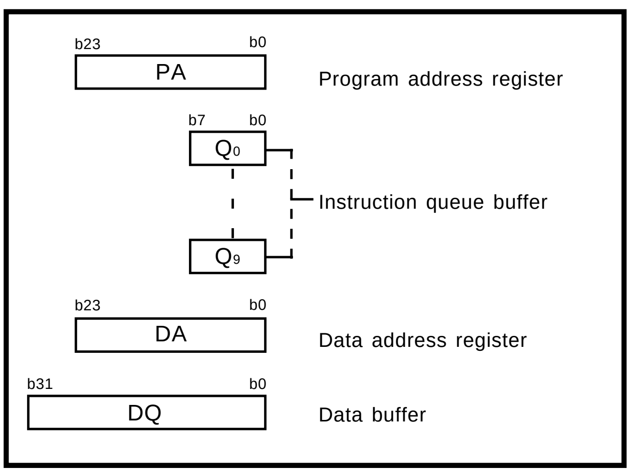


Fig. 2.2.2 BIU registers' structure

Table 2.2.1 Functions of BIU registers

Name	Functions
Program address register	Indicates a storage address of the instruction to be fetched into an instruction queue buffer, next.
Instruction queue buffer	Temporarily stores an instruction which has been fetched.
Data address register	Indicates an address from which data will be read or to which data will be written, next.
Data buffer	Temporarily stores data which has been read from memory•I/O device by BIU or which will be written to memory•I/O device by the CPU.

In the M37905, the internal buses are used when the CPU accesses the internal area (the internal memory and SFR).

2.2.1 Instruction prefetch

While the CPU does not use the internal buses, the BIU reads instructions from the memory and then stores them in the instruction queue buffer. The CPU reads instructions from the instruction queue buffer and executes them, so that the CPU can operate at high speed without access to the memory, which requires a long access time.

The instruction queue buffer can store instructions up to 10 bytes. The contents of the instruction queue buffer is initialized when a branch is made, and the BIU reads a new instruction from the branch destination address.

When instructions in the instruction queue buffer are insufficient for the CPU's needs, the BIU extends the low-level duration of ϕ_{CPU} (See Figure 4.2.1.) in order to keep the CPU waiting until the BIU fetches instructions of the required byte number or more.

Figure 2.2.3 shows operating waveform examples at instruction prefetch. Note that the operation of BIU's instruction prefetch also varies with the store addresses of instructions. Table 2.2.2 lists the store address of prefetched instructions.

When the instruction prefetch from internal memory, the instructions are fetched from 4-byte boundaries, 4 bytes at a time. (See Figure 2.2.3.)

Also, at branch, regardless of the low-order 2 bits' **contents (AD_1 and AD_0) of the branch destination address, 4 bytes are fetched at time from the 4-byte boundaries. (See Figure 2.2.3.) In this case,** out of the data (instructions) which will be output onto the internal code buses, 4 bytes at a time, the instructions assigned at the branch destination address and the following addresses will be fetched into the instruction queue buffer. Accordingly, as listed in Table 2.2.3, the number of bytes to be fetched into the instruction queue buffer varies according to the branch destination address.

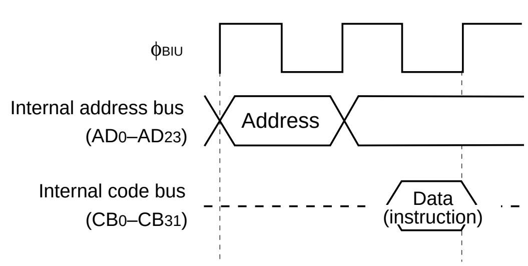
Table 2.2.2 Store address of prefetched instruction

	Low-order 3 bits at store address		
	AD_2	AD_1	AD_0
Even-numbered address	×	×	0
4-byte boundaries	×	0	0
8-byte boundaries	0	0	0

X: It may be either "0" or "1."

Table 2.2.3 Number of bytes to be fetched into instruction queue buffer

Low-order 2 bits of branch destination address		Low-order 2 bits of address to be output onto address bus		Number of bytes to be fetched into instruction queue buffer
AD_1	AD_0	AD_1	AD_0	
0	0	0	0	4
0	1	0	0	3
1	0	0	0	2
1	1	0	0	1



ϕ_{BIU}: Operation clock of BIU (Refer to **"CHAPTER 4. CLOCK GENERATING CIRCUIT."**)

Note: This waveform applies when bus cycle = 2φ. For details of the bus cycle at access to the internal area, see Table 2.2.4.

Fig. 2.2.3 Operation waveform examples at instruction prefetch

2.2.2 Data Transfer (read and write)

When the CPU reads or writes data from or to the internal area, it requests the BIU to read or write data. The BIU outputs control signals in order to control the internal address and data buses in response to the request from the CPU. The cycle where the following are performed is referred to "bus cycle":

- The BIU controls buses.
- Data transfer is performed between the internal area and BIU.

Table 2.2.4 lists the bus cycles at access to the internal area. Figure 2.2.4 shows operating waveform examples at reading from or writing to the internal area.

(1) Reading data

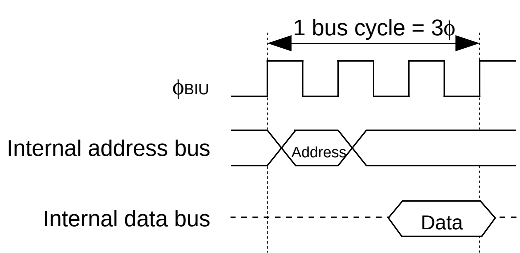
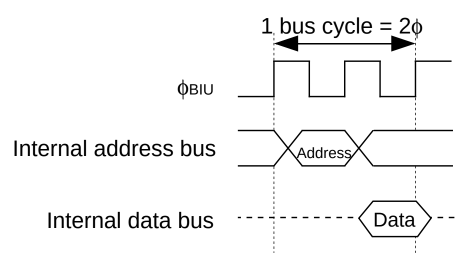
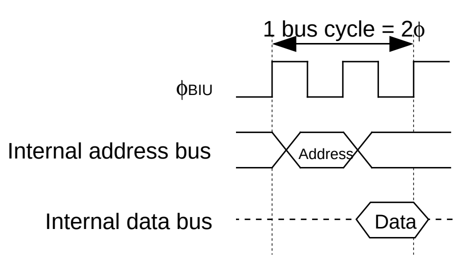
The CPU informs the BIU's data address register of the address where the data to be read is stored, so the CPU requests the data. In this case, the CPU waits until the requested data is ready in the BIU.

The BIU outputs the address informed by the CPU onto the internal address bus. Then, the CPU reads the contents of the informed address and takes them into the data buffer. The CPU continues processing using data in the data buffer.

(2) Writing data

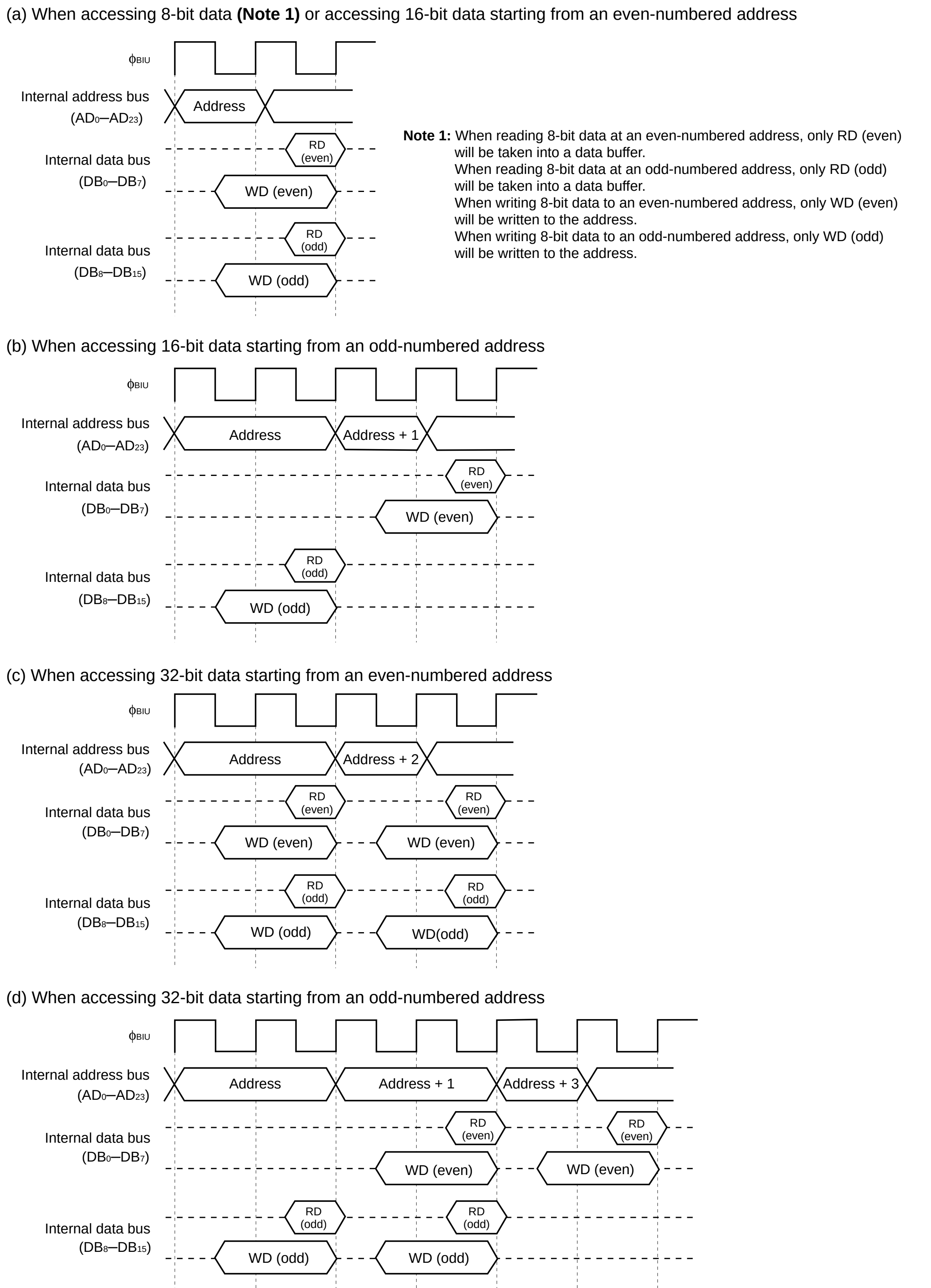
The CPU informs the BIU's data address register of the address to which the data will be written, so the CPU writes the data into the data buffer. The BIU outputs the address informed by the CPU onto the internal address bus. Then, the BIU writes the data in the data buffer into the informed address.

Table 2.2.4 Bus cycles at access to internal area

<table>
<tr><th></th><th>Bus cycle = 3φ (Note)
(Internal ROM bus cycle select bit = 0)</th><th>Bus cycle = 2φ (Note)
(Internal ROM bus cycle select bit = 1)</th></tr>
<tr><td>ROM</td><td>
</td><td>
</td></tr>
<tr><td>RAM
SFR</td><td colspan="2">
</td></tr>
</table>

Internal ROM bus cycle select bit: Bit 7 at address $5F_{16}$

Note: We usually recommend to select "bus cycle = 2 φ." When reprogramming the internal flash memory in the CPU reprogramming mode, be sure to select bus cycle = 3φ. (Refer to section **"19.2 Flash memory CPU reprogramming mode."**)



RD: Data to be read, WD: Data to be written

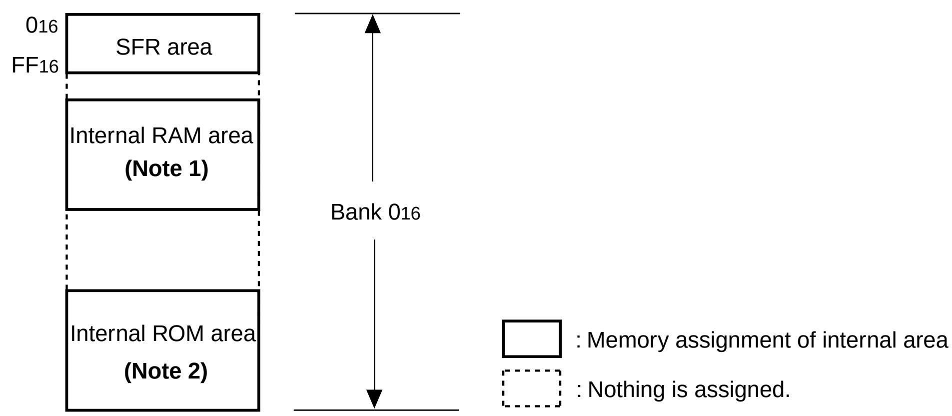
Note 2: The above waveforms apply when bus cycle = 2φ.
For the bus cycles at access to the internal area, see Table 2.2.4.

Fig. 2.2.4 Operation waveform examples at reading from or writing to internal area

2.3 Access space

The access space of the M37905 is assigned to a 16-Mbyte space from addresses 0_{16} to $FFFFFF_{16}$. (See Figure 2.3.1.) Note that only the internal memory can be accessed because the M37905 operates only in the single-chip mode.

The memory and I/O devices are assigned in the same access space. Accordingly, it is possible to perform transfer and arithmetic operations using the same instructions, without discrimination of the memory from I/O devices.



Notes 1: When the internal RAM area is followed by an unused area, do not assign a program to the last 8 bytes of the internal RAM area.
2: Do not assign a program to the last 8 bytes of the internal ROM area.
3: The memory assignment of the internal area varies according to the product type. Refer to section **"Appendix 11. Memory assignment of 7905 Group,"** or the latest datasheets, catalogs.
4: Refer to section **"19.1.1 Memory assignment,"** about the flash memory version.

SFR : Special Function Register

Fig. 2.3.1 M37905's access space

2.4 Memory assignment

This section describes the memory assignment in the internal area.

2.4.1 Memory assignment in internal area

SFR (Special Function Register), internal RAM, and internal ROM are assigned to the internal area. Figure 2.4.1 shows the memory assignment in the internal area.

(1) SFR area

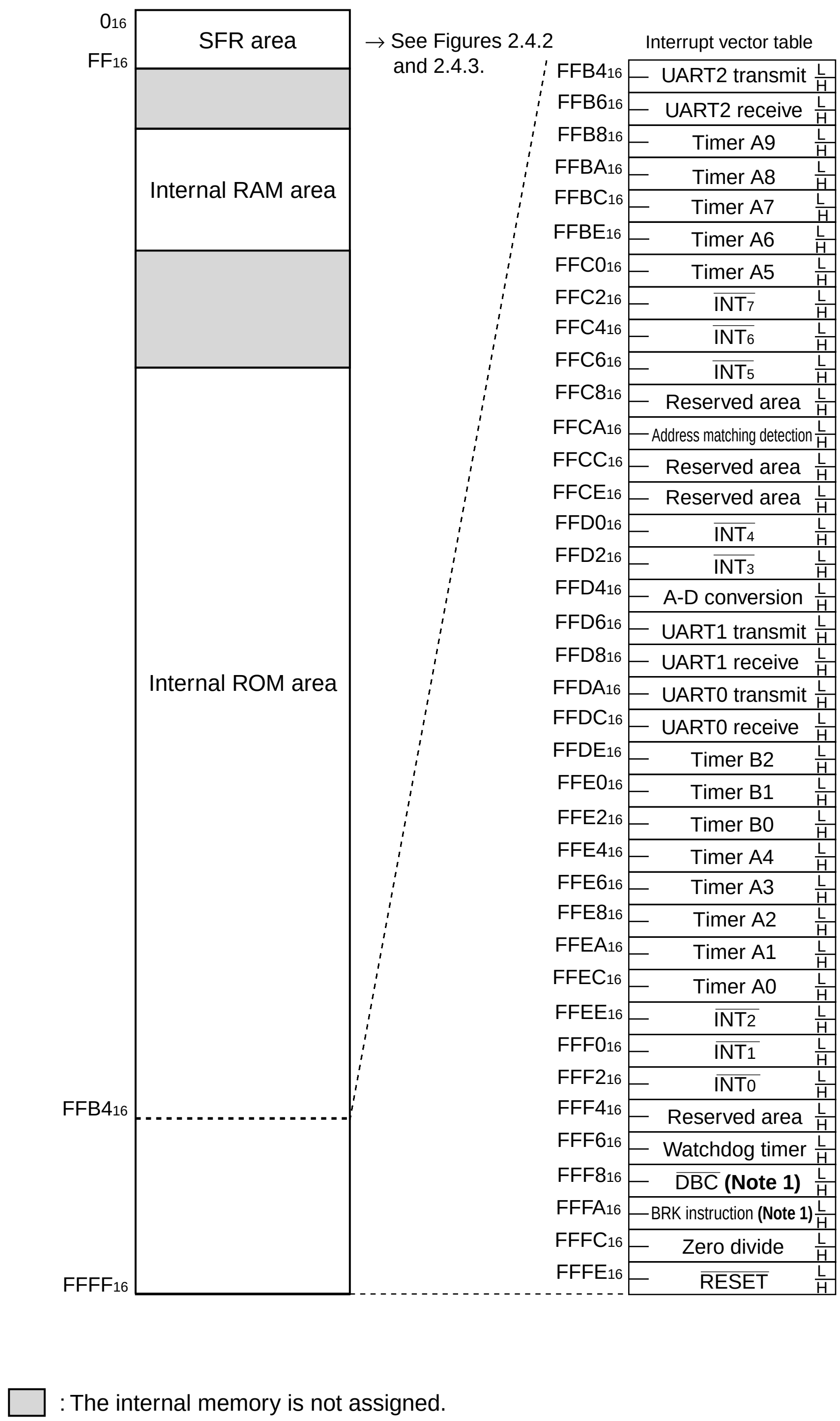
The registers used to set the internal peripheral devices are assigned to addresses 0_{16} to FF_{16}. This area is called SFR. Figures 2.4.2 and 2.4.3 show the SFR area's memory assignment.
For each register in the SFR area, refer to each functional description in this manual.
For the state of the SFR area immediately after reset, refer to section **"3.3 State of internal area."**

(2) Internal RAM area

The internal RAM area is used as a stack area, as well as an area to store data. Accordingly, be sure to set the nesting depth of a subroutine and multiple interrupts' level not to destroy the necessary data.
When the internal RAM area is followed by an unused area, do not assign a program to the last 8 bytes of the internal RAM area. (Data is allowed to be assigned there. Also, when the internal RAM area is followed by the internal ROM area succeedingly, a program is allowed to be assigned there.)

(3) Internal ROM area

Addresses $FFB4_{16}$ to $FFFF_{16}$ are the vector addresses for reset and interrupts. (This is called the interrupt vector table.)
Do not assign a program to the last 8 bytes of the internal ROM area. (Data is allowed to be assigned there.)



Notes 1: These are interrupts only for debugging; do not use these interrupts.

2: Memory assignment of the internal area varies according to the product type. Refer to section **"Appendix 11. Memory assignment of 7905 Group"** or the latest datasheets, catalogues.

Fig. 2.4.1 Memory assignment in internal area

Address	
0_{16}	**(Note 1)**
1_{16}	**(Note 1)**
2_{16}	**(Note 2)**
3_{16}	Port P1 register
4_{16}	**(Note 2)**
5_{16}	Port P1 direction register
6_{16}	Port P2 register
7_{16}	**(Note 2)**
8_{16}	Port P2 direction register
9_{16}	**(Note 2)**
A_{16}	Port P4 register
B_{16}	Port P5 register
C_{16}	Port P4 direction register
D_{16}	Port P5 direction register
E_{16}	Port P6 register
F_{16}	Port P7 register
10_{16}	Port P6 direction register
11_{16}	Port P7 direction register
12_{16}	Port P8 register
13_{16}	
14_{16}	Port P8 direction register
15_{16}	
16_{16}	**(Note 2)**
17_{16}	**(Note 2)**
18_{16}	**(Note 2)**
19_{16}	**(Note 2)**
$1A_{16}$	
$1B_{16}$	
$1C_{16}$	
$1D_{16}$	
$1E_{16}$	A-D control register 0
$1F_{16}$	A-D control register 1
20_{16}	A-D register 0
21_{16}	
22_{16}	A-D register 1
23_{16}	
24_{16}	A-D register 2
25_{16}	
26_{16}	A-D register 3
27_{16}	
28_{16}	A-D register 4
29_{16}	
$2A_{16}$	A-D register 5
$2B_{16}$	
$2C_{16}$	A-D register 6
$2D_{16}$	
$2E_{16}$	A-D register 7
$2F_{16}$	
30_{16}	UART0 transmit/receive mode register
31_{16}	UART0 baud rate register (BRG0)
32_{16}	UART0 transmit buffer register
33_{16}	
34_{16}	UART0 transmit/receive control register 0
35_{16}	UART0 transmit/receive control register 1
36_{16}	UART0 receive buffer register
37_{16}	
38_{16}	UART1 transmit/receive mode register
39_{16}	UART1 baud rate register (BRG1)
$3A_{16}$	UART1 transmit buffer register
$3B_{16}$	
$3C_{16}$	UART1 transmit/receive control register 0
$3D_{16}$	UART1 transmit/receive control register 1
$3E_{16}$	UART1 receive buffer register
$3F_{16}$	

Address	
40_{16}	Count start flag 0
41_{16}	Count start flag 1
42_{16}	One-shot start flag 0
43_{16}	One-shot start flag 1
44_{16}	Up-down flag 0
45_{16}	Timer A clock division select register
46_{16}	Timer A0 register
47_{16}	
48_{16}	Timer A1 register
49_{16}	
$4A_{16}$	Timer A2 register
$4B_{16}$	
$4C_{16}$	Timer A3 register
$4D_{16}$	
$4E_{16}$	Timer A4 register
$4F_{16}$	
50_{16}	Timer B0 register
51_{16}	
52_{16}	Timer B1 register
53_{16}	
54_{16}	Timer B2 register
55_{16}	
56_{16}	Timer A0 mode register
57_{16}	Timer A1 mode register
58_{16}	Timer A2 mode register
59_{16}	Timer A3 mode register
$5A_{16}$	Timer A4 mode register
$5B_{16}$	Timer B0 mode register
$5C_{16}$	Timer B1 mode register
$5D_{16}$	Timer B2 mode register
$5E_{16}$	Processor mode register 0
$5F_{16}$	Processor mode register 1
60_{16}	Watchdog timer register
61_{16}	Watchdog timer frequency select register
62_{16}	Particular function select register 0
63_{16}	Particular function select register 1
64_{16}	Particular function select register 2
65_{16}	**(Note 2)**
66_{16}	Debug control register 0
67_{16}	Debug control register 1
68_{16}	Address compare register 0 **(Note 3)**
69_{16}	
$6A_{16}$	
$6B_{16}$	Address compare register 1 **(Note 3)**
$6C_{16}$	
$6D_{16}$	
$6E_{16}$	$\overline{INT_3}$ interrupt control register
$6F_{16}$	$\overline{INT_4}$ interrupt control register
70_{16}	A-D conversion interrupt control register
71_{16}	UART0 transmit interrupt control register
72_{16}	UART0 receive interrupt control register
73_{16}	UART1 transmit interrupt control register
74_{16}	UART1 receive interrupt control register
75_{16}	Timer A0 interrupt control register
76_{16}	Timer A1 interrupt control register
77_{16}	Timer A2 interrupt control register
78_{16}	Timer A3 interrupt control register
79_{16}	Timer A4 interrupt control register
$7A_{16}$	Timer B0 interrupt control register
$7B_{16}$	Timer B1 interrupt control register
$7C_{16}$	Timer B2 interrupt control register
$7D_{16}$	$\overline{INT_0}$ interrupt control register
$7E_{16}$	$\overline{INT_1}$ interrupt control register
$7F_{16}$	$\overline{INT_2}$ interrupt control register

Address	
80_{16}	**(Note 2)**
81_{16}	**(Note 2)**
82_{16}	**(Note 2)**
83_{16}	**(Note 2)**
84_{16}	**(Note 2)**
85_{16}	**(Note 2)**
86_{16}	**(Note 2)**
87_{16}	**(Note 2)**
88_{16}	
89_{16}	
$8A_{16}$	**(Note 2)**
$8B_{16}$	
$8C_{16}$	**(Note 2)**
$8D_{16}$	
$8E_{16}$	**(Note 2)**
$8F_{16}$	
90_{16}	**(Note 2)**
91_{16}	
92_{16}	**(Note 2)**
93_{16}	
94_{16}	
95_{16}	External interrupt input read register
96_{16}	D-A control register
97_{16}	
98_{16}	D-A register 0
99_{16}	D-A register 1
$9A_{16}$	
$9B_{16}$	
$9C_{16}$	**(Note 2)**
$9D_{16}$	**(Note 2)**
$9E_{16}$	Flash memory control register **(Note 4)**
$9F_{16}$	

Notes 1: Do not read from and write to this register.
2: Do not write to this register.
3: When these registers are accessed, set the address compare register access enable bit (bit 2 at address 67_{16}) to "1." (Refer to **"CHAPTER 17. DEBUG FUNCTION."**)
4: This register is assigned only to the flash memory version. (Refer to **"CHAPTER 19. FLASH MEMORY VERSION."**) Nothing is assigned here in the mask ROM version.

Fig. 2.4.2 SFR area's memory map (1)

Address	
A0$_{16}$	Pulse output control register
A1$_{16}$	
A2$_{16}$	Pulse output data register 0
A3$_{16}$	
A4$_{16}$	Pulse output data register 1
A5$_{16}$	
A6$_{16}$	Waveform output mode register
A7$_{16}$	Dead-time timer
A8$_{16}$	Three-phase output data register 0
A9$_{16}$	Three-phase output data register 1
AA$_{16}$	Position-data-retain function control register
AB$_{16}$	
AC$_{16}$	Serial I/O pin control register
AD$_{16}$	
AE$_{16}$	Port P2 pin function control register
AF$_{16}$	
B0$_{16}$	UART2 transmit/receive mode register
B1$_{16}$	UART2 baud rate register (BRG2)
B2$_{16}$	UART2 transmit buffer register
B3$_{16}$	
B4$_{16}$	UART2 transmit/receive control register 0
B5$_{16}$	UART2 transmit/receive control register 1
B6$_{16}$	UART2 receive buffer register
B7$_{16}$	
B8$_{16}$	**(Note 5)**
B9$_{16}$	
BA$_{16}$	**(Note 5)**
BB$_{16}$	**(Note 5)**
BC$_{16}$	Clock control register 0
BD$_{16}$	**(Note 5)**
BE$_{16}$	**(Note 5)**
BF$_{16}$	**(Note 5)**

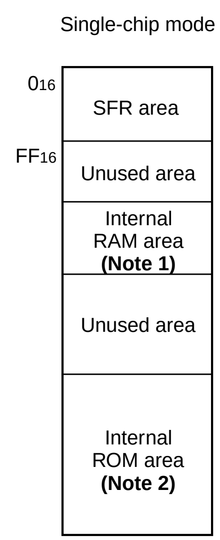
Address	
C0$_{16}$	
C1$_{16}$	
C2$_{16}$	
C3$_{16}$	
C4$_{16}$	Up-down flag 1
C5$_{16}$	
C6$_{16}$	Timer A5 register
C7$_{16}$	
C8$_{16}$	Timer A6 register
C9$_{16}$	
CA$_{16}$	Timer A7 register
CB$_{16}$	
CC$_{16}$	Timer A8 register
CD$_{16}$	
CE$_{16}$	Timer A9 register
CF$_{16}$	
D0$_{16}$	Timer A0$_1$ register
D1$_{16}$	
D2$_{16}$	Timer A1$_1$ register
D3$_{16}$	
D4$_{16}$	Timer A2$_1$ register
D5$_{16}$	
D6$_{16}$	Timer A5 mode register
D7$_{16}$	Timer A6 mode register
D8$_{16}$	Timer A7 mode register
D9$_{16}$	Timer A8 mode register
DA$_{16}$	Timer A9 mode register
DB$_{16}$	A-D control register 2
DC$_{16}$	Comparator function select register 0
DD$_{16}$	Comparator function select register 1
DE$_{16}$	Comparator result register 0
DF$_{16}$	Comparator result register 1
E0$_{16}$	A-D register 8
E1$_{16}$	
E2$_{16}$	A-D register 9
E3$_{16}$	
E4$_{16}$	A-D register 10
E5$_{16}$	
E6$_{16}$	A-D register 11
E7$_{16}$	
E8$_{16}$	**(Note 5)**
E9$_{16}$	**(Note 5)**
EA$_{16}$	**(Note 5)**
EB$_{16}$	**(Note 5)**
EC$_{16}$	**(Note 5)**
ED$_{16}$	**(Note 5)**
EE$_{16}$	**(Note 5)**
EF$_{16}$	**(Note 5)**
F0$_{16}$	
F1$_{16}$	UART2 transmit interrupt control register
F2$_{16}$	UART2 receive interrupt control register
F3$_{16}$	
F4$_{16}$	
F5$_{16}$	Timer A5 interrupt control register
F6$_{16}$	Timer A6 interrupt control register
F7$_{16}$	Timer A7 interrupt control register
F8$_{16}$	Timer A8 interrupt control register
F9$_{16}$	Timer A9 interrupt control register
FA$_{16}$	
FB$_{16}$	
FC$_{16}$	
FD$_{16}$	$\overline{INT_5}$ interrupt control register
FE$_{16}$	$\overline{INT_6}$ interrupt control register
FF$_{16}$	$\overline{INT_7}$ interrupt control register

Note 5: Do not write to this register.

Fig. 2.4.3 SFR area's memory map (2)

2.5 Processor modes

The M37905 operates only in the single-chip mode. Figure 2.5.1 shows the memory assignment of the M37905.



Notes 1: When the internal RAM area is followed by an unused area, do not assign a program to the last 8 bytes of the internal RAM area.
2: Do not assign a program to the last 8 bytes of the internal ROM area.
3: The memory assignment of the internal area varies according to the product type.
Refer to section **"Appendix 11. Memory assignment of 7905 Group,"** or the latest datasheets, catalogs.

Fig. 2.5.1 Memory assignment of M37905

2.5.1 Single-chip mode

In this mode, ports P1, P2, P4 to P8 serve as programmable I/O ports. (When an internal peripheral device is used, the corresponding port pin serves as the device's I/O pin).
In this mode, only the internal area (SFR, internal RAM, and internal ROM) can be accessed.

2.5.2 Setting of processor mode

The processor mode is set by the following:

- Voltage level applied to the MD0 and MD1 pins
- Processor mode bits (bits 1 and 0 at address $5E_{16}$)

The V_{SS}–level voltage must be applied to the MD0 and MD1 pins, because the M37905 operates only in the single-chip mode. Also, the processor mode bit must be "00."

Figure 2.5.2 shows the structure of the processor mode register 0 (address $5E_{16}$).

Processor mode register 0 (Address $5E_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
0				X	X	0	0

Bit	Bit name	Function	At reset	R/W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Do not select. 1 0 : Do not select. 1 1 : Do not select.	0	RW
1			0	RW
2	Any of these bits may be either "0" or "1."		0	RW
3			1	RW
4	Interrupt priority detection time select bits	b5 b4 0 0 : 7 cycles of f_{sys} 0 1 : 4 cycles of f_{sys} 1 0 : 2 cycles of f_{sys} 1 1 : Do not select.	0	RW
5			0	RW
6	Software reset bit	The microcomputer is reset by writing "1" to this bit. The value is "0" at reading.	0	WO
7	Fix this bit to "0."		0	RW

X : It may be either "0" or "1."

Fig. 2.5.3 Structure of processor mode register 0

[Precautions for setting of processor mode]

The M37905 operates only in the single-chip mode. Therefore, for the M37905, do as follows:

- The MD0 and MD1 pins must be connected to Vss.
- The processor mode bits (bits 0 and 1 at address $5E_{16}$) must be fixed to "00_2."

MEMORANDUM

CHAPTER 3
RESET

RESET

3.1 Reset operation

There are 3 ways to reset the microcomputer:

Hardware reset : Apply “L” level of voltage to pin $\overline{RESET}$ while the power source voltage (Vcc) meets the recommended operating conditions.

Software reset : Write “1” to the software reset bit (bit 6 at address $5E_{16}$) while the power source voltage (Vcc) meets the recommended operating conditions.

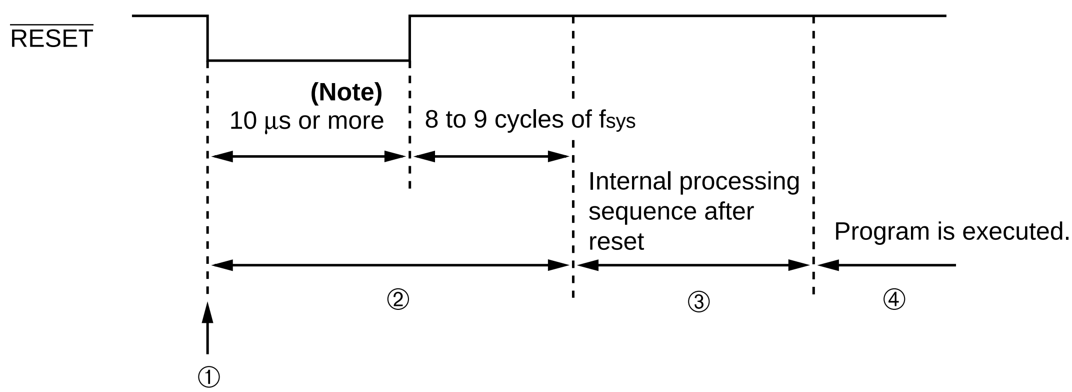
Power-on reset : After power-on, raise the voltage level at pin Vcc to the level, which meets the recommend operating conditions, with “L” level of voltage applied to pin $\overline{RESET}$.

3.1 Reset operation

Operations of hardware, software, and power-on reset are described below.

3.1.1 Hardware reset

Figure 3.1.1 shows an example of hardware reset timing.



Note: The above is applied when the oscillator is stably oscillating or when an external clock is stably input from pin X_{IN}. When the oscillator is not stably oscillating (including the case at the stop mode’s termination; refer to section **“15.3 Stop mode.”**), apply “L” level of voltage for 10 µs or more after the oscillation becomes stable.

Fig. 3.1.1 Example of hardware reset timing

The following explains how the microcomputer operates in the above periods, ① to ④.

① After applying “L” level of voltage to pin $\overline{RESET}$, the microcomputer initializes pins within a period of several ten cycles of f_{sys}. (Refer to section **“3.2 Pin state.”**)

② The microcomputer initializes the central processing unit (CPU) and SFR area in the following periods. (Refer to section **“3.3 State of internal area.”**)
- While pin $\overline{RESET}$ is at “L” level.
- In the period of 8 to 9 cycles of f_{sys} after pin $\overline{RESET}$ goes from “L” to “H.”

③ After ②, the microcomputer performs “Internal processing sequence after reset.” (Refer to section **“3.4 Internal processing sequence after reset.”**)

④ The microcomputer executes a program beginning with the address which has been set into the reset vector addresses (addresses $FFFE_{16}$ and $FFFF_{16}$).

3.1.2 Software reset

The microcomputer initializes pins, CPU, and SFR area just as in the case of hardware reset (Refer to sections **"3.2 Pin state"** and **"3.3 State of internal area."**) by writing "1" to the software reset bit. (See Figure 3.1.2.)

After initialization is completed, the microcomputer performs "Internal processing sequence after reset." (Refer to section **"3.4 Internal processing sequence after reset."**) After that, it executes a program beginning with the address which has been set into the reset vector addresses (addresses $FFFE_{16}$ and $FFFF_{16}$).

Processor mode register 0 (Address $5E_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
0				X	X	0	0

Bit	Bit name	Function	At reset	R/W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Do not select. 1 0 : Do not select. 1 1 : Do not select.	0	RW
1			0	RW
2	Any of these bits may be either "0" or "1."		0	RW
3			1	RW
4	Interrupt priority detection time select bits	b5 b4 0 0 : 7 cycles of f_{sys} 0 1 : 4 cycles of f_{sys} 1 0 : 2 cycles of f_{sys} 1 1 : Do not select.	0	RW
5			0	RW
6	Software reset bit	The microcomputer is reset by writing "1" to this bit. The value is "0" at reading.	0	WO
7	Fix this bit to "0."		0	RW

X: It may be either "0" or "1."

Fig. 3.1.2 Structure of processor mode register 0

3.1.3 Power-on reset

The following describes the operation of the microcomputer at power-on reset.

① After powered on, within the several ten cycles of f_{sys} after the voltage level at pin Vcc meets the recommended operating conditions with the voltage level at pin $\overline{\text{RESET}}$ = "L," the microcomputer initializes pins; refer to section **"3.2 Pin state."**

② After the voltage level at pin $\overline{\text{RESET}}$ goes from "L" to "H," the microcomputer initializes the CPU and SFR area within a period of 8 to 9 cycles of f_{sys}. (Contents of the internal RAM area become undefined; refer to section **"3.3 State of internal area."**)

③ After ②, the microcomputer performs "Internal processing sequence after reset."; refer to section **"3.4 Internal processing sequence after reset."**

④ The microcomputer executes a program beginning with the address which has been set into the reset vector addresses (addresses $FFFE_{16}$ and $FFFF_{16}$).

Figure 3.1.3 shows the power-on reset conditions. Figure 3.1.4 shows an example of a power-on reset circuit.

After the voltage level at pin Vcc meets the recommended operating conditions and the oscillator's operation is stabilized (See Figure 3.1.3.), apply "L" level of voltage to pin $\overline{\text{RESET}}$ for 10 µs or more. When an oscillator is used, the time required for stabilizing oscillation depends on the oscillator. For details, contact the oscillator manufacturer.

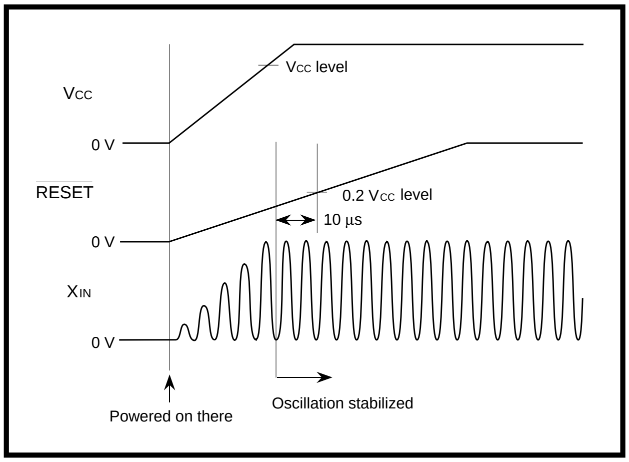


Fig. 3.1.3 Power-on reset conditions

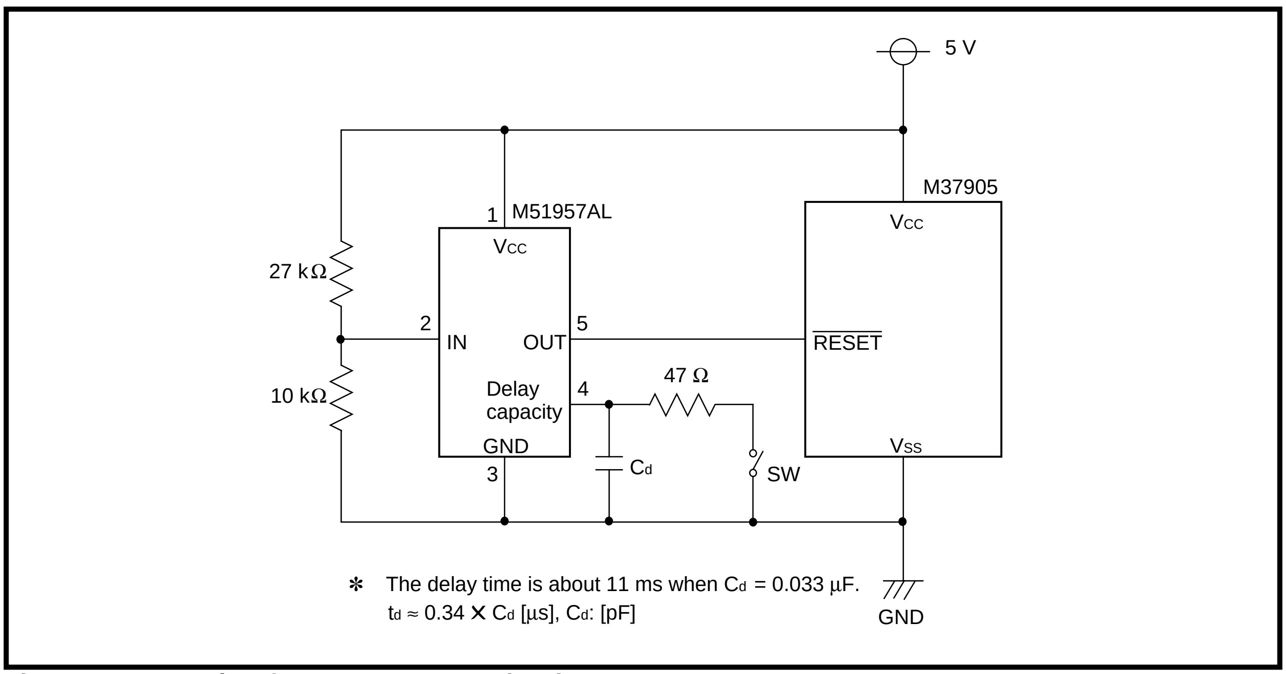


Fig. 3.1.4 Example of power-on reset circuit

3.2 Pin state

Table 3.2.1 lists the microcomputer's pin state while the voltage level at pin $\overline{\text{RESET}}$ is "L."

Table 3.2.1 Pin state while voltage level at pin $\overline{\text{RESET}}$ is "L"

	Pin MD1's level	Pin MD0's level	Pin (Bus, Port) name	Pin state
MASK ROM version, Flash memory version **(Note 1)**	Vss	Vss	P1, P2, P4–P8	Floating.
Flash memory version **(Note 1)**	Vcc	Vss	P1, P2, P4–P8	Floating.
		Vcc	P1, P2, P4–P8	Floating **(Note 2)**.

Notes 1: Refer to **"CHAPTER 19. FLASH MEMORY VERSION."**
2: Pins $P5_6$, $P5_7$ and $P6_0$ to $P6_5$ output "H" or "L" level when "H" level of voltage is applied to pin V_{CONT} and "L" level to pins $P7_0$, $P7_1$.

3.3 State of internal area

Figure 3.3.1 shows the state of CPU registers immediately after reset. Figures 3.3.2 to 3.3.9 show the state of the SFR and internal RAM areas immediately after reset.

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.
[0] (hatched) : "0" immediately after reset. Fix this bit to "0."

Register name	State immediately after reset: b15 to b8	State immediately after reset: b7 to b0
Accumulator A (A)	?	?
Accumulator B (B)	?	?
Index register X (X)	?	?
Index register Y (Y)	?	?
Stack pointer (S)	$0F_{16}$	FF_{16}
Data bank register (DT)		00_{16}
Program bank register (PG)		00_{16}
Program counter (PC)	Contents at address $FFFF_{16}$	Contents at address $FFFE_{16}$
Direct page register 0 (DPR0)	00_{16}	00_{16}
Direct page register i (DPRi) (i = 1 to 3)	?	?

Processor status register (PS):

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value	0 (fixed)	0 (fixed)	0 (fixed)	0 (fixed)	0 (fixed)	0	0	0	?	?	0	0	0	1	?	?
Flag						IPL	IPL	IPL	N	V	m	x	D	I	Z	C

Fig. 3.3.1 State of CPU registers immediately after reset

●SFR area (Addresses 0_{16} to FF_{16})

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid. It is impossible to read the bit state.
(shaded box) : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.

[0] : Always "0" at reading.
[1] : Always "1" at reading.
[?] : Always undefined at reading.
[0 hatched] : "0" immediately after reset. Fix this bit to "0."

(In the table below, [x] denotes a shaded bit cell and {0} a hatched bit cell.)

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
0_{16}		**(Note 1)**	?
1_{16}		**(Note 1)**	?
2_{16}		**(Note 2)**	?
3_{16}	Port P1 register	RW	?
4_{16}		**(Note 2)**	?
5_{16}	Port P1 direction register	RW	00_{16}
6_{16}	Port P2 register	RW	?
7_{16}		**(Note 2)**	?
8_{16}	Port P2 direction register	RW	00_{16}
9_{16}		**(Note 2)**	?
A_{16}	Port P4 register	RW	?
B_{16}	Port P5 register	RW (b7–b5), [ ] (b4), RW (b3–b1), [ ] (b0)	? ? ? [?] ? ? ? [?]
C_{16}	Port P4 direction register	RW	00_{16}
D_{16}	Port P5 direction register	RW (b7–b5), [ ] (b4), RW (b3–b1), [ ] (b0)	0 0 0 [?] 0 0 0 [?]
E_{16}	Port P6 register	RW	?
F_{16}	Port P7 register	RW	?
10_{16}	Port P6 direction register	RW	00_{16}
11_{16}	Port P7 direction register	RW	00_{16}
12_{16}	Port P8 register	[ ] (b7–b4), RW (b3–b0)	[?] [?] [?] [?] 0 0 0 0
13_{16}		[ ]	?
14_{16}	Port P8 direction register	[ ] (b7–b4), RW (b3–b0)	[?] [?] [?] [?] 0 0 0 0
15_{16}		[ ]	?
16_{16}		**(Note 2)**	?
17_{16}		**(Note 2)**	?
18_{16}		**(Note 2)**	?
19_{16}		**(Note 2)**	?
$1A_{16}$		[ ]	?
$1B_{16}$		[ ]	?
$1C_{16}$		[ ]	?
$1D_{16}$		[ ]	?
$1E_{16}$	A-D control register 0	RW	0 0 {0} 0 0 ? ? ?
$1F_{16}$	A-D control register 1	[ ] (b7), RW (b6–b0)	[0] 0 {0} 0 0 0 1 1

Notes 1: Do not read from and write to this register.
2: Do not write to this register.

Fig. 3.3.2 State of SFR and internal RAM areas immediately after reset (1)

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
20_{16}	A-D register 0	**(Note 3)**	?
21_{16}		**(Note 3)**	0 0 0 0 0 0 ?
22_{16}	A-D register 1	**(Note 3)**	?
23_{16}		**(Note 3)**	0 0 0 0 0 0 ?
24_{16}	A-D register 2	**(Note 3)**	?
25_{16}		**(Note 3)**	0 0 0 0 0 0 ?
26_{16}	A-D register 3	**(Note 3)**	?
27_{16}		**(Note 3)**	0 0 0 0 0 0 ?
28_{16}	A-D register 4	**(Note 3)**	?
29_{16}		**(Note 3)**	0 0 0 0 0 0 ?
$2A_{16}$	A-D register 5	**(Note 3)**	?
$2B_{16}$		**(Note 3)**	0 0 0 0 0 0 ?
$2C_{16}$	A-D register 6	**(Note 3)**	?
$2D_{16}$		**(Note 3)**	0 0 0 0 0 0 ?
$2E_{16}$	A-D register 7	**(Note 3)**	?
$2F_{16}$		**(Note 3)**	0 0 0 0 0 0 ?
30_{16}	UART0 transmit/receive mode register	RW	00_{16}
31_{16}	UART0 baud rate register (BRG0)	WO	?
32_{16}	UART0 transmit buffer register	WO	?
33_{16}		WO (b0)	?
34_{16}	UART0 transmit/receive control register 0	RW, RO, RW	0 0 0 0 1 0 0 0
35_{16}	UART0 transmit/receive control register 1	RO, RW, RO, RW	0 0 0 0 0 0 1 0
36_{16}	UART0 receive buffer register	RO	?
37_{16}		RO (b0)	0 0 0 0 0 0 0 ?
38_{16}	UART1 transmit/receive mode register	RW	00_{16}
39_{16}	UART1 baud rate register (BRG1)	WO	?
$3A_{16}$	UART1 transmit buffer register	WO	?
$3B_{16}$		WO (b0)	?
$3C_{16}$	UART1 transmit/receive control register 0	RW, RO, RW	0 0 0 0 1 0 0 0
$3D_{16}$	UART1 transmit/receive control register 1	RO, RW, RO, RW	0 0 0 0 0 0 1 0
$3E_{16}$	UART1 receive buffer register	RO	?
$3F_{16}$		RO (b0)	0 0 0 0 0 0 0 ?

Note 3: The access characteristics at addresses 20_{16} to $2F_{16}$ vary according to the contents of the comparator function select register 0 (address DC_{16}). (Refer to **"CHAPTER 12. A-D CONVERTER."**)

Fig. 3.3.3 State of SFR and internal RAM areas immediately after reset (2)

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
40_{16}	Count start register 0	RW	00_{16}
41_{16}	Count start register 1	(b7–b5 shaded) RW	? 0 0 0 0 0
42_{16}	One-shot start register 0	RW (shaded) WO	0 ? 0 0 0 0 0
43_{16}	One-shot start register 1	RW (shaded) WO	0 ? 0 0 0 0 0
44_{16}	Up-down register 0	WO RW	0 0 0 0 0 0 0 0
45_{16}	Timer A clock division select register	(b7–b2 shaded) RW RW	0 0 0 0 0 0 0 0
46_{16}	Timer A0 register	**(Note 4)**	?
47_{16}		**(Note 4)**	?
48_{16}	Timer A1 register	**(Note 4)**	?
49_{16}		**(Note 4)**	?
$4A_{16}$	Timer A2 register	**(Note 4)**	?
$4B_{16}$		**(Note 4)**	?
$4C_{16}$	Timer A3 register	**(Note 4)**	?
$4D_{16}$		**(Note 4)**	?
$4E_{16}$	Timer A4 register	**(Note 4)**	?
$4F_{16}$		**(Note 4)**	?
50_{16}	Timer B0 register	**(Note 5)**	?
51_{16}		**(Note 5)**	?
52_{16}	Timer B1 register	**(Note 5)**	?
53_{16}		**(Note 5)**	?
54_{16}	Timer B2 register	**(Note 5)**	?
55_{16}		**(Note 5)**	?
56_{16}	Timer A0 mode register	RW	00_{16}
57_{16}	Timer A1 mode register	RW	00_{16}
58_{16}	Timer A2 mode register	RW	00_{16}
59_{16}	Timer A3 mode register	RW	00_{16}
$5A_{16}$	Timer A4 mode register	RW	00_{16}
$5B_{16}$	Timer B0 mode register	RW (Note 6) RW	0 0 ? 0 0 0 0 0
$5C_{16}$	Timer B1 mode register	RW (Note 6) RW	0 0 ? 0 0 0 0 0
$5D_{16}$	Timer B2 mode register	RW (Note 6) RW	0 0 ? 0 0 0 0 0
$5E_{16}$	Processor mode register 0	RW WO RW	0 0 0 0 1 0 0 0
$5F_{16}$	Processor mode register 1	RW	0 0 0 0 0 0 0 1

Notes 4: The access characteristics at addresses 46_{16} to $4F_{16}$ vary according to the timer A's operating mode. (Refer to **"CHAPTER 7. TIMER A."**)

5: The access characteristics at addresses 50_{16} to 55_{16} vary according to the timer B's operating mode. (Refer to **"CHAPTER 8. TIMER B."**)

6: The access characteristics for bit 5 at addresses $5B_{16}$ to $5D_{16}$ vary according to the timer B's operating mode. (Refer to **"CHAPTER 8. TIMER B."**)

Fig. 3.3.4 State of SFR and internal RAM areas immediately after reset (3)

Address	Register name	Access b7							Access b0	State b7							State b0
6016	Watchdog timer register	(Note 7)								? (Note 8)							
6116	Watchdog timer frequency select register	RW	RW						RW	0	0	?					0
6216	Particular function select register 0	RW						RW (Note 9)		0	0	0	0	0	0	0	0
6316	Particular function select register 1		RW		RW	RW	RW	(Note 10)		0	0	0	0	0	0	(Note 11)	
6416	Particular function select register 2									?							
6516		(Note 12)								?							
6616	Debug control register 0		RW							1	0	(Note 11)	0	0	(Note 11)		
6716	Debug control register 1		RO	RO		RW	RW	RO	RW	0	0	0	?	0	0	0	0
6816	Address comparison register 0	RW (Note 13)								?							
6916		RW (Note 13)								?							
6A16		RW (Note 13)								?							
6B16	Address comparison register 1	RW (Note 13)								?							
6C16		RW (Note 13)								?							
6D16		RW (Note 13)								?							
6E16	$\overline{INT_3}$ interrupt control register			RW						?		0	0	0	0	0	0
6F16	$\overline{INT_4}$ interrupt control register			RW						?		0	0	0	0	0	0
7016	A-D conversion interrupt control register					RW				?				?	0	0	0
7116	UART0 transmit interrupt control register					RW				?				0	0	0	0
7216	UART0 receive interrupt control register					RW				?				0	0	0	0
7316	UART1 transmit interrupt control register					RW				?				0	0	0	0
7416	UART1 receive interrupt control register					RW				?				0	0	0	0
7516	Timer A0 interrupt control register					RW				?				0	0	0	0
7616	Timer A1 interrupt control register					RW				?				0	0	0	0
7716	Timer A2 interrupt control register					RW				?				0	0	0	0
7816	Timer A3 interrupt control register					RW				?				0	0	0	0
7916	Timer A4 interrupt control register					RW				?				0	0	0	0
7A16	Timer B0 interrupt control register					RW				?				0	0	0	0
7B16	Timer B1 interrupt control register					RW				?				0	0	0	0
7C16	Timer B2 interrupt control register					RW				?				0	0	0	0
7D16	$\overline{INT_0}$ interrupt control register			RW						?		0	0	0	0	0	0
7E16	$\overline{INT_1}$ interrupt control register			RW						?		0	0	0	0	0	0
7F16	$\overline{INT_2}$ interrupt control register			RW						?		0	0	0	0	0	0

Notes 7 : By writing dummy data to address 6016, a value of "FFF16" is set to the watchdog timer. The dummy data is not retained anywhere.

8 : A value of "FFF16" is set to the watchdog timer. (Refer to **"CHAPTER 14. WATCHDOG TIMER."**)

9 : After writing "5516" to address 6216, each bit must be set.

10 : It is possible to read the bit state at reading. By writing "0" to this bit, this bit becomes "0." But when writing "1" to this bit, this bit will not change.

11 : This bit becomes "0" at power-on reset. This bit retains the state immediately before reset in the case of hardware reset and software reset.

12 : Do not write to this register.

13 : When these registers are accessed, set the address comparison register access enable bit (bit 2 at address 6716) to "1." (Refer to **"CHAPTER 17. DEBUG FUNCTION."**)

Fig. 3.3.5 State of SFR and internal RAM areas immediately after reset (4)

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
80_{16}		(Note 14)	?
81_{16}		(Note 14)	?
82_{16}		(Note 14)	?
83_{16}		(Note 14)	?
84_{16}		(Note 14)	?
85_{16}		(Note 14)	?
86_{16}		(Note 14)	?
87_{16}		(Note 14)	?
88_{16}			?
89_{16}			?
$8A_{16}$		(Note 14)	?
$8B_{16}$			?
$8C_{16}$		(Note 14)	?
$8D_{16}$			?
$8E_{16}$		(Note 14)	?
$8F_{16}$			?
90_{16}		(Note 14)	?
91_{16}			?
92_{16}		(Note 14)	?
93_{16}			?
94_{16}			?
95_{16}	External interrupt input read-out register	RO	?
96_{16}	D-A control register	b1: RW, b0: RW	b7–b2: ?, b1: 0, b0: 0
97_{16}			?
98_{16}	D-A register 0	RW	00_{16}
99_{16}	D-A register 1	RW	00_{16}
$9A_{16}$			?
$9B_{16}$			?
$9C_{16}$		(Note 14)	?
$9D_{16}$		(Note 14)	?
$9E_{16}$	Flash memory control register **(Note 15)**	b5: RW, b3: RW, b1: RW, b0: RO	0 0 0 0 0 0 0 1
$9F_{16}$			?

Notes 14 : Do not write to this register.
15 : This register is assigned only to the flash memory version. (Refer to **"CHAPTER 19. FLASH MEMORY VERSION."**) Nothing is assigned here in the mask ROM version.

Fig. 3.3.6 State of SFR and internal RAM areas immediately after reset (5)

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
$A0_{16}$	Pulse output control register	RW	00_{16}
$A1_{16}$			?
$A2_{16}$	Pulse output data register 0	RW	00_{16}
$A3_{16}$			?
$A4_{16}$	Pulse output data register 1	RW	00_{16}
$A5_{16}$			?
$A6_{16}$	Waveform output mode register	RW	00_{16}
$A7_{16}$	Dead-time timer	WO	?
$A8_{16}$	Three-phase output data register 0	RW	00_{16}
$A9_{16}$	Three-phase output data register 1	RW	00_{16}
AA_{16}	Position-data-retain function control register	– – – – RW RO RO RO	? 0 0 0 0
AB_{16}			?
AC_{16}	Serial I/O pin control register	– – RW RW RW RW RW RW	0 0 0 0 0 0 0 0
AD_{16}			?
AE_{16}	Port P2 pin function control register	RW – – – – RW RW RW	0 ? ? ? ? 0 0 0
AF_{16}			?
$B0_{16}$	UART2 transmit/receive mode register	RW	00_{16}
$B1_{16}$	UART2 baud rate register (BRG2)	WO	?
$B2_{16}$	UART2 transmit buffer register	WO	?
$B3_{16}$		– – – – – – – WO	?
$B4_{16}$	UART2 transmit/receive control register 0	RW RW RW RW RO RW RW RW	0 0 0 0 1 0 0 0
$B5_{16}$	UART2 transmit/receive control register 1	RO RO RO RO RO RW RO RW	0 0 0 0 0 0 1 0
$B6_{16}$	UART2 receive buffer register	RO	?
$B7_{16}$		– – – – – – – RO	0 0 0 0 0 0 0 ?
$B8_{16}$		**(Note 16)**	?
$B9_{16}$			?
BA_{16}		**(Note 16)**	?
BB_{16}		**(Note 16)**	?
BC_{16}	Clock control register 0	RW RW RW RW **(Note 17)** RW RW	0 0 0 1 0 1 1 1
BD_{16}		**(Note 16)**	?
BE_{16}		**(Note 16)**	?
BF_{16}		**(Note 16)**	?

Notes 16 : Do not write to this register.
17 : After reset, these bits are allowed to be changed only once.

Fig. 3.3.7 State of SFR and internal RAM areas immediately after reset (6)

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
C0$_{16}$			?
C1$_{16}$			?
C2$_{16}$			?
C3$_{16}$			?
C4$_{16}$	Up-down register 1	WO (b7) / RW (b6–b0)	0 0 0 0 0 0 0 0
C5$_{16}$			?
C6$_{16}$	Timer A5 register	**(Note 18)**	?
C7$_{16}$		**(Note 18)**	?
C8$_{16}$	Timer A6 register	**(Note 18)**	?
C9$_{16}$		**(Note 18)**	?
CA$_{16}$	Timer A7 register	**(Note 18)**	?
CB$_{16}$		**(Note 18)**	?
CC$_{16}$	Timer A8 register	**(Note 18)**	?
CD$_{16}$		**(Note 18)**	?
CE$_{16}$	Timer A9 register	**(Note 18)**	?
CF$_{16}$		**(Note 18)**	?
D0$_{16}$	Timer A0$_1$ register	WO	?
D1$_{16}$		WO	?
D2$_{16}$	Timer A1$_1$ register	WO	?
D3$_{16}$		WO	?
D4$_{16}$	Timer A2$_1$ register	WO	?
D5$_{16}$		WO	?
D6$_{16}$	Timer A5 mode register	RW	00$_{16}$
D7$_{16}$	Timer A6 mode register	RW	00$_{16}$
D8$_{16}$	Timer A7 mode register	RW	00$_{16}$
D9$_{16}$	Timer A8 mode register	RW	00$_{16}$
DA$_{16}$	Timer A9 mode register	RW	00$_{16}$
DB$_{16}$	A-D control register 2	RW	0 0 0 0 0 0 0 0
DC$_{16}$	Comparator function select register 0	RW	0 0 0 0 0 0 0 0
DD$_{16}$	Comparator function select register 1	RW	0 0 0 0 0 0 0 0
DE$_{16}$	Comparator result register 0	RW	0 0 0 0 0 0 0 0
DF$_{16}$	Comparator result register 1	RW	0 0 0 0 0 0 0 0

Note 18: The access characteristics at addresses C6$_{16}$ to CF$_{16}$ vary according to the timer A's operating mode.
(Refer to **"CHAPTER 7. TIMER A."**)

Fig. 3.3.8 State of SFR and internal RAM areas immediately after reset (7)

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
E016	A-D register 8	(Note 19)	?
E116		(Note 19)	0 0 0 0 0 0 ?
E216	A-D register 9	(Note 19)	?
E316		(Note 19)	0 0 0 0 0 0 ?
E416	A-D register 10	(Note 19)	?
E516		(Note 19)	0 0 0 0 0 0 ?
E616	A-D register 11	(Note 19)	?
E716		(Note 19)	0 0 0 0 0 0 ?
E816		(Note 20)	?
E916		(Note 20)	?
EA16		(Note 20)	?
EB16		(Note 20)	?
EC16		(Note 20)	?
ED16		(Note 20)	?
EE16		(Note 20)	?
EF16		(Note 20)	?
F016			?
F116	UART2 transmit interrupt control register	RW	? 0 0 0 0
F216	UART2 receive interrupt control register	RW	? 0 0 0 0
F316			?
F416			?
F516	Timer A5 interrupt control register	RW	? 0 0 0 0
F616	Timer A6 interrupt control register	RW	? 0 0 0 0
F716	Timer A7 interrupt control register	RW	? 0 0 0 0
F816	Timer A8 interrupt control register	RW	? 0 0 0 0
F916	Timer A9 interrupt control register	RW	? 0 0 0 0
FA16		(Note 20)	?
FB16		(Note 20)	?
FC16		(Note 20)	?
FD16	$\overline{INT_5}$ interrupt control register	RW	? 0 0 0 0 0 0
FE16	$\overline{INT_6}$ interrupt control register	RW	? 0 0 0 0 0 0
FF16	$\overline{INT_7}$ interrupt control register	RW	? 0 0 0 0 0 0

Notes 19: The access characteristics at addresses E016 to E716 vary according to the contents of the comparator function select register 1 (address DD16). (Refer to **"CHAPTER 12. A-D CONVERTER."**)
20: Do not write to this register.

● Internal RAM area

- At hardware reset .... Retains the state immediately before reset **(Note 21)**.
- At software reset.. Retains the state immediately before reset.
- At termination of the stop or wait mode
  (when hardware reset is used for the termination.)...................................Retains the state immediately before the **STP** or **WIT** instruction is executed.
- At power-on reset... Undefined.

Notes 21 : When a reset operation starts while writing to the internal RAM area is in process, the microcomputer will be reset before the completion of writing. Accordingly, the contents of the area where the writing was in process will become undefined.

Fig. 3.3.9 State of SFR and internal RAM areas immediately after reset (8)

3.4 Internal processing sequence after reset

Figure 3.4.1 shows the internal processing sequence after reset.

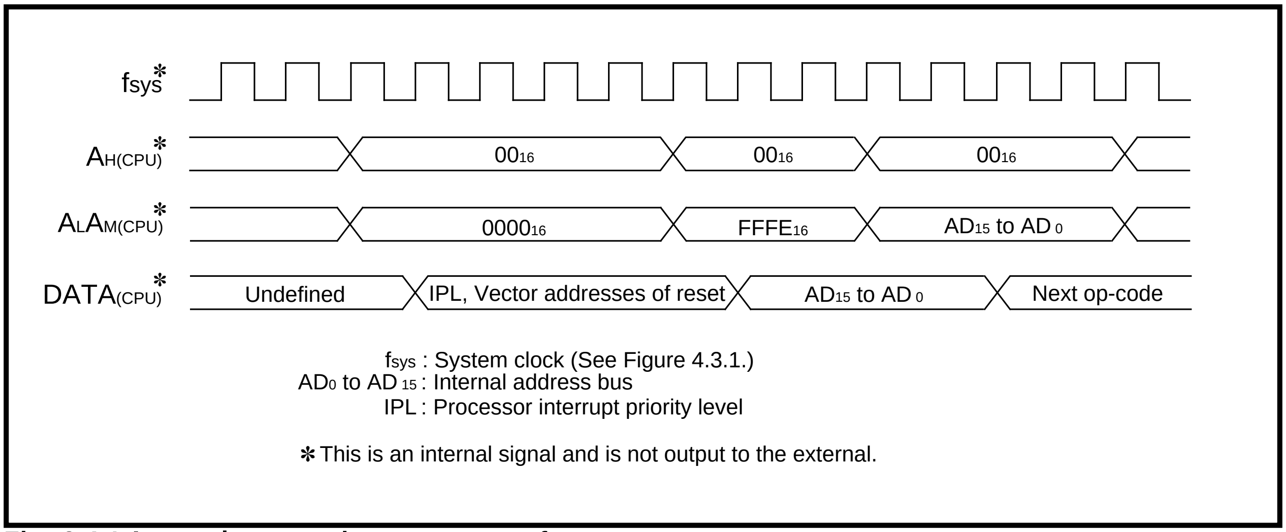


Fig. 3.4.1 Internal processing sequence after reset

MEMORANDUM

CHAPTER 4

CLOCK GENERATING CIRCUIT

4.1 Oscillation circuit examples
4.2 Clocks
[Precautions for clcok generating circuit]

4.1 Oscillation circuit examples

To the oscillation circuit, a ceramic resonator or a quartz-crystal oscillator can be connected, or the clock which is externally generated can be input. Oscillation circuit examples are shown below.

4.1.1 Connection example with resonator/oscillator

Figure 4.1.1 shows an example where pins X_{IN} and X_{OUT} connect across a ceramic resonator/quartz-crystal oscillator.
The circuit constants such as R_f, R_d, C_{IN}, and C_{OUT} (shown in **"Figure 4.1.1"**) depend on the resonator/oscillator. These values shall be set to the values recommended by the resonator/oscillator manufacturer.

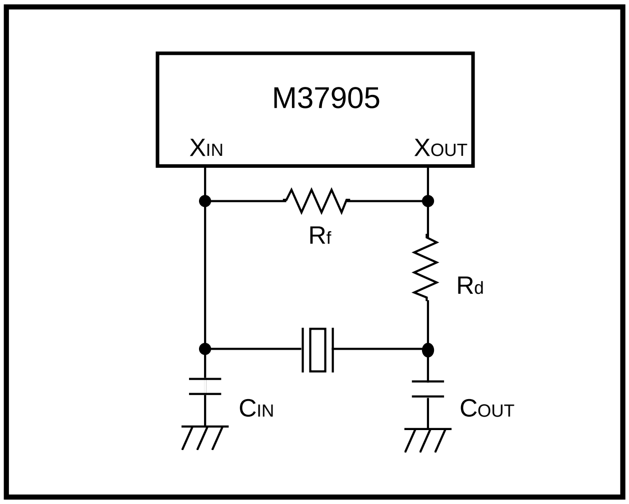


Fig. 4.1.1 Connection example of resonator/oscillator

4.1.2 Externally generated clock input example

Figure 4.1.2 shows an input example of a clock which is externally generated. An external clock must be input from pin X_{IN}, and pin X_{OUT} must be left open.
When an externally generated clock is input, the power source current consumption can be saved by the stop of internal circuit's operation between pins X_{IN} and X_{OUT}. (Refer to **"CHAPTER 16. POWER SAVING FUNCTION."**)

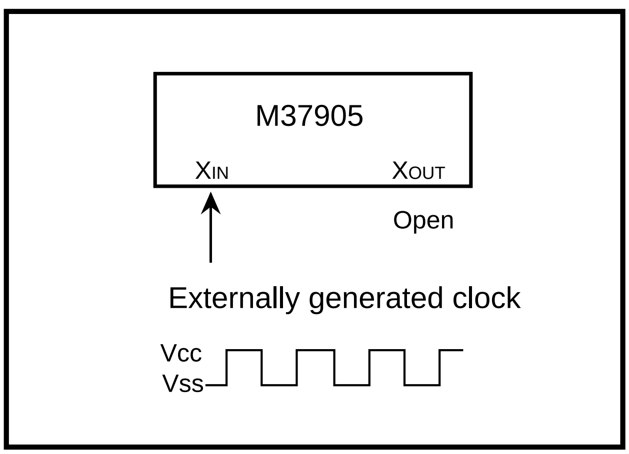
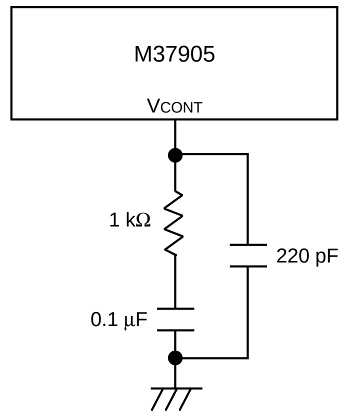


Fig. 4.1.2 Externally generated clock input example

4.1.3 Connection example of filter circuit

In the usage of the PLL frequency multiplier, be sure to connect a filter circuit with pin V_{CONT}.
Figure 4.1.3 shows a connection example of the filter circuit.



Note: Connect the elements of the filter circuit as close as possible and enclose the whole circuit with a Vss pattern.

Fig. 4.1.3 Connection example of filter circuit

4.2 Clocks

Figure 4.2.1 shows the clock generating circuit block diagram.

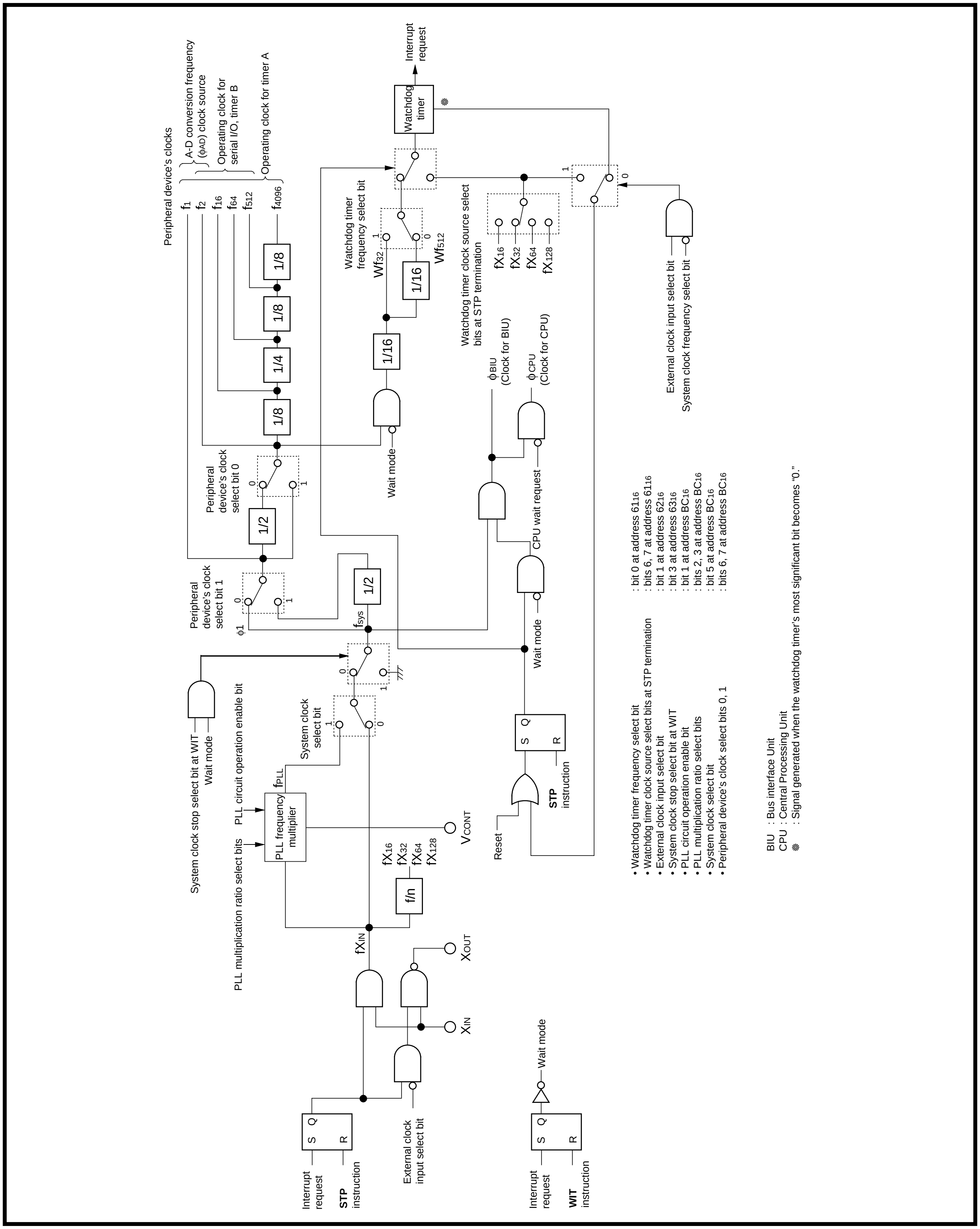


Fig. 4.2.1 Clock generating circuit block diagram

4.2.1 Clocks generated in clock generating circuit

(1) fX_{IN}
It is the input clock from pin X_{IN}.

(2) f_{PLL}
It is the output clock from the PLL frequency multiplier.

(3) f_{sys}
It is the system clock which becomes the clock source of CPU, BIU, and internal peripheral devices. Whether $fX_{IN} = f_{sys}$ or $f_{PLL} = f_{sys}$ can be selected by software.

(4) ϕ_{CPU}
It is the operating clock of CPU.

(5) ϕ_{BIU}
It is the operating clock of BIU.

(6) Clock ϕ_1
It has the same period as f_{sys}.

(7) f_1, f_2, f_{16}, f_{64}, f_{512}, f_{4096}
Each of them is the internal peripheral device's operating clock.

(8) Wf_{32}, Wf_{512}
These are the operating clocks of the watchdog timer, and their clock source is f_2.

(9) fX_{16}, fX_{32}, fX_{64}, fX_{128}
Each of them is the divide clock of fX_{IN} and becomes the watchdog timer's clock source at STP termination.

4.2.2 Clock control register 0

Figure 4.2.2 shows the structure of the clock control register 0, and Figure 4.2.3 shows the setting procedure for the clock control register 0 when using the PLL frequency multiplier.

Clock control register 0 (Address BC_{16})

b7	b6	b5	b4	b3	b2	b1	b0
			1				1

Bit	Bit name	Function	At reset	R/W
0	Fix this bit to "1."		1	RW
1	PLL circuit operation enable bit **(Note 1)**	0 : PLL frequency multiplier is inactive, and pin V_{CONT} is invalid. (Floating) 1 : PLL frequency multiplier is active, and pin V_{CONT} is valid.	1	RW
2	PLL multiplication ratio select bits **(Note 2)**	b3 b2 0 0 : Do not select. 0 1 : × 2 1 0 : × 3 1 1 : × 4	1	RW
3			0	RW
4	Fix this bit to "1."		1	RW
5	System clock select bit **(Note 3)**	0 : fX_{IN} 1 : f_{PLL}	0	RW
6	Peripheral device's clock select bit 0	See Table 4.2.2.	0	RW
7	Peripheral device's clock select bit 1		0	RW

Notes 1: Clear this bit to "0" if the PLL frequency multiplier needs not to be active.
In the stop and flash memory parallel I/O modes, the PLL frequency multiplier is inactive and pin V_{CONT} is invalid regardless of the contents of this bit.
2: Rewriting of these bits must be performed simultaneously with clearance of the system clock select bit (bit 5) to "0." Then, set bit 5 to "1" 2 ms after the rewriting of these bits. (After reset, these bits are allowed to be changed only once.)
3: Clearance of the PLL circuit operation enable bit (bit 1) to "0" clears the system clock select bit to "0." Also, while the PLL circuit operation enable bit = "0," nothing can be written to the system clock select bit. (Fixed to be "0.")
Before setting of set the system clock select bit to "1" after reset, it is necessary to insert an interval of 2 ms after the stabilization of $f(X_{IN})$.

Fig. 4.2.2 Structure of clock control register

(1) PLL circuit operation enable bit (bit 1)

Setting this bit to "1" enables the PLL frequency multiplier to be active and pin V_{CONT} to be valid. This bit = "1" while pin $\overline{RESET}$ = "L" level and after reset, so that, in this case, the PLL frequency multiplier is active. Clear this bit to "0" if the PLL frequency multiplier need not to be active.
Note that, in the stop and flash memory parallel I/O modes, the PLL frequency multiplier is in active and pin V_{CONT} is invalid regardless of the contents of this bit. (Refer to sections **"15.3 Stop mode"** and **"19.4 Flash memory parallel I/O mode."**)

(2) PLL multiplication ratio select bits (bits 2, 3)

These bits select the multiplication ratio of the PLL frequency multiplier. (See Table 4.2.1.) To rewrite these bits, clear the system clock select bit (bit 5) to "0" simultaneously. Then, set the system clock select bit to "1" 2 ms after the rewriting of this bit. (See Figure 4.2.3.)
Note that, after reset, these bits are allowed to be changed only once.

(3) System clock select bit (bit 5)

This bit selects a clock source of f_{sys}. When this bit = “0,” fX_{IN} is selected as f_{sys}; and when this bit = “1,” f_{PLL} as the one. (See Table 4.2.1.)
Clearing the PLL circuit operation enable bit (bit 1) to “0” clears the system clock select bit to “0.” Also, while the PLL circuit operation enable bit = “0,” nothing can be written to the system clock select bit. (Fixed to be “0.”)
In order to set the system clock select bit to “1” after reset, it is necessary to wait 2 ms after the stabilization of $f(X_{IN})$.
To rewrite the PLL multiplication ratio select bits (bits 2 and 3), clear the system clock select bit to “0” simultaneously. Then, set this bit to “1” 2 ms after the rewriting of the PLL multiplication ratio select bits. (See Figure 4.2.3.)

Table 4.2.1 f_{sys} selection

System clock select bit (bit 5)	PLL circuit operation enable bit (bit 1)	PLL multiplication ratio select bits (bits 3, 2) **(Note 1)**	f_{sys}	
			Clock source	Frequency **(Note 2)**
0	–	–	fX_{IN}	$f(X_{IN})$
1	1	01 (double)	f_{PLL}	$f(X_{IN}) \times 2$
		10 (triple)	f_{PLL}	$f(X_{IN}) \times 3$
		11 (quadruple)	f_{PLL}	$f(X_{IN}) \times 4$

Notes 1: The PLL multiplication ratio select bits must be set so that f_{sys} is in the range from 10 MHz to 20 MHz. After reset, these bits are allowed to be changed only once.
2: Be sure that f_{sys} does not exceed 20 MHz.

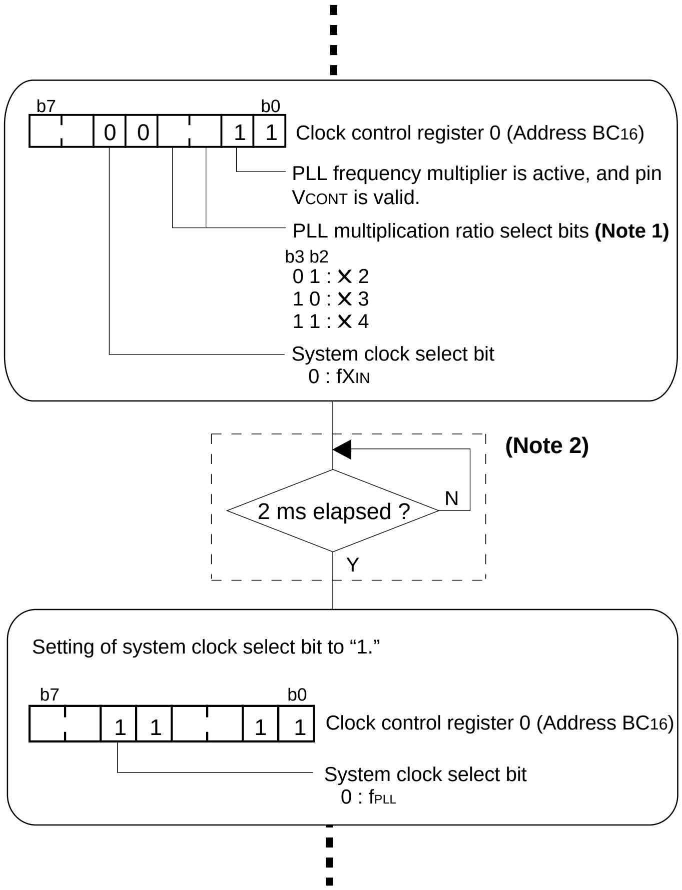
(4) Peripheral device’s clock select bits 1, 0 (bits 7, 6)

These bits select the internal peripheral device’s operation clock frequency listed in Table 4.2.2.

Table 4.2.2 Internal peripheral device’s operation clock frequency

Internal peripheral device’s operation clock	Peripheral device’s clock select bits 1, 0			
	00	01 **(Note)**	10	11
f_1	f_{sys}	f_{sys}	$f_{sys}/2$	Do not select.
f_2	$f_{sys}/2$	f_{sys}	$f_{sys}/4$	
f_{16}	$f_{sys}/16$	$f_{sys}/8$	$f_{sys}/32$	
f_{64}	$f_{sys}/64$	$f_{sys}/32$	$f_{sys}/128$	
f_{512}	$f_{sys}/512$	$f_{sys}/256$	$f_{sys}/1024$	
f_{4096}	$f_{sys}/4096$	$f_{sys}/2048$	$f_{sys}/8192$	

Note: To set the peripheral device’s clock select bits 1, 0 to “01_2,” be sure that a frequency of f_{sys} must be 10 MHz or less.



Notes 1: After reset, these bits are allowed to be changed only once. If it is necessary to write a certain value to these bits, be sure to write the same value that has been written after the latest reset.

2: This decision is unnecessary If double is selected and the period of $\overline{\text{RESET}}$ = "L" is "the oscillation stabilizing time of an oscillator + 2 ms" or more.

Fig. 4.2.3 Setting procedure for clock control register 0 when using PLL frequency multiplier

4.2.3 Particular function select register 0

Figure 4.2.4 shows the structure of the particular function select register 0, and Figure 4.2.5 shows the writing procedure for the particular function select register 0.

Particular function select register 0 (Address 62_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0		

Bit	Bit name	Function	At reset	R/W
0	STP instruction invalidity select bit	0 : STP instruction is valid. 1 : STP instruction is invalid.	0	RW **(Note)**
1	External clcok input select bit	0 : Oscillation circuit is active. (Oscillator is connected.) Watchdog timer is used at stop mode termination. 1 : Oscillation circuit is inactive. (External clock is input.) When the system clock select bit (bit 5 at address BC_{16}) = "0," watchdog timer is not used at stop mode termination. When the system clock select bit = "1," watchdog timer is used at stop mode termination.	0	RW **(Note)**
7 to 2	Fix this bit to "0."		0	RW

Note: Writing to these bits requires the following procedure:
- Write "55_{16}" to this register. (The bit status does not change only by this writing.)
- Succeedingly, write "0" or "1" to each bit.

Also, use the **MOVMB** (**MOVM** when m = 1) instruction or **STAB** (**STA** when m = 1) instruction.
If an interrupt occurs between writing of "55_{16}" and next writing of "0" or "1," latter writing may be ignored. When there is a possibility that an interrupt occurs at the above timing, be sure to read this bit's contents after writing of "0" or "1," and verify whether "0" or "1" has correctly been written or not.

Fig. 4.2.4 Structure of particular function select register 0

(1) External clock input select bit (bit 1)

When this bit is "0," the oscillation driver circuit between pins X_{IN} and X_{OUT} operates. At the stop mode termination owing to an interrupt request occurrence, the watching timer is used.

Setting this bit to "1" stops the oscillation driver circuit between pins X_{IN} and X_{OUT} and keeps the output level at pin X_{OUT} being "H." (Refer to section **"16.3 Stop of oscillation circuit."**) At the stop mode termination owing to an interrupt request occurrence, the watchdog timer is not used if the system clock select bit (bit 5 at address BC_{16}) = "0," where as the watchdog timer is used if the system clock select bit = "1."

To rewrite this bit, write "0" or "1" just after writing of "55_{16}" to address 62_{16}. (See Figure 4.2.5.)

Note that if an interrupt occurs between writing of "55_{16}" and next writing of "0" or "1," latter writing may be ignored. When there is a possibility that an interrupt occurs at the above timing, be sure to read this bit's contents after writing of "0" or "1," and verify whether "0" or "1" has correctly been written or not.

In addition, even when the watchdog timer is disabled by the particular function select register 2 (address 64_{16}), the watchdog timer can be active only at the stop mode termination if this bit = "0." (Refer to section **"15.3 Stop mode."**)

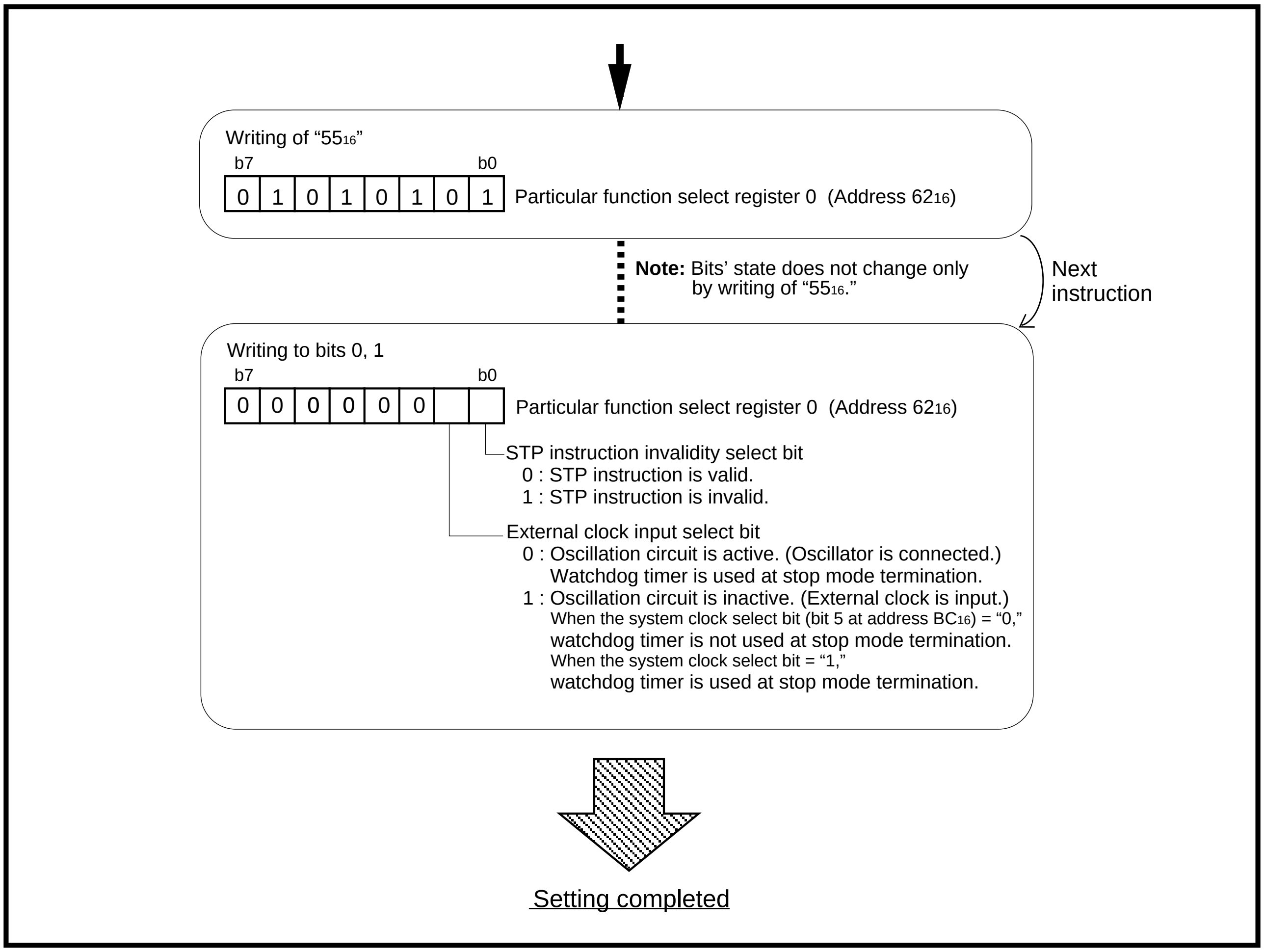


Fig. 4.2.5 Writing procedure for particular function select register 0

[Precautions for clock generating circuit]

1. While pin $\overline{RESET}$ = “L” level and after reset, the PLL frequency multiplier is inactive. Clear the PLL circuit operation enable bit (bit 1 at address BC_{16}) to “0” if the PLL frequency multiplier needs not to be active.

2. To select f_{PLL} as f_{sys} after reset, set the system clock select bit (bit 5 at address BC_{16}) to “1” 2 ms after $f(X_{IN})$ has been stabilized. (See Figure 4.2.3.)

3. To change the multiplication ratio for the PLL frequency multiplier, clear the system clock select bit (bit 5 at address BC_{16}) to “0” simultaneously. Then, set the system clock select bit to “1” 2 ms after the rewriting of the PLL multiplication ratio select bits (bits 2, 3 at address BC_{16}). (See Figure 4.2.3.)
   After reset, the PLL multiplication ratio select bits are allowed to be changed only once. If it is necessary to write a certain value to these bits, be sure to write the same value that has been written after the latest reset.

MEMORANDUM

CHAPTER 5
INPUT/OUTPUT PINS

5.1 Overview

Input/output pins (hereafter called I/O pins) have functions as programmable I/O port pins, internal peripheral devices's I/O pins, etc.
For the basic functions of each I/O pin, refer to section **"1.3 Pin description."** For the I/O functions of the internal peripheral devices, refer to relevant sections of each internal peripheral device.
This chapter describes the programmable I/O ports and examples of handling unused pins.

5.2 Programmable I/O ports

The programmable I/O ports have direction registers and port registers in the SFR area. Figure 5.2.1 shows the memory map of direction registers and port registers.

Addresses	
3_{16}	Port P1 register
4_{16}	**(Note)**
5_{16}	Port P1 direction register
6_{16}	Port P2 register
7_{16}	**(Note)**
8_{16}	Port P2 direction register
9_{16}	**(Note)**
A_{16}	Port P4 register
B_{16}	Port P5 register
C_{16}	Port P4 direction register
D_{16}	Port P5 direction register
E_{16}	Port P6 register
F_{16}	Port P7 register
10_{16}	Port P6 direction register
11_{16}	Port P7 direction register
12_{16}	Port P8 register
13_{16}	**(Note)**
14_{16}	Port P8 direction register

Note: Do not write to this address.

Fig. 5.2.1 Memory map of direction registers and port registers

5.2.1 Direction register

This register determines the I/O direction of programmable I/O ports. One bit of this register corresponds to one pin of the microcomputer, and this is the one-to-one relationship.
Figure 5.2.2 shows the structure of port Pi (i = 1, 2, 4 to 8) direction register.

Port Pi direction register (i = 1, 2, 4 to 8)
(Addresses 5_{16}, 8_{16}, C_{16}, D_{16}, 10_{16}, 11_{16}, 14_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Port Pi_0 direction bit	0 : Input mode (The port functions as an input port.) 1 : Output mode (The port functions as an output port.)	0	RW
1	Port Pi_1 direction bit		0	RW
2	Port Pi_2 direction bit		0	RW
3	Port Pi_3 direction bit		0	RW
4	Port Pi_4 direction bit		0	RW
5	Port Pi_5 direction bit		0	RW
6	Port Pi_6 direction bit		0	RW
7	Port Pi_7 direction bit		0	RW

Notes 1: Nothing is assigned for bits 0 and 4 of the port P5 direction register. These bits are undefined at reading.
2: Nothing is assigned for bits 4 to 7 of the port P8 direction register. These bits are undefined at reading.
3: Any of bits 0 to 7 of the port P4 direction register becomes "0" by input of a falling edge to pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$. (Refer to section **"5.2.3 Pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$."**)
4: Any of bits 0 to 7 of the port P6 direction register becomes "0" by input of a falling edge to pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$. (Refer to section **"5.2.4 Pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$."**)

Fig. 5.2.2 Structure of port Pi (i = 1, 2, 4 to 8) direction register

5.2.2 Port register

Data is input from or output to the external by writing/reading data to/from a port register. A port register consists of a port latch which holds the output data and a circuit which reads the pin state. One bit of the port register corresponds to one pin of the microcomputer. (This is the one-to-one relationship.) Figure 5.2.3 shows the structure of the port Pi (i = 1, 2, 4 to 8) register.

● **When outputting data from programmable I/O port which has been set to output mode**

① By writing data to the corresponding bit of the port register, the data is written into the port latch.
② The data is output from the pin according to the contents of the port latch.

By reading the port register of a port which has been set to the output mode, the contents of the port latch is read out, instead of the pin state. Accordingly, the output data can be correctly read out without being affected by an external load, etc. (See **"Figures 5.2.4 and 5.2.5."**)

● **When inputting data from programmable I/O port which has been set to input mode**

① A pin which has been set to the input mode enters the floating state.
② By reading the corresponding bit of the port register, the data which has been input from the pin can be read out.

By writing data to a port register of a programmable I/O port which has been set to the input mode, the data is written only into the port latch and is not output to the external **(Note)**. This pin remains floating state.

Note: When executing a read-modify-write instruction to a port register of a programmable I/O port which has been set to the input mode, the instruction will be executed to the data which has been input from the pin and the result will be written into the port register.

Port Pi register (i = 1, 2, 4 to 8)
(Addresses 3_{16}, 6_{16}, A_{16}, B_{16}, E_{16}, F_{16}, 12_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Funtion	At reset	R/W
0	Port pin Pi_0	Data is input from or output to a pin by reading from or writing to the corresponding bit. 0 : "L" level 1 : "H" level	Undefined	RW
1	Port pin Pi_1		Undefined	RW
2	Port pin Pi_2		Undefined	RW
3	Port pin Pi_3		Undefined	RW
4	Port pin Pi_4		Undefined	RW
5	Port pin Pi_5		Undefined	RW
6	Port pin Pi_6		Undefined	RW
7	Port pin Pi_7		Undefined	RW

Notes 1: Nothing is assigned for bits 0 and 4 of the port P5 register. These bits are undefined at reading.
2: Nothing is assigned for bits 4 to 7 of the port P8 register. These bits are undefined at reading.

Fig. 5.2.3 Structure of port Pi (i = 1, 2, 4 to 8) register

Figures 5.2.4 to 5.2.6 show the port peripheral circuits.

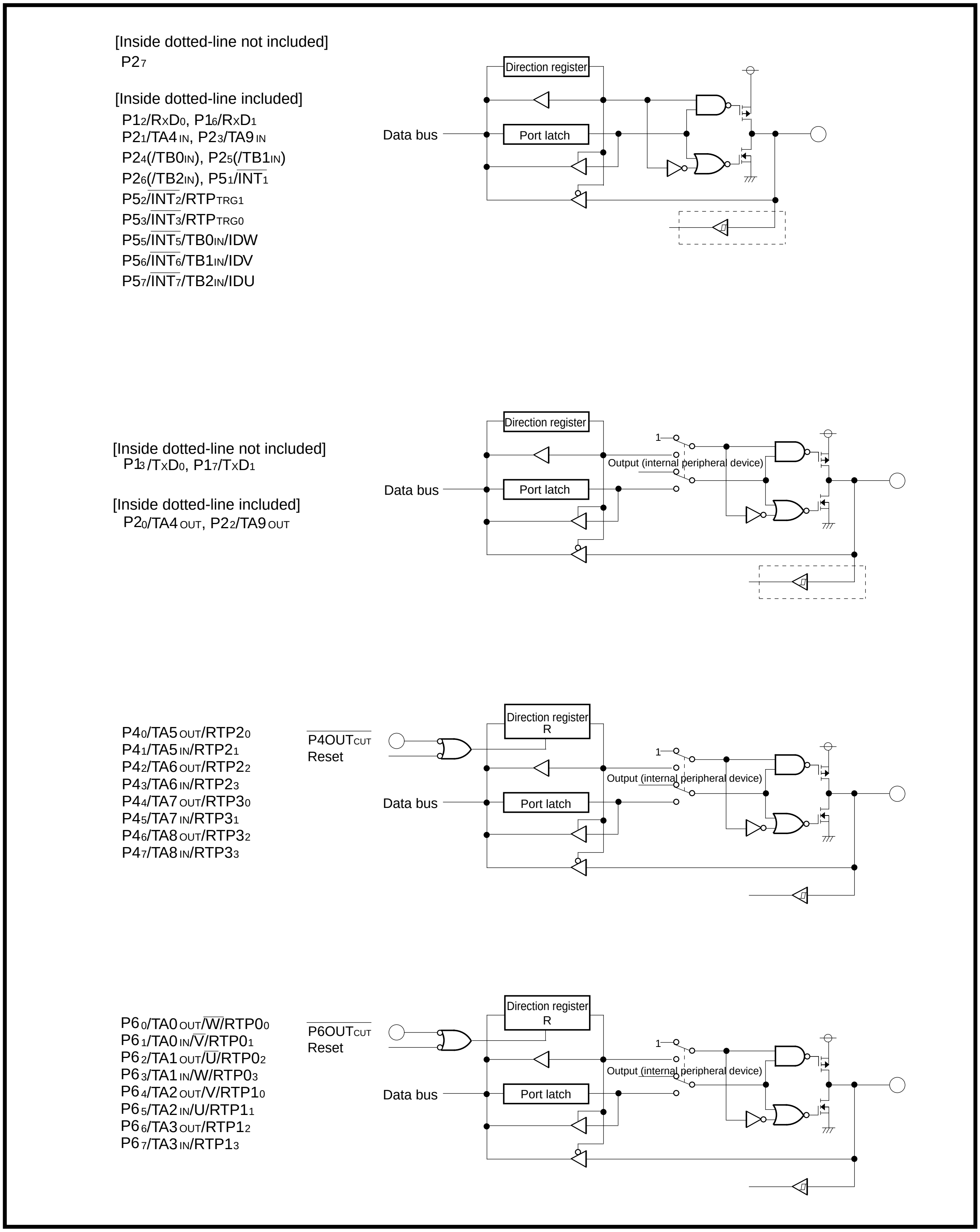


Fig. 5.2.4 Port peripheral circuits (1)

5.2 Programmable I/O ports

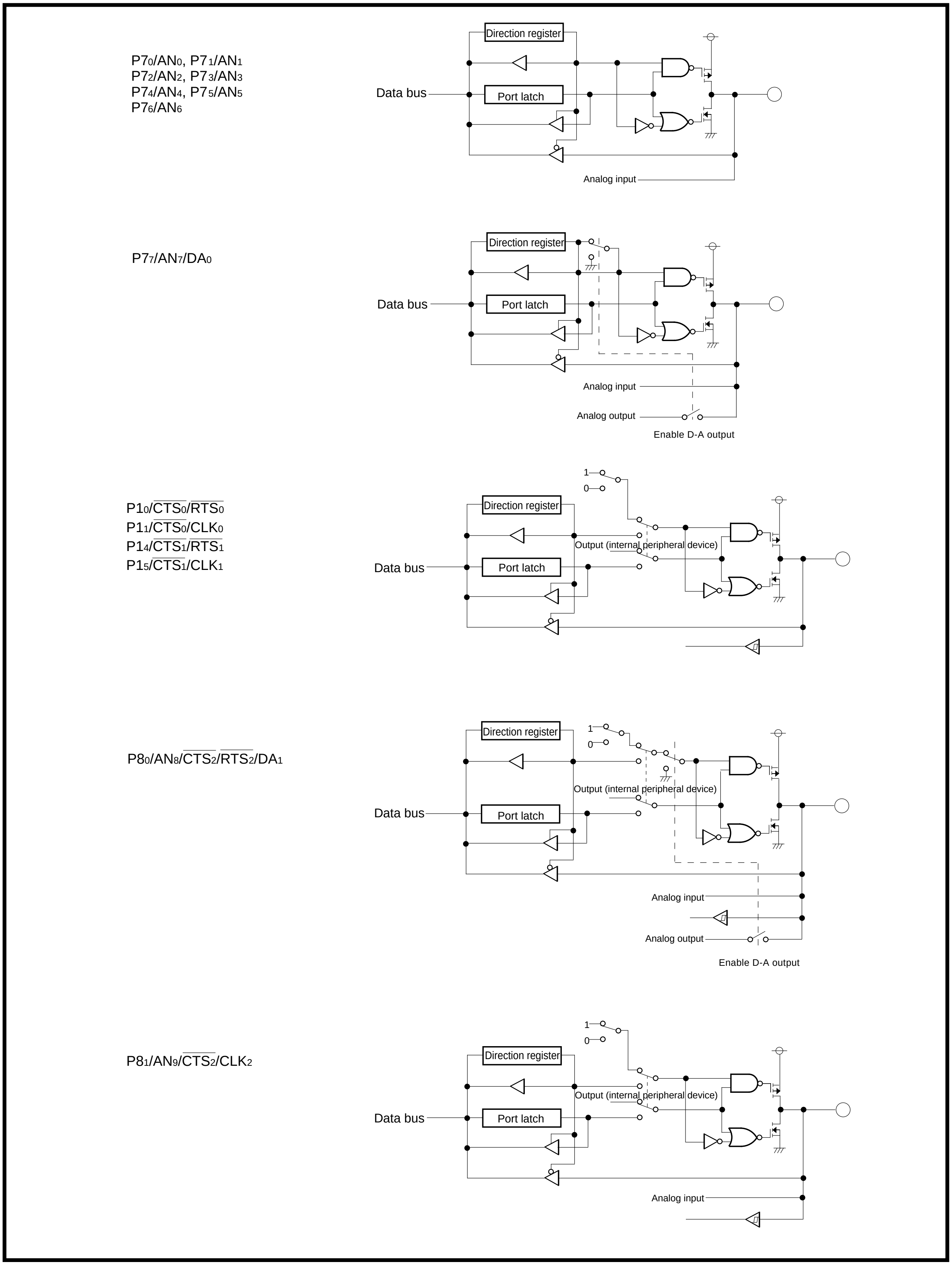


Fig. 5.2.5 Port peripheral circuits (2)

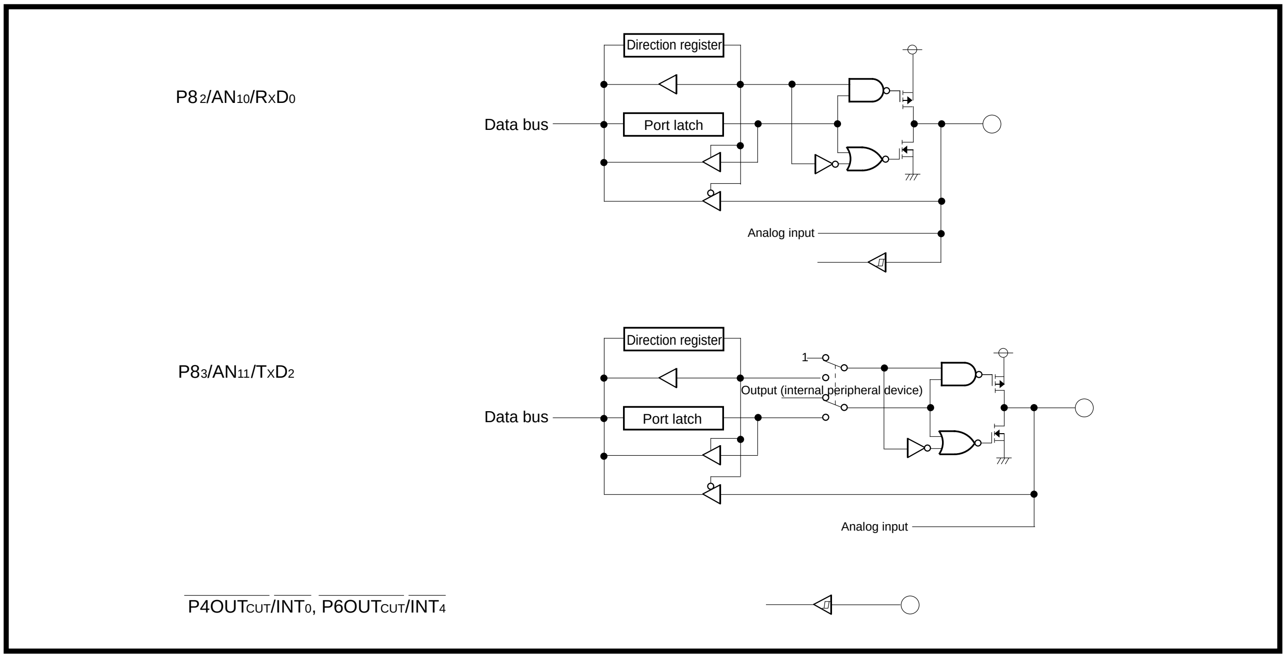


Fig. 5.2.6 Port peripheral circuits (3)

5.2.3 Pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$ (Port-P4-output-cutoff signal input pin)

Any of bits 0 through 7 of the port P4 direction register (address C_{16}) are forcibly cleared to “0” by input of a falling edge to pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$, regardless of the mode of port pins $P4_0$ through $P4_7$; therefore, port pins $P4_0$ through $P4_7$ enter the input mode. After that, if it is necessary to output data from port pins $P4_0$ through $P4_7$, be sure to do as follows:

① Return the input level at pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$ to “H” level.

② Write data to the port P4 register (address A_{16})’s bits, corresponding to the port P4 pins which will output data.

③ Set the port P4 direction register’s bits, corresponding to the port P4 pins in ②, to “1” in order to set these port pins to the output mode.

When input level at pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$ is “L”, no bit of the port P4 direction register can be set to “1.”
When using port pins $P4_0$ through $P4_7$ as output port pins at all the time, connect pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$ to Vcc via a resistor. Pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$ cannot serve as pin $\overline{INT_0}$.
Also, when using pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$ as an input pin of an external interrupt (pin $\overline{INT_0}$), use port pins $P4_0$ through $P4_7$ in the input mode.

5.2.4 Pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$ (Port-P6-output-cutoff signal input pin)

Any of bits 0 through 7 of the port P6 direction register (address 10_{16}) are forcibly cleared to “0” by input of a falling edge to pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$, regardless of the mode of port pins $P6_0$ through $P6_7$; therefore, port pins $P6_0$ through $P6_7$ enter the input mode. After that, if it is necessary to output data from port pins $P6_0$ through $P6_7$, be sure to do as follows:

① Return the input level at pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$ to “H” level.

② Write data to the port P6 register (address E_{16})’s bits, corresponding to the port P6 pins which will output data.

③ Set the port P6 direction register’s bits, corresponding to the port P6 pins in ②, to “1” in order to set these port pins to the output mode.

When input level at pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$ is “L”, no bit of the port P6 direction register can be set to “1.”
When using port pins $P6_0$ through $P6_7$ as output port pins at all the time, connect pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$ to Vcc via a resistor. Pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$ cannot serve as pin $\overline{INT_4}$.
Also, when using pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$ as an input pin of an external interrupt (pin $\overline{INT_4}$), use port pins $P6_0$ through $P6_7$ in the input mode.

5.3 Examples of handling unused pins

When unusing an I/O pin, some handling is necessary for this pin. Examples of handling unused pins are described below.
The following are just examples. In actual use, the user shall modify them according to the user's application and properly evaluate their performance.

Table 5.3.1 Example of handling unused pins

Pin name	Handling example
P1, P2, P4 to P8	Set these pins to the input mode and connect each pin to Vcc or Vss via a resistor; or set these pins to the output mode and leave them open **(Note 1).**
$\overline{P4OUT_{CUT}}/\overline{INT_0}$, $\overline{P6OUT_{CUT}}/\overline{INT_4}$	Connect this pin to Vcc via a resistor. Select a falling edge for pins $\overline{INT_0}$ and $\overline{INT_4}$.
X_{OUT} **(Note 2)**, V_{CONT} **(Note 3)**	Leave these pins open.
AV_{CC}	Connect this pin to Vcc.
AV_{SS}, V_{REF}	Connect these pins to Vss.

Notes 1: When leaving these pins open after they have been set to the output mode, note the following: these port pins are placed in the input mode from reset until they are switched to the output mode by software. Therefore, voltage levels of these pins are undefined and the power source current may increase while these port pins are placed in the input mode.
Software reliability can be enhanced by setting the contents of the above ports' direction registers periodically. This is because these contents may be changed by noise, a program runaway which occurs owing to noise, etc.
For unused pins, use the shortest possible wiring (within 20 mm from the microcomputer's pins).
2: This applies when a clock externally generated is input to pin X_{IN}.
3: Be sure that the PLL circuit operation enable bit (bit 1 at address BC_{16}) = "0."

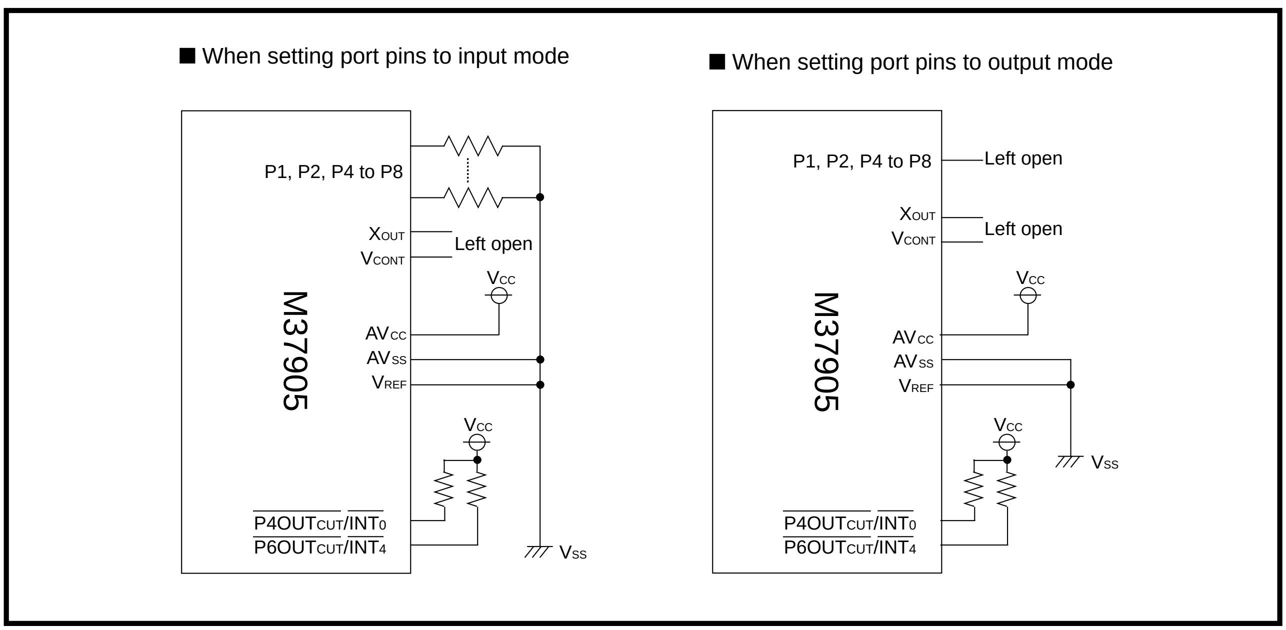


Fig. 5.3.1 Example of handling unused pins

CHAPTER 6

INTERRUPTS

6.1 Overview

The M37905 provides 32 (including the reset) interrupt sources to generate interrupt requests.
Figure 6.1.1 shows the interrupt processing sequence.
When an interrupt request is accepted, a branch is made to the start address of the interrupt routine set in the interrupt vector table (addresses $FFB4_{16}$ to $FFFF_{16}$). Set the start address of each interrupt routine to the corresponding interrupt vector address in the interrupt vector table.

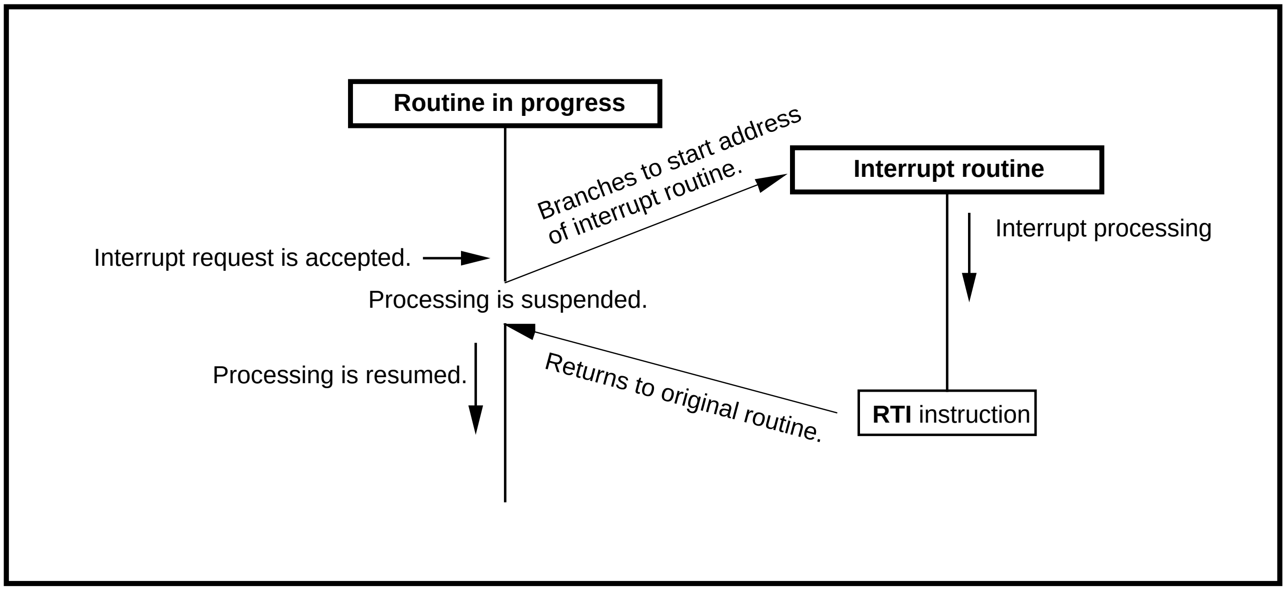


Fig. 6.1.1 Interrupt processing sequence

When an interrupt request is accepted, the following registers' contents just before acceptance of an interrupt request are automatically pushed onto the stack area in ascending sequence from ① to ③.
For other registers of which contents are necessary, be sure to push and pop them by software.

① Program bank register (PG)
② Program counter (PC_L, PC_H)
③ Processor status register (PS_L, PS_H)

Figure 6.1.2 shows the state of the stack area just before entering an interrupt routine.
Execute the **RTI** instruction at the end of this interrupt routine in order to return to the routine that the microcomputer was executing just before the interrupt request was accepted. By executing the **RTI** instruction, the register contents pushed onto the stack area are pulled in descending sequence from ③ to ①. Then, the suspended processing is resumed from where it left off.

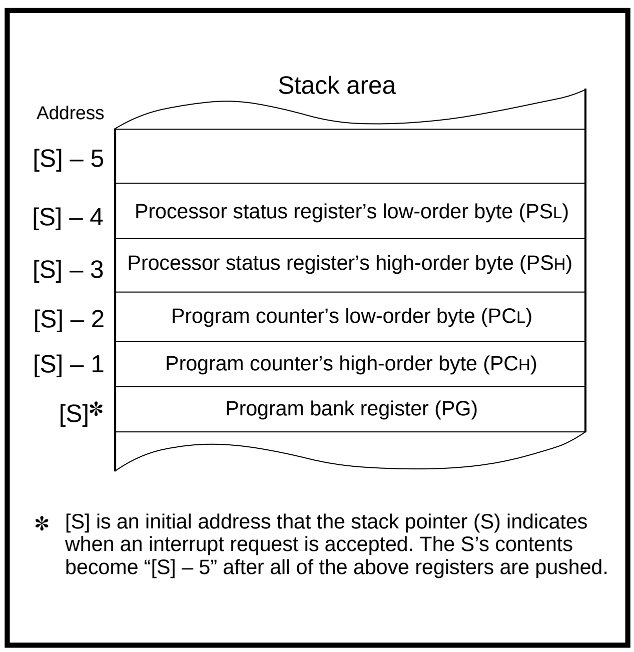


Fig. 6.1.2 State of stack area just before entering interrupt routine

6.2 Interrupt sources

Tables 6.2.1 and 6.2.2 list the interrupt sources and the interrupt vector addresses. When programming, set the start address of each interrupt routine to the vector addresses listed in these tables.

Table 6.2.1 Interrupt sources and interrupt vector addresses (1)

Interrupt source	Interrupt vector addresses High-order address	Interrupt vector addresses Low-order address	Remarks	Reference
Reset	$FFFF_{16}$	$FFFE_{16}$	Non-maskable	3. RESET
Zero division	$FFFD_{16}$	$FFFC_{16}$	Non-maskable software interrupt	7900 Series Software Manual
BRK instruction **(Note)**	$FFFB_{16}$	$FFFA_{16}$	Do not use.	————
$\overline{DBC}$ **(Note)**	$FFF9_{16}$	$FFF8_{16}$		
Watchdog timer	$FFF7_{16}$	$FFF6_{16}$	Non-maskable internal interrupt	14. WATCHDOG TIMER
Reserved area	$FFF5_{16}$	$FFF4_{16}$	Do not use.	————
$\overline{INT_0}$	$FFF3_{16}$	$FFF2_{16}$	Maskable external interrupts	6.10 External interrupts
$\overline{INT_1}$	$FFF1_{16}$	$FFF0_{16}$		
$\overline{INT_2}$	$FFEF_{16}$	$FFEE_{16}$		
Timer A0	$FFED_{16}$	$FFEC_{16}$	Maskable internal interrupts	7. TIMER A
Timer A1	$FFEB_{16}$	$FFEA_{16}$		
Timer A2	$FFE9_{16}$	$FFE8_{16}$		
Timer A3	$FFE7_{16}$	$FFE6_{16}$		
Timer A4	$FFE5_{16}$	$FFE4_{16}$		
Timer B0	$FFE3_{16}$	$FFE2_{16}$	Maskable internal interrupts	8. TIMER B
Timer B1	$FFE1_{16}$	$FFE0_{16}$		
Timer B2	$FFDF_{16}$	$FFDE_{16}$		
UART0 receive	$FFDD_{16}$	$FFDC_{16}$	Maskable internal interrupts	11. SERIAL I/O
UART0 transmit	$FFDB_{16}$	$FFDA_{16}$		
UART1 receive	$FFD9_{16}$	$FFD8_{16}$		
UART1 transmit	$FFD7_{16}$	$FFD6_{16}$		
A-D conversion	$FFD5_{16}$	$FFD4_{16}$	Maskable internal interrupt	12. A-D CONVERTER
$\overline{INT_3}$	$FFD3_{16}$	$FFD2_{16}$	Maskable external interrupts	6.10 External interrupts
$\overline{INT_4}$	$FFD1_{16}$	$FFD0_{16}$		
Reserved area	$FFCF_{16}$	$FFCE_{16}$	Do not use.	————
Reserved area	$FFCD_{16}$	$FFCC_{16}$		
Address matching detection	$FFCB_{16}$	$FFCA_{16}$	Non-maskable software interrupt	17. DEBUG FUNCTION
Reserved area	$FFC9_{16}$	$FFC8_{16}$	Do not use.	————
$\overline{INT_5}$	$FFC7_{16}$	$FFC6_{16}$	Maskable external interrupts	6.10 External interrupts
$\overline{INT_6}$	$FFC5_{16}$	$FFC4_{16}$		
$\overline{INT_7}$	$FFC3_{16}$	$FFC2_{16}$		

Note: The **BRK** instruction and the $\overline{DBC}$ interrupt are used exclusively for a debugger.

- **Maskable interrupt:** An interrupt of which request's acceptance can be disabled by software.
- **Non-maskable interrupt** (including zero division, watchdog timer, and address matching detection interrupts)**:** An interrupt which is certain to be accepted when its request occurs. These interrupts do not have their interrupt control registers and are not affected by the interrupt disable flag (I).

Table 6.2.2 Interrupt sources and interrupt vector addresses (2)

Interrupt source	Interrupt vector addresses		Remarks	Reference
	High-order address	Low-order address		
Timer A5	$FFC1_{16}$	$FFC0_{16}$	Maskable internal interrupts	7. TIMER A
Timer A6	$FFBF_{16}$	$FFBE_{16}$		
Timer A7	$FFBD_{16}$	$FFBC_{16}$		
Timer A8	$FFBB_{16}$	$FFBA_{16}$		
Timer A9	$FFB9_{16}$	$FFB8_{16}$		
UART2 transmit	$FFB7_{16}$	$FFB6_{16}$	Maskable internal interrupts	11. SERIAL I/O
UART2 receive	$FFB5_{16}$	$FFB4_{16}$		

● **Maskable interrupt:** An interrupt of which request's acceptance can be disabled by software.

6.3 Interrupt control

The maskable interrupts are controlled by the following :

- •Interrupt request bit
- •Interrupt priority level select bits

} Assigned to an interrupt control register of each interrupt.

- •Processor interrupt priority level (IPL)
- •Interrupt disable flag (I)

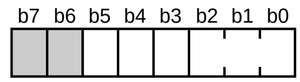
} Assigned to the processor status register (PS).

Figure 6.3.1 shows the memory assignment of the interrupt control registers, and Figures 6.3.2 shows their structures.

Address	
$6E_{16}$	$\overline{INT_3}$ interrupt control register
$6F_{16}$	$\overline{INT_4}$ interrupt control register
70_{16}	A-D conversion interrupt control register
71_{16}	UART0 transmit interrupt control register
72_{16}	UART0 receive interrupt control register
73_{16}	UART1 transmit interrupt control register
74_{16}	UART1 receive interrupt control register
75_{16}	Timer A0 interrupt control register
76_{16}	Timer A1 interrupt control register
77_{16}	Timer A2 interrupt control register
78_{16}	Timer A3 interrupt control register
79_{16}	Timer A4 interrupt control register
$7A_{16}$	Timer B0 interrupt control register
$7B_{16}$	Timer B1 interrupt control register
$7C_{16}$	Timer B2 interrupt control register
$7D_{16}$	$\overline{INT_0}$ interrupt control register
$7E_{16}$	$\overline{INT_1}$ interrupt control register
$7F_{16}$	$\overline{INT_2}$ interrupt control register
≈	≈
$F1_{16}$	UART2 transmit interrupt control register
$F2_{16}$	UART2 receive interrupt control register
$F5_{16}$	Timer A5 interrupt control register
$F6_{16}$	Timer A6 interrupt control register
$F7_{16}$	Timer A7 interrupt control register
$F8_{16}$	Timer A8 interrupt control register
$F9_{16}$	Timer A9 interrupt control register
FD_{16}	$\overline{INT_5}$ interrupt control register
FE_{16}	$\overline{INT_6}$ interrupt control register
FF_{16}	$\overline{INT_7}$ interrupt control register

Fig. 6.3.1 Memory assignment of interrupt control registers

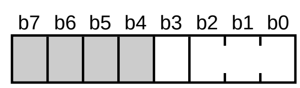
$\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_2}$ interrupt control registers (Addresses $7D_{16}$, $7E_{16}$, $7F_{16}$)
$\overline{INT_3}$, $\overline{INT_4}$ interrupt control registers (Addresses $6E_{16}$, $6F_{16}$)
$\overline{INT_5}$, $\overline{INT_6}$, $\overline{INT_7}$ interrupt control registers (Addresses FD_{16}, FE_{16}, FF_{16})



Bit	Bit name	Function	At reset	R/W
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit **(Note 1)**	0 : No interrupt requested 1 : Interrupt requested	0	RW **(Note 2)**
4	Polarity select bit	0 : The interrupt request bit is set to "1" at "H" level when level sense is selected; this bit is set to "1" at falling edge when edge sense is selected. 1 : The interrupt request bit is set to "1" at "L" level when level sense is selected; this bit is set to "1" at rising edge when edge sense is selected.	0	RW
5	Level sense/Edge sense select bit	0 : Edge sense 1 : Level sense	0	RW
7, 6	Nothing is assigned.		Undefined	—

Notes 1: The interrupt request bits of $\overline{INT_0}$ to $\overline{INT_7}$ interrupts are invalid when the level sense is selected.
2: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

A-D conversion, UART0 and 1 transmit, UART0 and 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers (Addresses 70_{16} to $7C_{16}$)
UART2 transmit, UART2 receive interrupt control registers (Addresses $F1_{16}$, $F2_{16}$)
Timers A5 to A9 interrupt control registers (Addresses $F5_{16}$ to $F9_{16}$)



Bit	Bit name	Function	At reset	R/W
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt requested 1 : Interrupt requested	0 **(Note 1)**	RW **(Note 2)**
7 to 4	Nothing is assigned.		Undefined	—

Notes 1: The A-D conversion interrupt request bit is undefined after reset.
2: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

Fig. 6.3.2 Structure of interrupt control register

6.3.1 Interrupt disable flag (I)
All maskable interrupts can be disabled by this flag. When this flag is set to "1," all maskable interrupts are disabled; when this flag is cleared to "0," those interrupts are enabled. Because this flag is set to "1" at reset, clear this flag to "0" when enabling interrupts.

6.3.2 Interrupt request bit
When an interrupt request occurs, this bit is set to "1." This bit remains set to "1" until the interrupt request is accepted; it is cleared to "0" when the interrupt request is accepted.
This bit can also be set to "0" or "1" by software.
The $\overline{INT_i}$ interrupt request bit (i = 0 to 7) is ignored when the corresponding $\overline{INT_i}$ interrupt is used with the level sense.

6.3.3 Interrupt priority level select bits and Processor interrupt priority level (IPL)
The interrupt priority level select bits are used to determine the priority level of each interrupt.
When an interrupt request occurs, its interrupt priority level is compared with the processor interrupt priority level (IPL). The requested interrupt is enabled only when the comparison result meets the following condition. Accordingly, any interrupt can be disabled by setting its interrupt priority level to 0.

Each interrupt priority level > Processor interrupt priority level (IPL)

Table 6.3.1 lists the setting of interrupt priority levels, and Table 6.3.2 lists the enabled interrupt's levels according to the IPL contents.

The interrupt disable flag (I), interrupt request bit, interrupt priority level select bits, and processor interrupt priority level (IPL) are independent of one another; they do not affect one another. Interrupt requests are accepted only when all of the following conditions are satisfied.

- Interrupt disable flag (I) = "0"
- Interrupt request bit = "1"
- Interrupt priority level > Processor interrupt priority level (IPL)

6.3 Interrupt control

Table 6.3.1 Setting of interrupt priority level

Interrupt priority level select bits			Interrupt priority level	Priority
b2	b1	b0		
0	0	0	Level 0 (Interrupt disabled)	—
0	0	1	Level 1	Low
0	1	0	Level 2	↓
0	1	1	Level 3	
1	0	0	Level 4	
1	0	1	Level 5	
1	1	0	Level 6	
1	1	1	Level 7	High

Table 6.3.2 Enabled interrupt's levels according to IPL contents

IPL_2	IPL_1	IPL_0	Enabled interrupt's level
0	0	0	Level 1 and above are enabled.
0	0	1	Level 2 and above are enabled.
0	1	0	Level 3 and above are enabled.
0	1	1	Level 4 and above are enabled.
1	0	0	Level 5 and above are enabled.
1	0	1	Levels 6 and 7 are enabled.
1	1	0	Only level 7 is enabled.
1	1	1	All maskable interrupts are disabled.

IPL_0: Bit 8 in processor status register (PS)
IPL_1: Bit 9 in processor status register (PS)
IPL_2: Bit 10 in processor status register (PS)

6.4 Interrupt priority level

When the interrupt disable flag (I) = “0” (interrupts enabled) and more than one interrupt request is detected at the same sampling timing, which means a timing to check whether an interrupt request exists or not, they are accepted in descending sequence from the highest priority level.
A maskable interrupt can be set to the desired priority level by using the interrupt priority level select bits. The priority levels of reset and a watchdog timer interrupt are set by hardware. Figure 6.4.1 shows the interrupt priority levels set by hardware.

Note that software interrupts are not affected by the interrupt priority levels. Whenever an instruction is executed, a branch is certainly made to the interrupt routine.

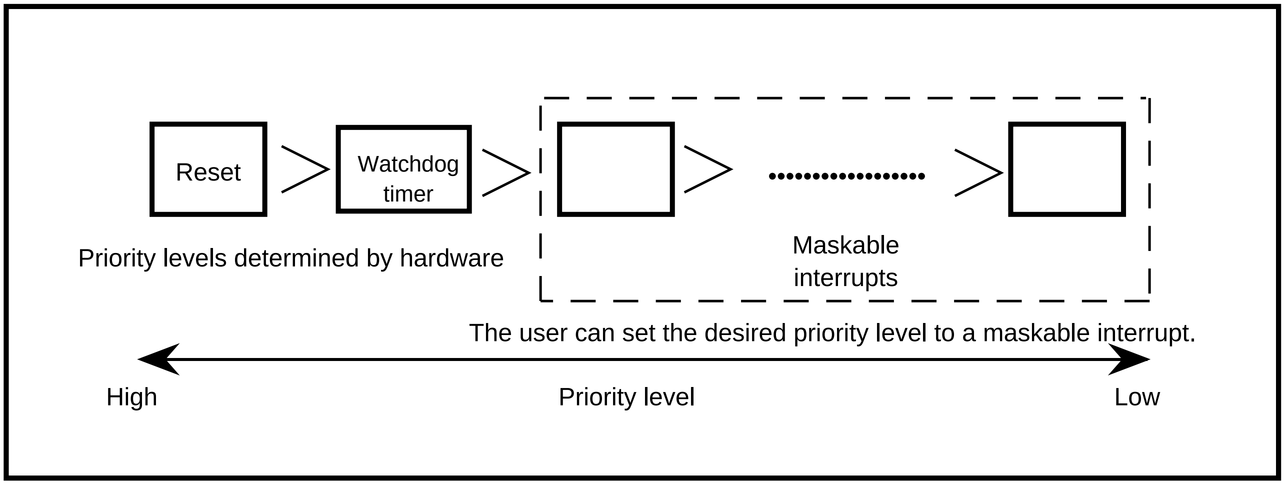


Fig. 6.4.1 Interrupt priority levels set by hardware

6.5 Interrupt priority level detection circuit

The interrupt priority level detection circuit is used to select the interrupt with the highest priority level from multiple interrupt requests sampled at the same timing. Figure 6.5.1 shows the interrupt priority level detection circuit.

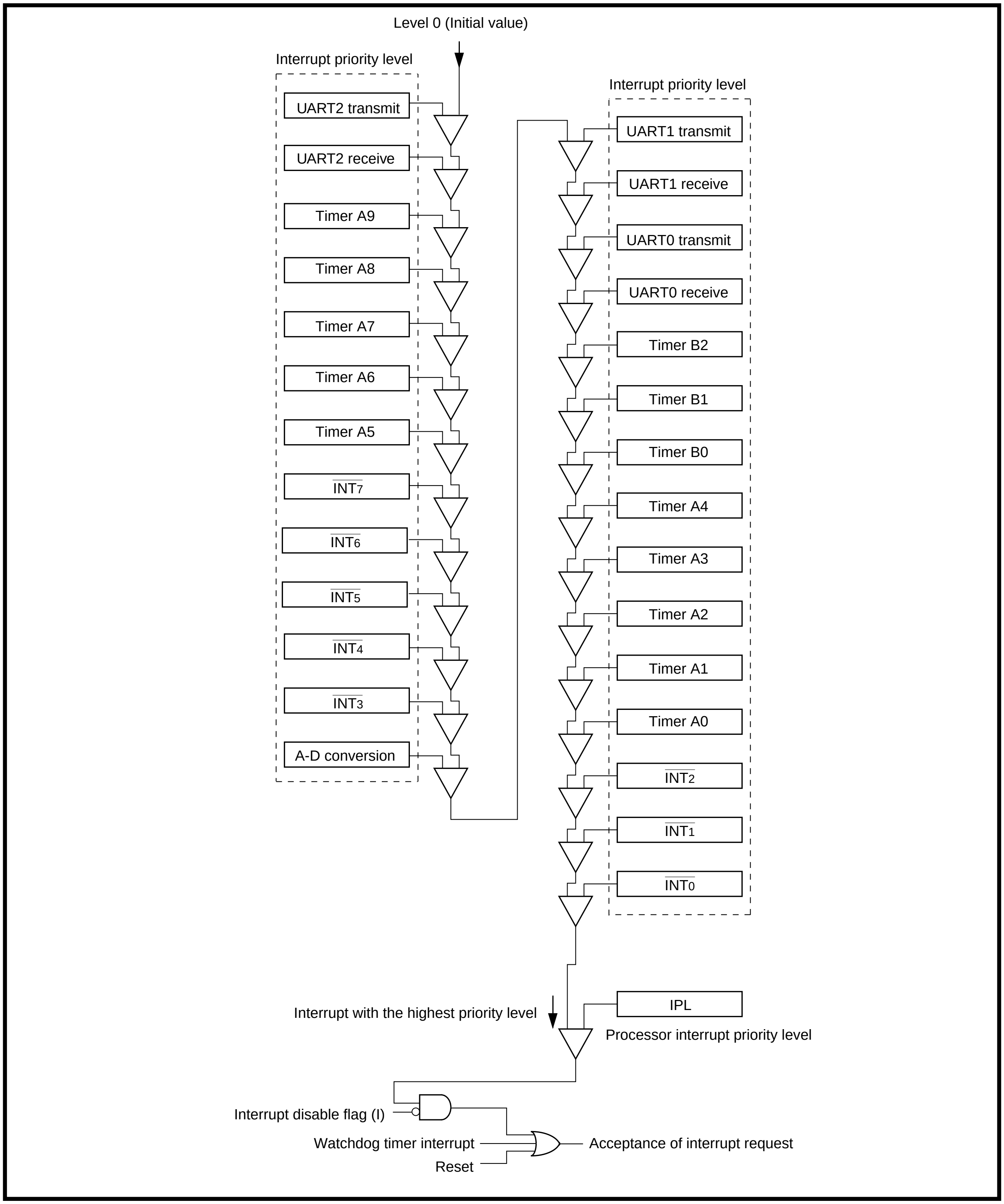


Fig. 6.5.1 Interrupt priority level detection circuit

The following explains the operation of the interrupt priority level detection circuit using Figure 6.5.2.
The interrupt priority level of a requested interrupt (Y in Figure 6.5.2) is compared with the resultant priority level which is sent from the preceding comparator (X in Figure 6.5.2); the interrupt with the higher priority level will be sent to the next comparator (Z in Figure 6.5.2). (The initial value of the comparison level is “0.”)
For an interrupt which is not requested, the comparison is not performed, and the priority level which is sent from the preceding comparator is sent to the next comparator as it is. When the two priority levels are found the same, as a resultant of the comparison, the priority level which is sent from the preceding comparator will be sent to the next comparator. Accordingly, when the same priority level is set to multiple interrupts by software, their interrupt priority levels are handled as follows:

UART2 transmit > UART2 receive > Timer A9 > Timer A8 > Timer A7 > Timer A6 > Timer A5 > $\overline{INT_7}$ > $\overline{INT_6}$ > $\overline{INT_5}$ > $\overline{INT_4}$ > $\overline{INT_3}$ > A-D conversion > UART1 transmit > UART1 receive > UART0 transmit > UART0 receive > Timer B2 > Timer B1 > Timer B0 > Timer A4 > Timer A3 > Timer A2 > Timer A1 > Timer A0 > $\overline{INT_2}$ > $\overline{INT_1}$ > $\overline{INT_0}$

Among the multiple interrupt requests sampled at the same timing, one request with the highest priority level is detected by the above comparison.

Then, this highest interrupt priority level is compared with the processor interrupt priority level (IPL). When this interrupt priority level is higher than IPL and the interrupt disable flag (I) is “0,” the interrupt request is accepted. An interrupt request which is not accepted here is retained until it is accepted or its interrupt request bit is cleared to “0” by software.
The interrupt priority level is detected when the CPU fetches an op code, which is called the CPU’s op-code fetch cycle. However, when an op-code fetch cycle starts during detection of an interrupt priority, a new interrupt priority detection does not start. (See Figure 6.6.2.) Since the state of the interrupt request bit and interrupt priority levels are latched during the interrupt priority detection, even if they change, the interrupt priority detection is performed for the state just before the change occurs.

The interrupt priority level is detected when the CPU fetches an op code. Therefore, in the following case, no interrupt request is accepted until the CPU fetches the op code of the next instruction after the following operation is completed:
•Execution of an instruction which requires many cycles, such as the **MVN** and **MVP** instructions

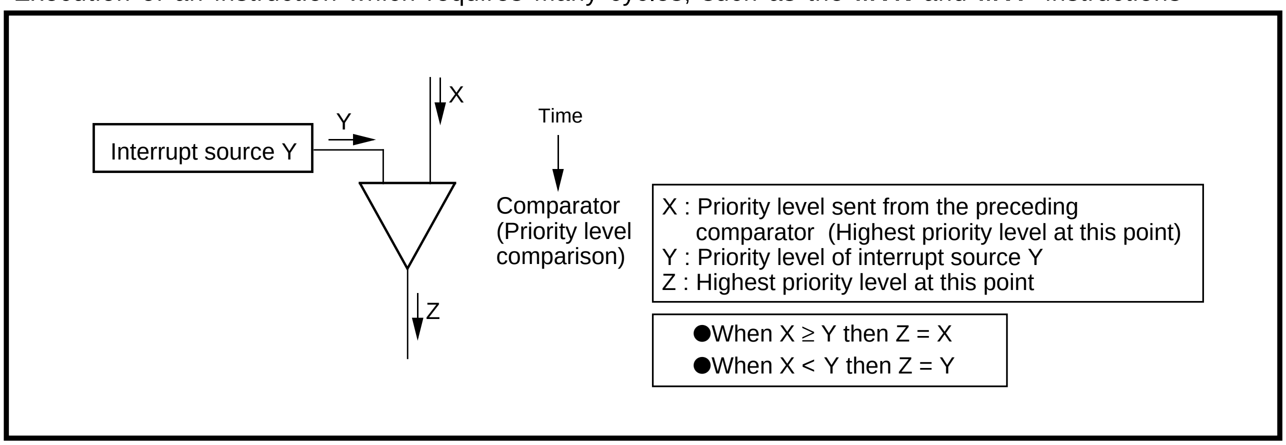


Fig. 6.5.2 Interrupt priority level detection model

6.6 Interrupt priority level detection time

When the interrupt priority level detection time has passed after sampling starts, an interrupt request is accepted. The interrupt priority level detection time can be selected by software. (See Figure 6.6.1.) Usually, select “2 cycles of f_{sys}” as the interrupt priority level detection time.
Figure 6.6.2 shows the interrupt priority level detection time.

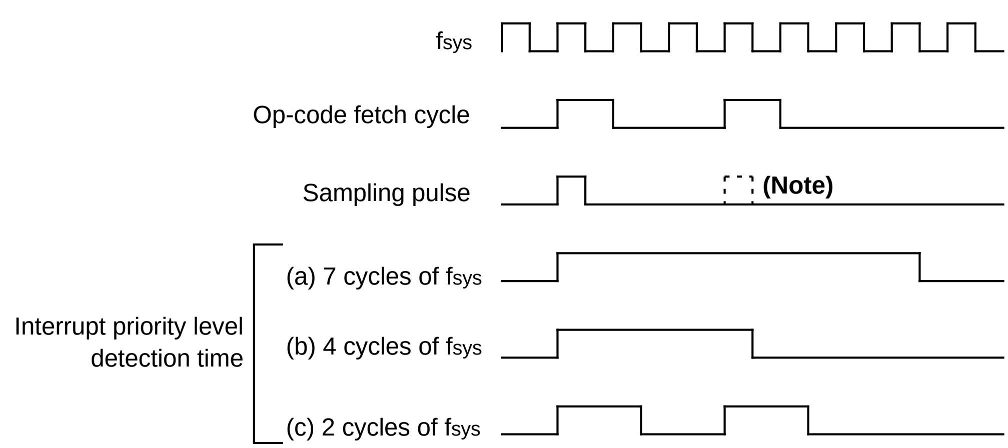
Processor mode register 0 (Address $5E_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
0				X	X	0	0

Bit	Bit name	Function	At reset	R/W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Do not select. 1 0 : Do not select. 1 1 : Do not select.	0	RW
1			0	RW
2	Any of these bits may be either “0” or “1.”		0	RW
3			1	RW
4	Interrupt priority detection time select bits	b5 b4 0 0 : 7 cycles of f_{sys} 0 1 : 4 cycles of f_{sys} 1 0 : 2 cycles of f_{sys} 1 1 : Do not select.	0	RW
5			0	RW
6	Software reset bit	The microcomputer is reset by writing “1” to this bit. The value is “0” at reading.	0	WO
7	Fix this bit to “0.”		0	RW

X : It may be either “0” or “1.”

Fig. 6.6.1 Structure of processor mode register 0



Note: The pulse resides when “2 cycles of f_{sys}” is selected.

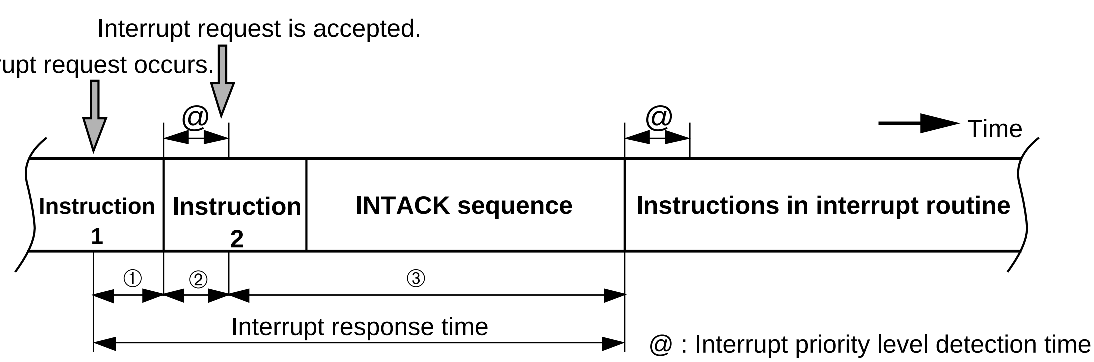
Fig. 6.6.2 Interrupt priority level detection time

6.7 Sequence from acceptance of interrupt request until execution of interrupt routine

The sequence from acceptance of an interrupt request until execution of the interrupt routine is described below.
When an interrupt request is accepted, the interrupt request bit of the accepted interrupt is cleared to "0." And then, the interrupt processing starts from the cycle just after completion of the instruction execution which was executed at acceptance of the interrupt request. Figure 6.7.1 shows the sequence from occurrence of an interrupt request until execution of the interrupt routine. After execution of an instruction at acceptance of the interrupt request is completed, an INTACK (Interrupt Acknowledge) sequence is executed, and a branch is made to the start address of the interrupt routine allocated in addresses 0_{16} to $FFFF_{16}$.
In the INTACK sequence, the following are automatically performed in ascending sequence from ① to ⑥.

① The contents of the program bank register (PG) just before performing the INTACK sequence are pushed onto stack.
② The contents of the program counter (PC) just before performing the INTACK sequence are pushed onto stack.
③ The contents of the processor status register (PS) just before performing the INTACK sequence is pushed onto stack.
④ The interrupt disable flag (I) is set to "1."
⑤ The interrupt priority level of the accepted interrupt is set into the processor interrupt priority level (IPL).
⑥ The contents of the program bank register (PG) are cleared to "00_{16}," and the contents of the interrupt vector address are set into the program counter (PC).

Performing the INTACK sequence requires at least 15 cycles of f_{sys}. Figure 6.7.2 shows the INTACK sequence timing. After the INTACK sequence is completed, the instruction execution starts from the start address of the interrupt routine.

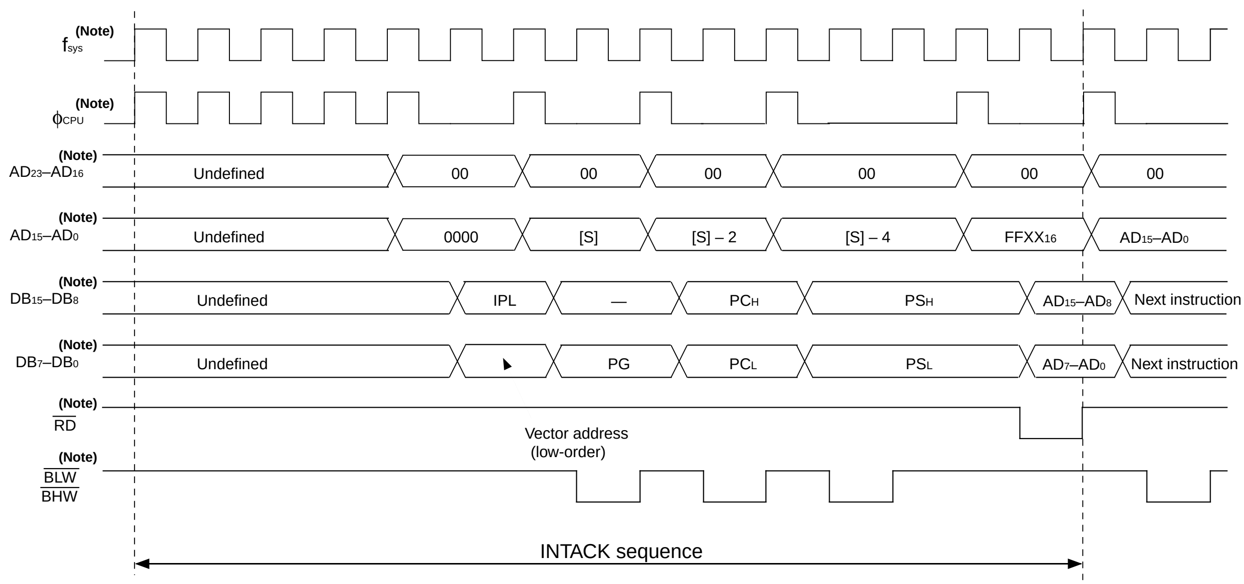


① Time from occurrence of an interrupt request until comparison of an instruction execution which is in progress at that time.
② Time from execution start of an instruction next to ① until completion of execution of the instruction which was in progress at detection completed.
③ Time required to perform the INTACK sequence (15 cycles of ϕ at minimum)

Fig. 6.7.1 Sequence from occurrence of interrupt request until execution of interrupt routine

6.7 Sequence from acceptance of interrupt request until execution of interrupt routine

● When stack pointer (S)'s contents are even at acceptance of an interrupt request with bus cycle = 2 φ



[S]: Contents of stack pointer (S)
$FFXX_{16}$: Vector address
f_{sys}, $φ_{CPU}$: Internal clock (See Figure 4.3.1.)
AD_{23}–AD_0: Internal address bus
DB_{15}–DB_0: Internal data bus

Note: These are internal signals and are not output to the external.

Fig. 6.7.2 INTACK sequence timing (at minimum)

6.7.1 Change in IPL at acceptance of interrupt request

When an interrupt request is accepted, the processor interrupt priority level (IPL) is replaced with the interrupt priority level of the accepted interrupt. This results in easy control of the processing for multiple interrupts. (Refer to section **"6.9 Multiple interrupts."**)

At acceptance of a watchdog timer interrupt request, a zero division request, or address matching detection interrupt request or at reset, a value in Table 6.7.1 is set into the IPL.

Table 6.7.1 Change in IPL at acceptance of interrupt request

Interrupts	Change in IPL
Reset	Level 0 ("000_2") is set.
Watchdog timer	Level 7 ("111_2") is set.
Zero division	Not changed.
Address matching detection	Not changed.
Other interrupts	Accepted interrupt's priority level is set.

6.7.2 Push operation for registers

The push operation for registers performed in the INTACK sequence depends on whether the contents of the stack pointer (S) at acceptance of an interrupt request are even or odd.
When the contents of the stack pointer (S) are even, the contents of the program counter (PC) and the processor status register (PS) are simultaneously pushed in a unit of 16 bits. When the contents of the stack pointer (S) are odd, each of PC and PS is pushed in a unit of 8 bits. Figure 6.7.3 shows the push operation for registers.
In the INTACK sequence, only the contents of the program bank register (PG), program counter (PC), and processor status register (PS) are pushed onto the stack area. Other necessary registers must be pushed by software at the start of the interrupt routine.
By using the **PSH** instruction, all CPU registers, except the stack pointer (S), can be pushed with 1 instruction.

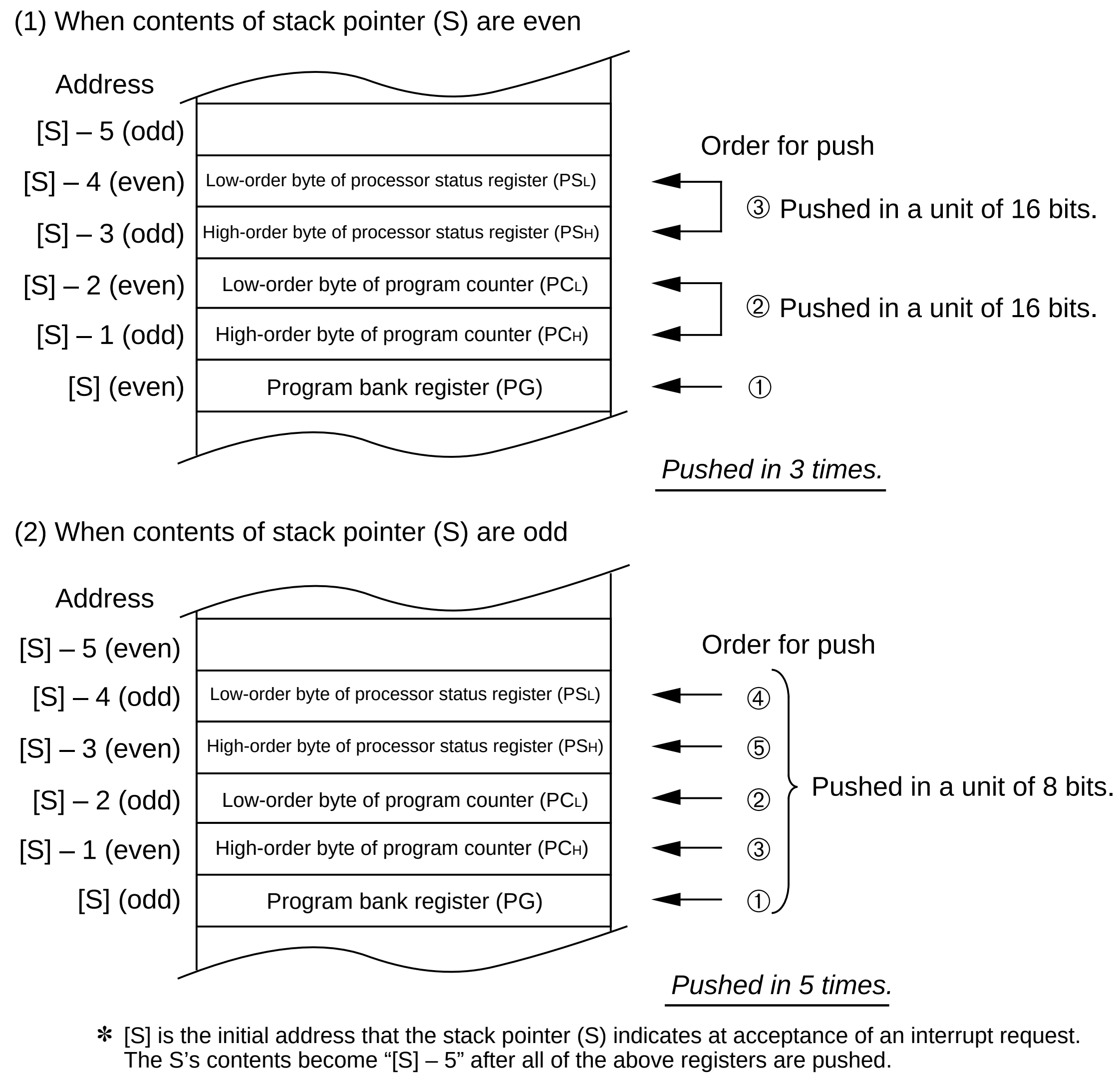


Fig. 6.7.3 Push operation for registers

6.8 Return from interrupt routine

When the **RTI** instruction is executed at the end of the interrupt routine, the contents of the program bank register (PG), program counter (PC), and processor status register (PS) which were pushed onto the stack area just before the INTACK sequence are automatically pulled. After this, the control returns to the original routine. And then, the suspended processing, which was in progress before acceptance of the interrupt request, is resumed.
Before the **RTI** instruction is executed, registers which were pushed by software in the interrupt routine must be pulled in the same data length and register length as those in pushing, using the **PUL** instruction, etc.

6.9 Multiple interrupts

Just after a branch is made to an interrupt routine, the following occur:

- Interrupt disable flag (I) = “1” (Interrupts are disabled.)
- Interrupt request bit of accepted interrupt = “0”
- Processor interrupt priority level (IPL) = Interrupt priority level of accepted interrupt

Accordingly, as long as the IPL remains unchanged, an interrupt request, whose priority level is higher than that of the interrupt which is in progress, can be accepted by clearing the interrupt disable flag (I) to “0” in an interrupt routine. In this way, multiple interrupts are processed.
Figure 6.9.1 shows the processing for multiple interrupts.
An interrupt request which has not been accepted because its priority level is lower is retained. When the **RTI** instruction is executed, the interrupt priority level of the routine which was in progress just before acceptance of an interrupt request is pulled into the IPL. Therefore, if the following relationship is satisfied when interrupt priority level detection is performed next, the retained interrupt request will be accepted.

Retained interrupt request’s priority level > Processor interrupt priority level (IPL)

Note: When any of the following interrupt requests is generated while an interrupt routine is in progress, this interrupt request is accepted at once: zero division, watchdog timer, and address matching detection.

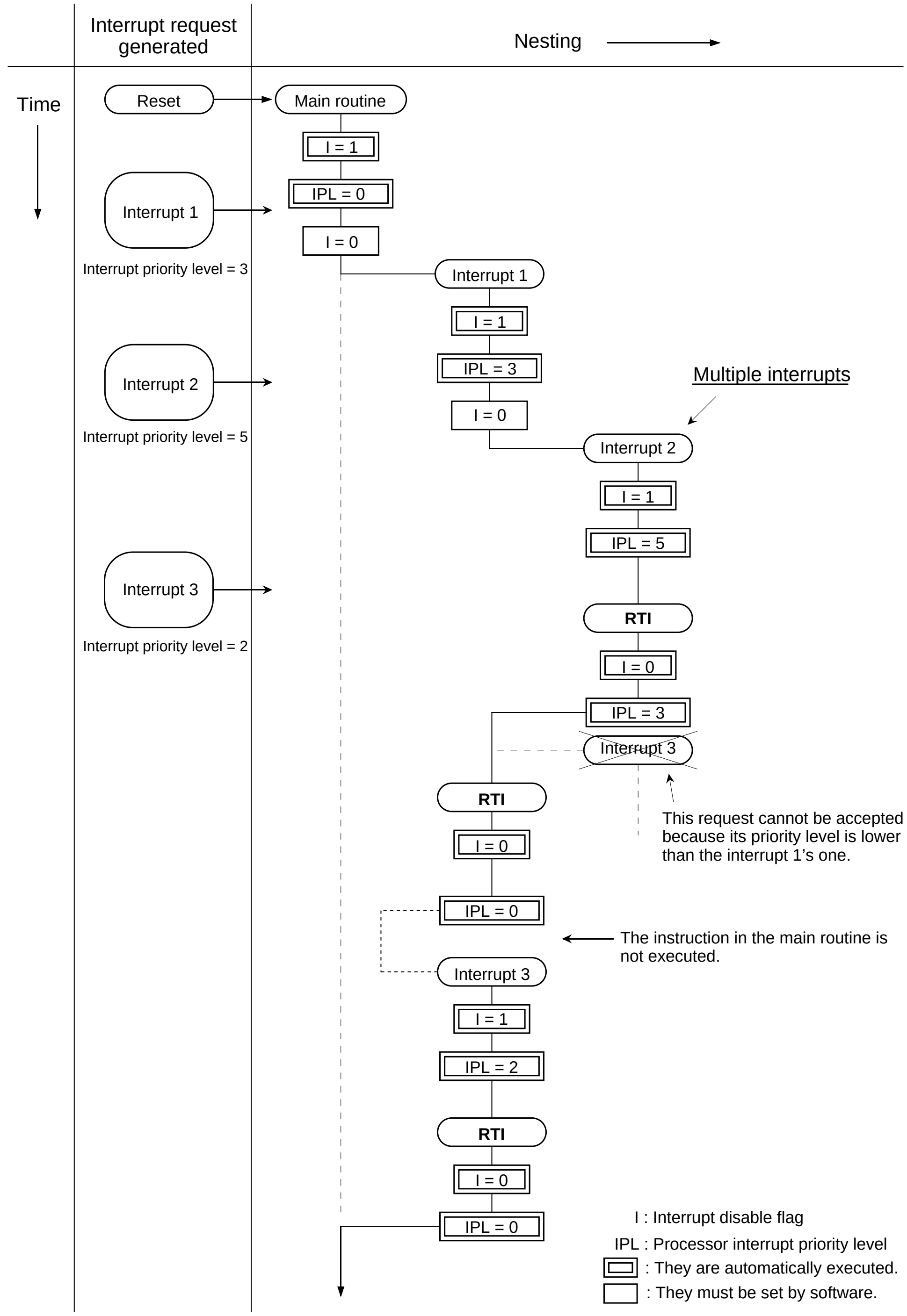


Fig. 6.9.1 Processing for multiple interrupts

6.10 External interrupts

The external interrupts consist of $\overline{INT_i}$ interrupts.

6.10.1 $\overline{INT_i}$ interrupt

An $\overline{INT_i}$ (i = 0 to 7) interrupt request occurs by an input signal to pin $\overline{INT_i}$. Table 6.10.1 lists the occurrence factor of the $\overline{INT_i}$ interrupt request.

When using any of pins $P5_1/\overline{INT_1}$, $P5_2/\overline{INT_2}$, $P5_3/\overline{INT_3}$, $P5_5/\overline{INT_5}$, $P5_6/\overline{INT_6}$, $P5_7/\overline{INT_7}$ as an input pin of the external interrupt, be sure to clear the port P5 direction register's bit. (See Figure 6.10.2.)

When using pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$ as an input pin of an external interrupt (pin $\overline{INT_0}$), be sure to use port pins $P4_0$ to $P4_7$ in the input mode. (Refer to section **"5.2.3 Pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$."**)

When using pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$ as an input pin of an external interrupt (pin $\overline{INT_4}$), be sure to use port pins $P6_0$ to $P6_7$ in the input mode. (Refer to section **"5.2.4 Pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$."**)

The signal input to pin $\overline{INT_i}$ requires "H" or "L" level width of 250 ns or more, independent of $f(X_{IN})$.

By reading out the $\overline{INT_i}$ read bit (See Figure 6.10.1.), the state of pin $\overline{INT_i}$ can be read out.

Note: Selection of the interrupt occurrence factor requires the following conditions:

- when an input signal's falling edge or "L" level is selected, be sure that "L" level width ≥ 250 ns.
- when an input signal's rising edge or "H" level is selected, be sure that "H" level width ≥ 250 ns.

Table 6.10.1 Occurrence factor of $\overline{INT_i}$ interrupt request

	Level sense/Edge sense select bit (bit 5 at addresses $7D_{16}$ to $7F_{16}$, $6E_{16}$, $6F_{16}$, FD_{16} to FF_{16})	Polarity select bit (bit 4 at addresses $7D_{16}$ to $7F_{16}$, $6E_{16}$, $6F_{16}$, FD_{16} to FF_{16})	Occurrence factor of interrupt request (An interrupt request occurs when the input signal of pin $\overline{INT_i}$ is as follows.)
$\overline{INT_0}$ to $\overline{INT_7}$	0	0	Falling edge (Edge sense)
	0	1	Rising edge (Edge sense)
	1	0	"H" level (Level sense)
	1	1	"L" level (Level sense)

The $\overline{INT_i}$ interrupt request occurs by detecting the state of pin $\overline{INT_i}$ all the time. Therefore, when the user does not use an $\overline{INT_i}$ interrupt, be sure to set the $\overline{INT_i}$ interrupt's priority level to 0.

External interrupt input read register (Address 95_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	$\overline{INT_0}$ read out bit	The input level at the corresponding pin is read out. 0 : "L" level 1 : "H" level	Undefined	RO
1	$\overline{INT_1}$ read out bit		Undefined	RO
2	$\overline{INT_2}$ read out bit		Undefined	RO
3	$\overline{INT_3}$ read out bit		Undefined	RO
4	$\overline{INT_4}$ read out bit		Undefined	RO
5	$\overline{INT_5}$ read out bit		Undefined	RO
6	$\overline{INT_6}$ read out bit		Undefined	RO
7	$\overline{INT_7}$ read out bit		Undefined	RO

Fig. 6.10.1 Structure of external interrupt input read register

Port P5 direction register (Address D_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin	Function	At reset	R/W
0	Nothing is assigned.		Undefined	—
1	Pin $\overline{INT_1}$	0 : Input mode 1 : Output mode When using this pin as an external interrupt's input pin, be sure to clear the corresponding bit to "0."	0	RW
2	Pin $\overline{INT_2}$ (RTP_{TRG1})		0	RW
3	Pin $\overline{INT_3}$ (RTP_{TRG0})		0	RW
4	Nothing is assigned.		Undefined	—
5	Pin $\overline{INT_5}$ ($TB0_{IN}$/IDW)	0 : Input mode 1 : Output mode When using this pin as an external interrupt's input pin, be sure to clear the corresponding bit to "0."	0	RW
6	Pin $\overline{INT_6}$ ($TB1_{IN}$/IDV)		0	RW
7	Pin $\overline{INT_7}$ ($TB2_{IN}$/IDU)		0	RW

Note: ( ) shows the I/O pins of other internal peripheral devices which are multiplexed.

Fig. 6.10.2 Relationship between port P5 direction register and external interrupt's input pins

6.10.2 Functions of $\overline{INT_i}$ interrupt request bit

Figure 6.10.3 shows an $\overline{INT_i}$ interrupt request.

(1) Functions when edge sense is selected

In this case, the interrupt request bit has the same function as that of an internal interrupt. That is, when an interrupt request occurs, the interrupt request bit is set to "1" and retains this state until the interrupt request is accepted. When this bit is cleared to "0" by software, the interrupt request is cancelled; when this bit is set to "1" by software, the interrupt request can occur.

(2) Functions when level sense is selected

In this case, the interrupt request bit is ignored.

$\overline{INT_i}$ interrupt requests continuously occur while the level at pin $\overline{INT_i}$ is the valid level*1; when the level at pin $\overline{INT_i}$ changes from the valid level to the invalid level*2 before the corresponding $\overline{INT_i}$ interrupt request is accepted, this interrupt request is not retained. (See Figure 6.10.4.)

Valid level*1: This means the level selected by the polarity select bit (bit 4 at addresses $7D_{16}$ to $7F_{16}$, $6E_{16}$, $6F_{16}$, FD_{16} to FF_{16})

Invalid level*2: This means the reversed level of "valid level"

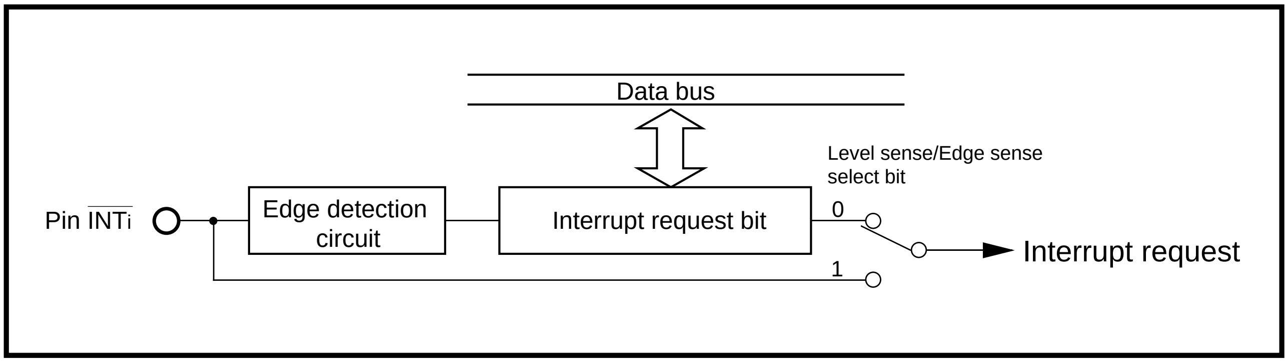


Fig. 6.10.3 $\overline{INT_i}$ Interrupt request

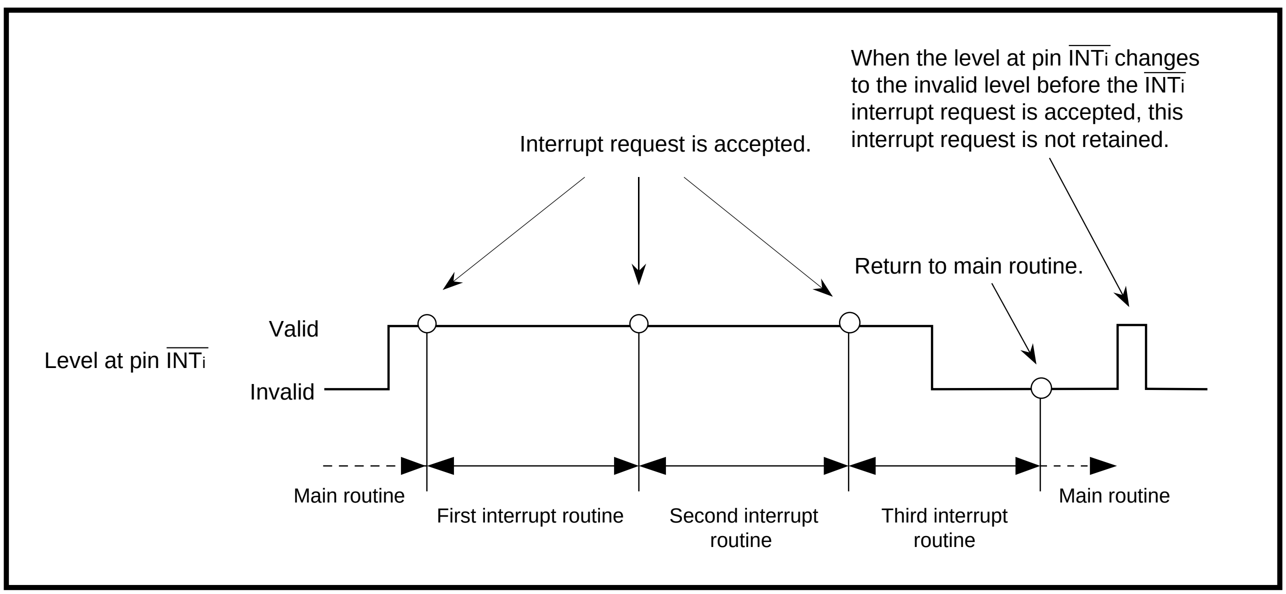


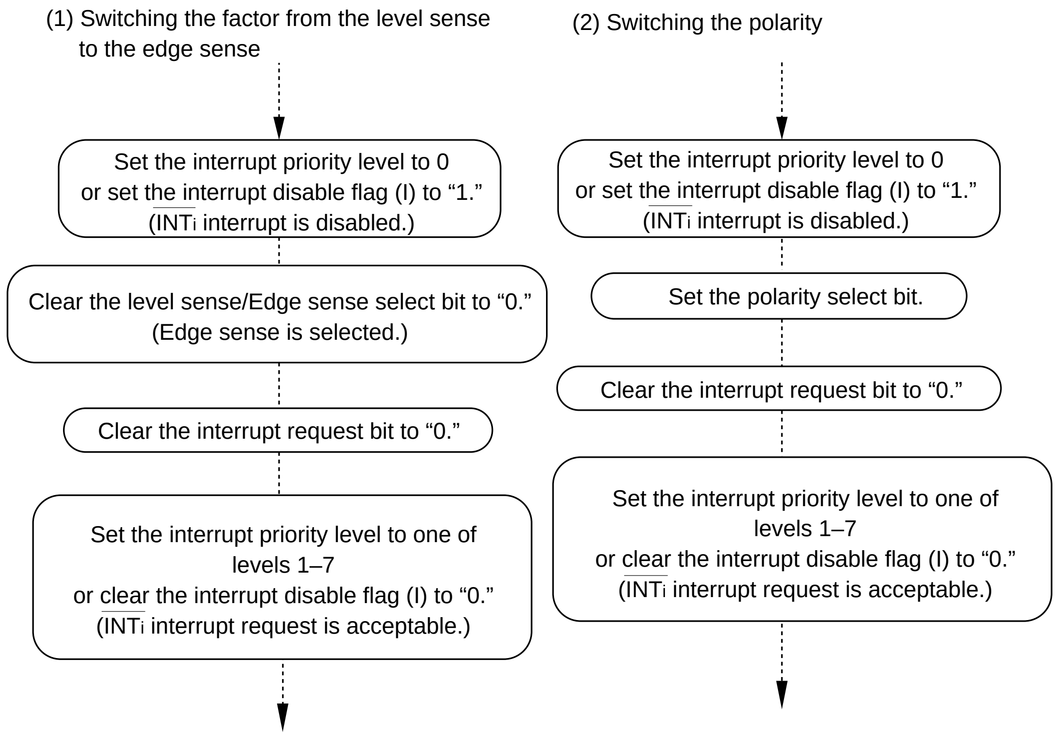
Fig. 6.10.4 Occurrence of $\overline{INT_i}$ interrupt request when level sense is selected

6.10.3 Switching of $\overline{INT_i}$ to interrupt request occurrence factor

When the $\overline{INT_i}$ interrupt request occurrence factor is switched in one of the following ways, there is a possibility that the corresponding interrupt request bit is set to "1":

- Switching the factor from the level sense to the edge sense
- Switching the polarity

Therefore, after this switching, make sure to clear the corresponding interrupt request bit to "0." Figure 6.10.5 shows an example of the switching procedure for the $\overline{INT_i}$ interrupt request's occurrence factor.



Note: The above settings must be done separately.
Multiple settings must not be done at the same time, in other words, they must not be done only by 1 instruction.

Fig. 6.10.5 Example of switching procedure for $\overline{INT_i}$ interrupt request's occurrence factor

[Precautions for interrupts]

1. In order to change the interrupt priority level select bits (bits 0 to 2 at addresses $6E_{16}$ to $7F_{16}$, $F1_{16}$, $F2_{16}$, $F5_{16}$ to $F9_{16}$, FD_{16} to FF_{16}), 2 to 7 cycles of f_{sys} are required after execution of a write instruction until change of the interrupt priority level. Therefore, when the interrupt priority level of a certain interrupt source is repeatedly changed in a very short time, which consists of a few instructions, it is necessary to reserve the time required for the change by software. Figure 6.10.6 shows a program example to reserve the time required for the change. Note that the time required for the change depends on the contents of the interrupt priority detection time select bits (bits 4 and 5 at address $5E_{16}$). Table 6.10.2 lists the correspondence between the number of instructions inserted in Figure 6.10.6 and the interrupt priority detection time select bits.

```
:
MOVMB 00XXH, #0XH  ; Write instruction for the interrupt priority level select bits
NOP                ; Inserted NOP instruction (Note)
NOP                ;
NOP                ;
MOVMB 00XXH, #0XH  ; Write instruction for the interrupt priority level select bits
:
```

Note: Except a write instruction for address XX_{16}, any instruction which has the same cycles as the **NOP** instruction can also be inserted, instead of the **NOP** instruction. For the number of inserted **NOP** instructions, see Table 6.10.2.

XX: any of 6E to 7F, F1, F2, F5 to F9, and FD to FF

Fig. 6.10.6 Program example to reserve time required for change of interrupt priority level

Table 6.10.2 Correspondence between number of instructions to be inserted in Figure 6.10.6 and interrupt priority detection time select bits

Interrupt priority detection time select bits **(Note)**		Interrupt priority level detection time	Number of inserted **NOP** instructions
b5	b4		
0	0	7 cycles of f_{sys}	7 or more
0	1	4 cycles of f_{sys}	4 or more
1	0	2 cycles of f_{sys}	2 or more
1	1	Do not select.	

Note: We recommend [b5 = “1”, b4 = “0”].

2. When using pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$ as an input pin of an external interrupt (pin $\overline{INT_0}$), be sure to use port pins $P4_0$ to $P4_7$ in the input mode. (Refer to section **“5.2.3 Pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$.”**)

3. When using pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$ as an input pin of an external interrupt (pin $\overline{INT_4}$), be sure to use port pins $P6_0$ to $P6_7$ in the input mode. (Refer to section **“5.2.4 Pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$.”**)

CHAPTER 7
TIMER A

7.1 Overview

Timer A consists of ten counters, Timers A0 to A9, each equipped with a 16-bit reload function. Timers A0 to A9 operate independently of one other.
Timer Ai (i = 0 to 9) has four operating modes listed below. Except for the event counter mode, timer Ai has the same functions.
Table 7.1.1 lists the functions of timer Ai.

(1) Timer mode

In this mode, the timer counts an internally generated count source. Following functions can be used in this mode:
- Gate function
- Pulse output function

(2) Event counter mode

In this mode, the timer counts an external signal. Following functions can be used in this mode:
- Pulse output function
- Two-phase pulse signal processing function (Timers A2 to A4, A7 to A9)

(3) One-shot pulse mode

In this mode, the timer outputs a pulse which has an arbitrary width once.

(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses which have an arbitrary width in succession. In this mode, the timer serves as one of the following pulse width modulators:
- 16-bit pulse width modulator
- 8-bit pulse width modulator

Table 7.1.1 Functions of timer Ai

<table>
<tr><th colspan="2" rowspan="2">Functions of timers</th><th colspan="10">Timer Ai (i = 0 to 9)</th></tr>
<tr><th>TA0</th><th>TA1</th><th>TA2</th><th>TA3</th><th>TA4</th><th>TA5</th><th>TA6</th><th>TA7</th><th>TA8</th><th>TA9</th></tr>
<tr><td rowspan="3">Timer mode</td><td>Timer</td><td colspan="2">√</td><td colspan="3">√</td><td colspan="2">√</td><td colspan="3">√</td></tr>
<tr><td>Gate function</td><td colspan="2">√</td><td colspan="3">√</td><td colspan="2">√</td><td colspan="3">√</td></tr>
<tr><td>Pulse output function</td><td colspan="2">√</td><td colspan="3">√</td><td colspan="2">√</td><td colspan="3">√</td></tr>
<tr><td rowspan="2">Event counter mode</td><td>Pulse output function</td><td colspan="2">√</td><td colspan="3">√</td><td colspan="2">√</td><td colspan="3">√</td></tr>
<tr><td>Two-phase pulse signal processing function</td><td colspan="2">—</td><td colspan="3">√ (Note)</td><td colspan="2">—</td><td colspan="3">√ (Note)</td></tr>
<tr><td colspan="2">One-shot pulse mode</td><td colspan="2">√</td><td colspan="3">√</td><td colspan="2">√</td><td colspan="3">√</td></tr>
<tr><td colspan="2">Pulse width modulation (PWM) mode</td><td colspan="2">√</td><td colspan="3">√</td><td colspan="2">√</td><td colspan="3">√</td></tr>
</table>

Note: Normal processing for TA2, TA3, TA7, TA8; and quadruple processing for TA4, TA9

7.2 Block description

Figure 7.2.1 shows the block diagram of timer Ai (i = 0 to 9). Explanation of registers relevant to timer A is described below.

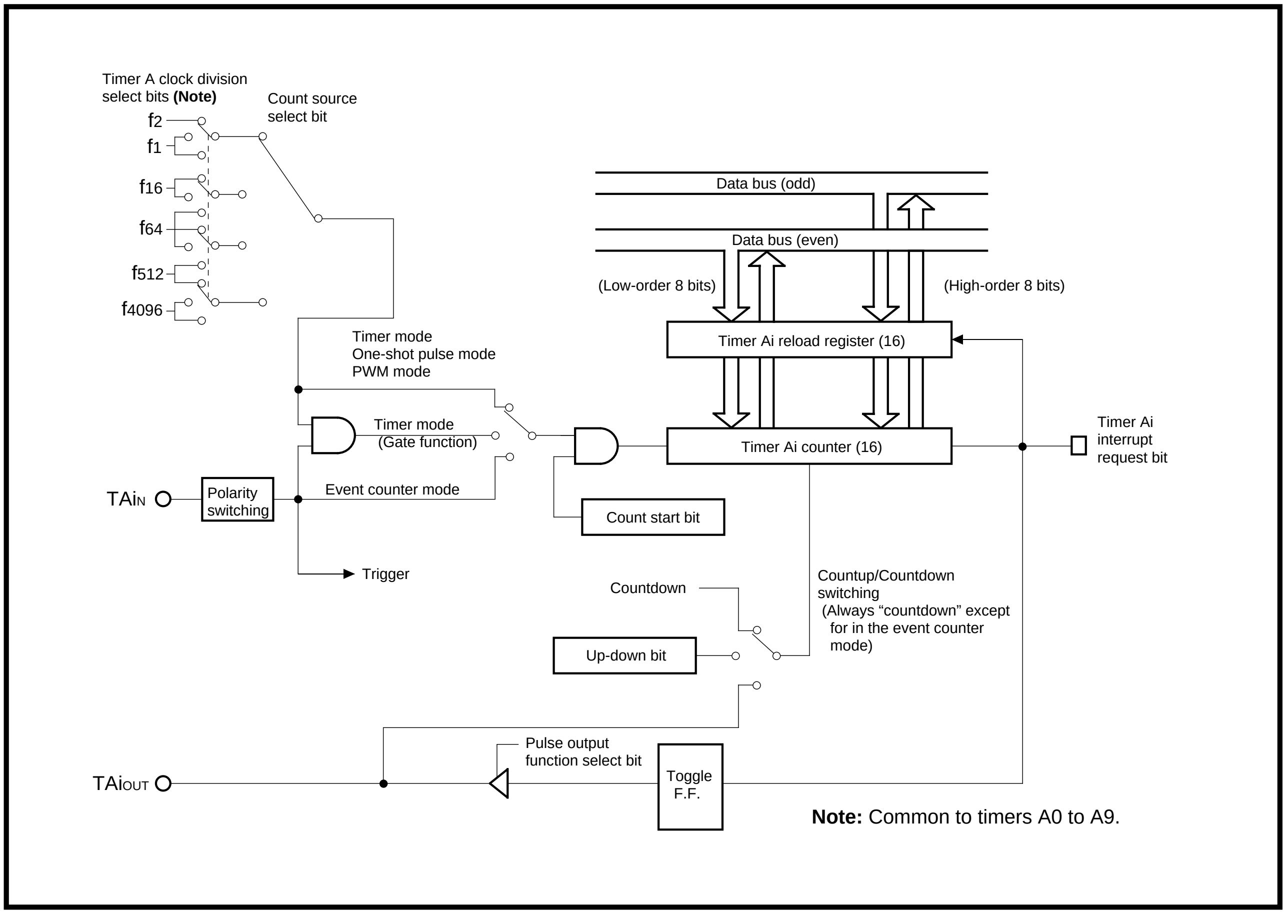


Fig. 7.2.1 Block diagram of timer Ai (i = 0 to 9)

7.2 Block description

7.2.1 Counter and Reload register (timer Ai register)

Each of timer Ai counter and reload register consists of 16 bits.
Countdown in the counter is performed each time the count source is input. In the event counter mode, it can also function as an up-counter.
The reload register is used to store the initial value of the counter. When a counter underflow or overflow occurs, the reload register's contents are reloaded into the counter.
A value is set to the counter and reload register by writing the value to the timer Ai register. Table 7.2.1 lists the memory assignment of the timer Ai register.
The value written into the timer Ai register while counting is not in progress is set to the counter and reload register. The value written into the timer Ai register while counting is in progress is set only to the reload register. In this case, the reload register's updated contents are transferred to the counter at the next reload time. The value obtained when reading out the timer Ai register varies according to the operating mode. Table 7.2.2 lists reading from and writing to the timer Ai register.

Table 7.2.1 Memory assignment of timer Ai register

Timer Ai register	High-order byte	Low-order byte
Timer A0 register	Address 47_{16}	Address 46_{16}
Timer A1 register	Address 49_{16}	Address 48_{16}
Timer A2 register	Address $4B_{16}$	Address $4A_{16}$
Timer A3 register	Address $4D_{16}$	Address $4C_{16}$
Timer A4 register	Address $4F_{16}$	Address $4E_{16}$
Timer A5 register	Address $C7_{16}$	Address $C6_{16}$
Timer A6 register	Address $C9_{16}$	Address $C8_{16}$
Timer A7 register	Address CB_{16}	Address CA_{16}
Timer A8 register	Address CD_{16}	Address CC_{16}
Timer A9 register	Address CF_{16}	Address CE_{16}

Note: At reset, the contents of the timer Ai register are undefined.

Table 7.2.2 Reading from and writing to timer Ai register

Operating mode	Read	Write
Timer mode	Counter value is read out. (**Note 1**)	<While counting> Written only to reload register. <While not counting> Written to both of the counter and reload register.
Event counter mode		
One-shot pulse mode	Undefined value is read out.	
Pulse width modulation (PWM) mode		

Notes 1: Also refer to sections **"[Precautions for timer mode]"** and **"[Precautions for event counter mode]."**
2: When reading from and writing to the timer Ai register, perform it in a unit of 16 bits.

7.2.2 Timer A clock division select register

In the timer mode, one-shot pulse mode, and pulse width modulation (PWM) mode, the count source select bits (bits 6 and 7 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16}), and timer A clock division select bits (bits 0 and 1 at address 45_{16}) select the count source. Figure 7.2.2 shows the structure of the timer A clock division select register. Table 7.2.3 lists the count source (in the timer mode, one-shot pulse mode, and pulse width modulation (PWM) mode).

Timer A clock division select register (Address 45_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Timer A clock division select bits	See Table 7.2.3.	0	RW
1			0	RW
7 to 2	The value is “0” at reading.		0	–

Fig. 7.2.2 Structure of timer A clock division select register

Table 7.2.3 Count source (in timer mode, one-shot pulse mode, and pulse width modulation (PWM) mode)

Count source select bits (bits 6 and 7 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16})	Timer A clock division select bits (bits 0 and 1 at address 45_{16})			
	00	01	10	11
00	f_2	f_1	f_1	Do not select.
01	f_{16}	f_{16}	f_{64}	
10	f_{64}	f_{64}	f_{512}	
11	f_{512}	f_{4096}	f_{4096}	

7.2 Block description

7.2.3 Count start register

This register is used to start and stop counting. One bit of this registar corresponds to one timer. (This is the one-to-one relationship.) Figure 7.2.3 shows the structures of the count start registers 0 and 1.

Count start register 0 (Address 40_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Timer A0 count start bit	0 : Stop counting 1 : Start counting	0	RW
1	Timer A1 count start bit		0	RW
2	Timer A2 count start bit		0	RW
3	Timer A3 count start bit		0	RW
4	Timer A4 count start bit		0	RW
5	Timer B0 count start bit		0	RW
6	Timer B1 count start bit		0	RW
7	Timer B2 count start bit		0	RW

Count start register 1 (Address 41_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Timer A5 count start bit	0 : Stop counting 1 : Start counting	0	RW
1	Timer A6 count start bit		0	RW
2	Timer A7 count start bit		0	RW
3	Timer A8 count start bit		0	RW
4	Timer A9 count start bit		0	RW
7 to 5	Nothing is assigned.		Undefined	–

Fig. 7.2.3 Structures of count start registers 0 and 1

7.2.4 Timer Ai mode register

Figure 7.2.4 shows the structure of the timer Ai mode register. The operating mode select bits are used to select the operating mode of timer Ai. Bits 2 to 7 have different functions according to the operating mode. These bits are described in the paragraph of each operating mode.

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
(i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0

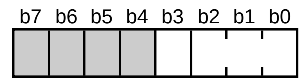
Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits (Note)	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot pulse mode 1 1 : Pulse width modulation (PWM) mode	0	RW
1			0	RW
2	These bits have different functions according to the operating mode.		0	RW
3			0	RW
4			0	RW
5			0	RW
6			0	RW
7			0	RW

Fig. 7.2.4 Structure of timer Ai mode register

7.2.5 Timer Ai interrupt control register

Figure 7.2.5 shows the structure of the timer Ai interrupt control register. For details about interrupts, refer to **"CHAPTER 6. INTERRUPTS."**

Timer Ai interrupt control register (i = 0 to 4) (Addresses 75_{16} to 79_{16})
(i = 5 to 9) (Addresses $F5_{16}$ to $F9_{16}$)



Bit	Bit name	Function	At reset	R/W
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt requested 1 : Interrupt requested	0	RW **(Note)**
7 to 4	Nothing is assigned.		Undefined	—

Note: When writing to this bit, use the **MOVM (MOVMB)** instruction or **STA (STAB, STAD)** instruction.

Fig. 7.2.5 Structure of timer Ai interrupt control register

(1) Interrupt priority level select bits (bits 2 to 0)

These bits are used to select a timer Ai interrupt's priority level. When using timer Ai interrupts, select the priority level from levels 1 through 7. When a timer Ai interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL), so that the requested interrupt is enabled only when its priority level is higher than the IPL. (However, this applies when the interrupt disable flag (I) = "0.") To disable timer Ai interrupts, set these bits to "000_2" (level 0).

(2) Interrupt request bit (bit 3)

This bit is set to "1" when a timer Ai interrupt request occurs. This bit is automatically cleared to "0" when the timer Ai interrupt request is accepted. This bit can be set to "1" or cleared to "0" by software.

7.2.6 Port P2, port P4 and port P6 direction registers

The I/O pins of timers A0 to A3 are multiplexed with port P6 pins, and the I/O pins of timers A4 and A9 are multiplexed with port P2 pins, and the I/O pins of timers A5 to A8 are multiplexed with port P4 pins. When using these pins as timer Ai (i = 0 to 9)'s input pins, clear the corresponding bits of the port P6, port P2, and port P4 direction registers to "0" in order to set these port pins for the input mode. When used as timer Ai's output pins, these pins are forcibly set to the output pins of timer Ai regardless of the direction registers' contents. Figure 7.2.6 shows the relationship between the port P6 direction register and the timer Ai's I/O pins, Figure 7.2.7 shows the relationship between port P2 and port P4 direction registers and timer Ai's I/O pins.

Note that each bit of the port P4 direction register becomes "0" by an input of a falling edge to pin $\overline{P4OUT_{CUT}}$. (Refer to section **"5.2.3 Pin $\overline{P4OUT_{CUT}}$/$\overline{INT_0}$."**) When switching the output pins of timers A5 to A8 to the port output pins, the following procedure is required.

① Return the input level at pin $\overline{P4OUT_{CUT}}$ to "H."
② Write data to the port P4 register's bit corresponding to the port P4 pin, where data is to be output.
③ Set "1" to the port P4 direction register's bit corresponding to the above P4 register's bit; therefore, this bit enters the output mode.

When the input level at pin $\overline{P4OUT_{CUT}}$ = "L," no bit of the port P4 direction register can be set to "1."

Similarly, each bit of the port P6 direction register becomes "0" by an input of a falling edge to pin $\overline{P6OUT_{CUT}}$. (Refer to section **"5.2.4 Pin $\overline{P6OUT_{CUT}}$/$\overline{INT_4}$."**) When switching the output pins of timers A0 to A3 to the port output pins, the following procedure is required.

① Return the input level at pin $\overline{P6OUT_{CUT}}$ to "H."
② Write data to the port P6 register's bit corresponding to the port P6 pin, where data is to be output.
③ Set "1" to the port P6 direction register's bit corresponding to the above P6 register's bit; therefore, this bit enters the output mode.

When the input level at pin $\overline{P6OUT_{CUT}}$ = "L," no bit of the port P6 direction register can be set to "1."

Port P6 direction register (Address 10_{16})   b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin	Functions	At reset	R/W
0	Pin $TA0_{OUT}$ (Pin $\overline{W}$/$RTP0_0$)	0 : Input mode 1 : Output mode When using this pin as timer Ai's input pin, be sure to clear the corresponding bit to "0."	0	RW
1	Pin $TA0_{IN}$ (Pin $\overline{V}$/$RTP0_1$)		0	RW
2	Pin $TA1_{OUT}$ (Pin $\overline{U}$/$RTP0_2$)		0	RW
3	Pin $TA1_{IN}$ (Pin W/$RTP0_3$)		0	RW
4	Pin $TA2_{OUT}$ (Pin V/$RTP1_0$)		0	RW
5	Pin $TA2_{IN}$ (Pin U/$RTP1_1$)		0	RW
6	Pin $TA3_{OUT}$ (Pin $RTP1_2$)		0	RW
7	Pin $TA3_{IN}$ (Pin $RTP1_3$)		0	RW

Notes 1: Each of bits 0 to 7 becomes "0" by an input of the falling edge to pin $\overline{P6OUT_{CUT}}$/$\overline{INT_4}$. (Refer to section **"5.2.4 Pin $\overline{P6OUT_{CUT}}$/$\overline{INT_4}$."**)
2: The pins in ( ) are I/O pins of other internal peripheral devices, which are multiplexed.

Fig. 7.2.6 Relationship between port P6 direction register and timer Ai's I/O pins

Port P2 direction register (Address 8_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin	Functions	At reset	R/W
0	Pin $TA4_{OUT}$	0 : Input mode 1 : Output mode When using this pin as timer Ai's input pin, be sure to clear the corresponding bit to "0."	0	RW
1	Pin $TA4_{IN}$		0	RW
2	Pin $TA9_{OUT}$		0	RW
3	Pin $TA9_{IN}$		0	RW
4	Pin $TB0_{IN}$ **(Note 1)**		0	RW
5	Pin $TB1_{IN}$ **(Note 2)**		0	RW
6	Pin $TB2_{IN}$ **(Note 3)**		0	RW
7	Pin $P2_7$		0	RW

Notes 1: This applies when the $TB0_{IN}$ pin select bit (bit 0 at address AE_{16}) = 1.
2: This applies when the $TB1_{IN}$ pin select bit (bit 1 at address AE_{16}) = 1.
3: This applies when the $TB2_{IN}$ pin select bit (bit 2 at address AE_{16}) = 1.

Port P4 direction register (Address C_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin	Functions	At reset	R/W
0	Pin $TA5_{OUT}$ (Pin $RTP2_0$)	0 : Input mode 1 : Output mode When using this pin as timer Ai's input pin, be sure to clear the corresponding bit to "0."	0	RW
1	Pin $TA5_{IN}$ (Pin $RTP2_1$)		0	RW
2	Pin $TA6_{OUT}$ (Pin $RTP2_2$)		0	RW
3	Pin $TA6_{IN}$ (Pin $RTP2_3$)		0	RW
4	Pin $TA7_{IN}$ (Pin $RTP3_0$)		0	RW
5	Pin $TA7_{IN}$ (Pin $RTP3_1$)		0	RW
6	Pin $TA8_{OUT}$ (Pin $RTP3_2$)		0	RW
7	Pin $TA8_{IN}$ (Pin $RTP3_3$)		0	RW

Notes 1: Each of bits 0 to 7 becomes "0" by an input of the falling edge to pin $\overline{P4OUT_{CUT}/INT_0}$. (Refer to section **"5.2.3 Pin $\overline{P4OUT_{CUT}/INT_0}$."**)
2: The pins in ( ) are I/O pins of other internal peripheral devices, which are multiplexed.

Fig. 7.2.7 Relationship between port P4 and port P2 direction registers and timer Ai's I/O pins

7.3 Timer mode

In this mode, the timer counts an internally generated count source. Table 7.3.1 lists the specifications of the timer mode. Figure 7.3.1 shows the structures of the timer Ai register and timer Ai mode register in the timer mode.

Table 7.3.1 Specifications of timer mode

Item	Specifications
Count source f_i	f_1, f_2, f_{16}, f_{64}, f_{512}, or f_{4096}
Count operation	• Countdown • When a counter underflow occurs, reload register's contents are reloaded, and counting continues.
Division ratio	$\frac{1}{(n + 1)}$ n : Timer Ai register setting value
Count start condition	When count start bit is set to "1."
Count stop condition	When count start bit is cleared to "0."
Interrupt request occurrence timing	When a counter underflow occurs.
TAi_{IN} pin function	Programmable I/O port pin or gate input pin
TAi_{OUT} pin function	Programmable I/O port pin or pulse output pin
Read from timer Ai register	Counter value can be read out.
Write to timer Ai register	● While counting is stopped When a value is written to the timer Ai register, it is written to both reload register and counter. ● While counting is in progress When a value is written to the timer Ai register, it is written to only reload register. (Transferred to the counter at the next reload timing.)

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 — (b8) b0 | b7 — b0

Bit	Function	At reset	R/W
15 to 0	Any value in the range from "0000_{16}" to "$FFFF_{16}$" can be set. Assuming that the set value = n, the counter divides the count source frequency by (n + 1). When reading, the register indicates the counter value.	Undefined	RW

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits	b1 b0 0 0 : Timer mode	0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output (TAi$_{OUT}$ pin functions as a programmable I/O port pin.) 1 : Pulse output (TAi$_{OUT}$ pin functions as a pulse output pin.)	0	RW
3	Gate function select bits	b4 b3 0 0 : } No gate function 0 1 : } (TAi$_{IN}$ pin functions as a programmable I/O port pin.) 1 0 : Gate function (Counter is active only while TAi$_{IN}$ pin's input signal is at "L" level.) 1 1 : Gate function (Counter is active only while TAi$_{IN}$ pin's input signal is at "H" level.)	0	RW
4			0	RW
5	Fix this bit to "0" in timer mode.		0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 7.3.1 Structures of timer Ai register and timer Ai mode register in timer mode

7.3.1 Setting for timer mode

Figure 7.3.2 shows an initial setting example for registers related to the timer mode.
Note that when using interrupts, set up to enable the interrupts. For details, refer to section **"CHAPTER 6. INTERRUPTS."**

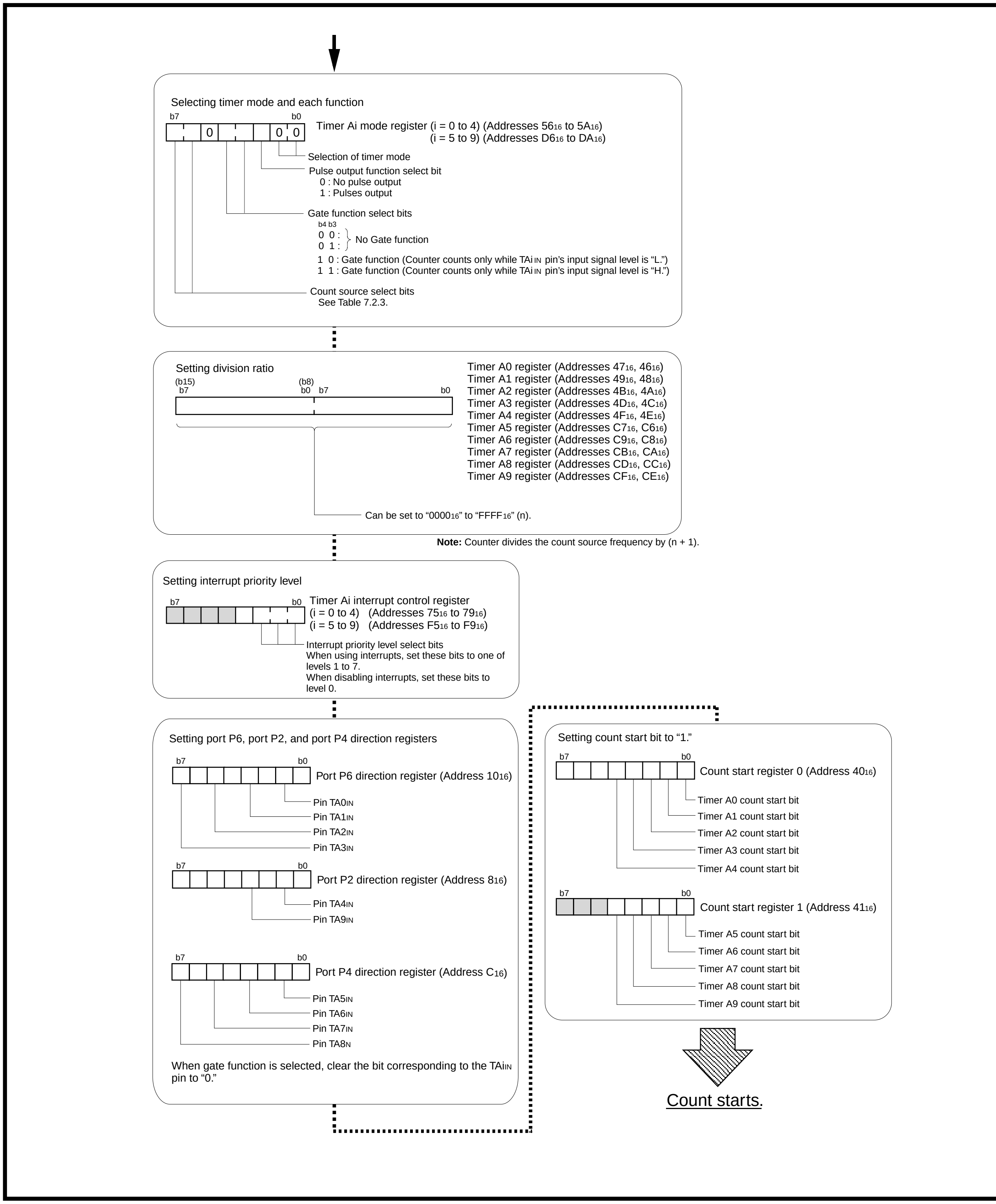


Fig. 7.3.2 Initial setting example for registers relevant to timer mode

7.3.2 Operation in timer mode

① When the count start bit is set to "1," the counter starts counting of the count source.
② When a counter underflow occurs, the reload register's contents are reloaded, and counting continues.
③ The timer Ai interrupt request bit is set to "1" at the underflow in ②. The interrupt request bit remains set to "1" until the interrupt request is accepted or until the interrupt request bit is cleared to "0" by software.

Figure 7.3.3 shows an example of operation in the timer mode.

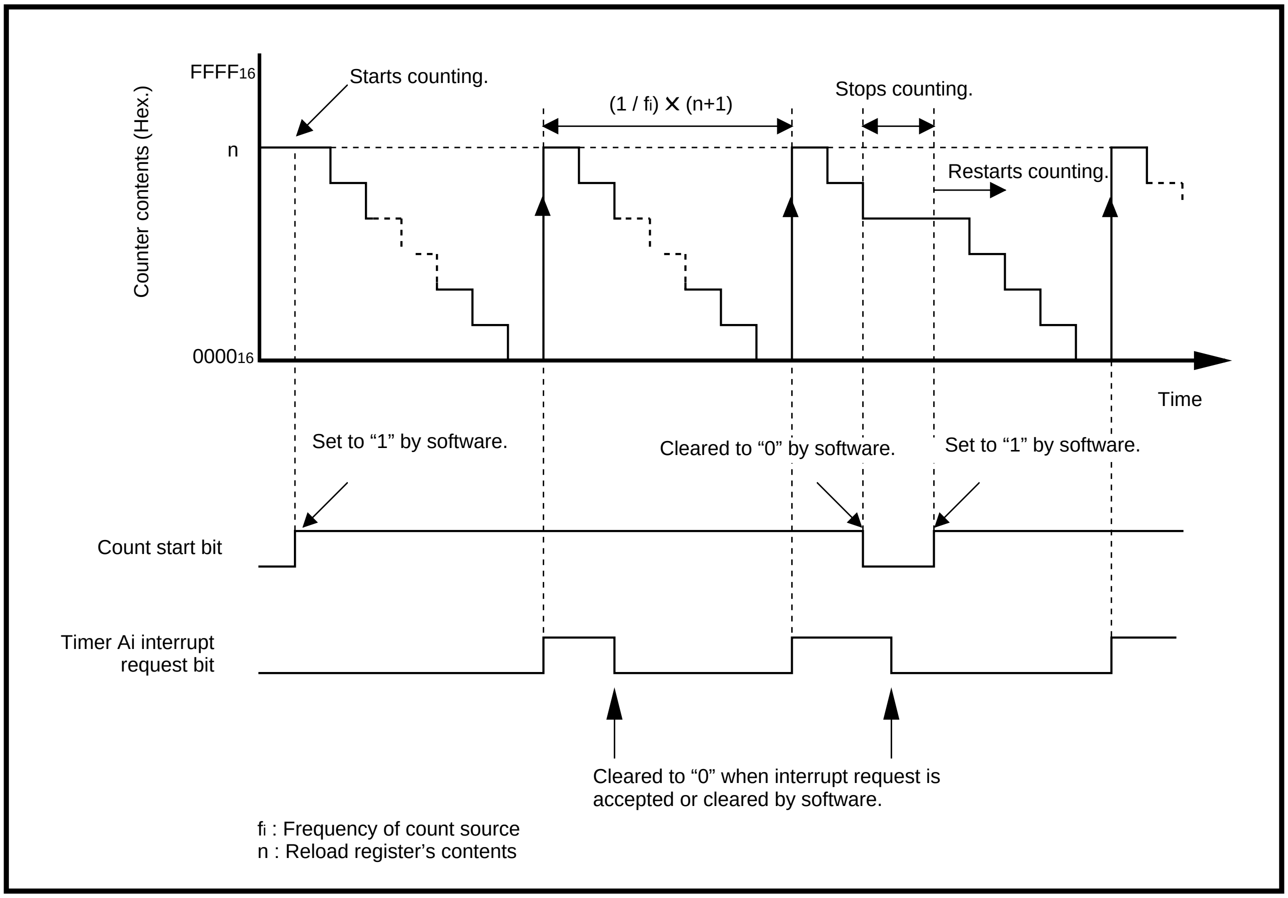


Fig. 7.3.3 Example of operation in timer mode (without pulse output and gate functions)

7.3.3 Select function

The following describes the gate and pulse output functions.

(1) Gate function

The gate function is selected by setting the gate function select bits (bits 4 and 3 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16}) to "10_2" or "11_2." The gate function makes it possible to start or stop counting depending on the TAi_{IN} pin's input signal. Table 7.3.2 lists the count valid levels.

Figure 7.3.4 shows an example of operation with the gate function selected.

When selecting the gate function, set the port P6, P2, and P4 direction registers' bits which correspond to the TAi_{IN} pins for the input mode. Additionally, make sure that the TAi_{IN} pin's input signal has a pulse width equal to or more than two cycles of the count source.

Table 7.3.2 Count valid levels

Gate function select bits		Count valid level (Duration while counter counts)
b4	b3	
1	0	While TAi_{IN} pin's input signal level is at "L" level
1	1	While TAi_{IN} pin's input signal level is at "H" level

Note: The counter does not count while the TAi_{IN} pin's input signal is not at the count valid level.

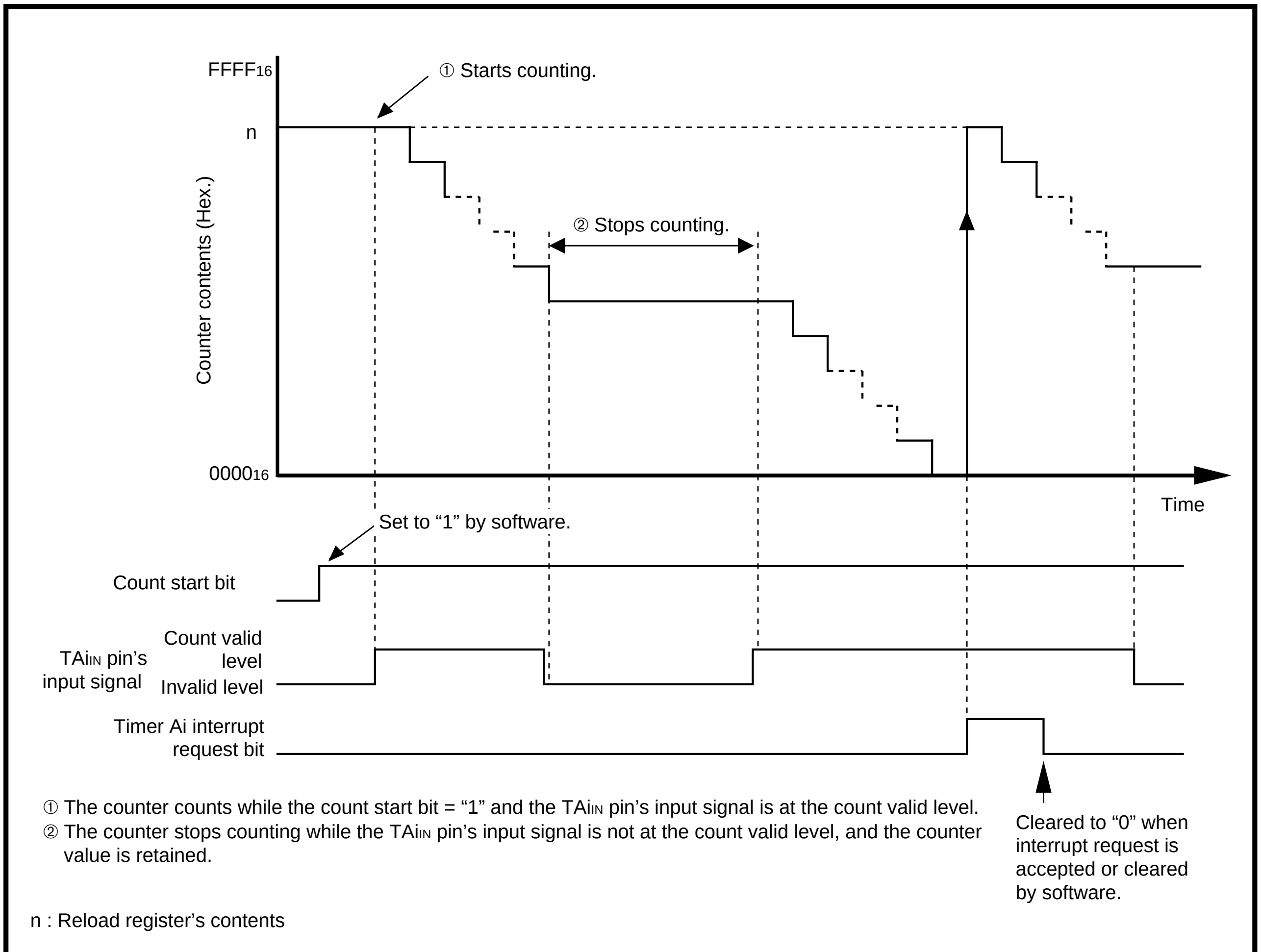


Fig. 7.3.4 Example of operation with gate function selected

(2) Pulse output function

The pulse output function is selected by setting the pulse output function select bit (bit 2 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16}) to “1.” When this function is selected, the TAi_{OUT} pin is forcibly set for the pulse output pin regardless of the corresponding bits of the port P6, P2, and P4 direction registers. The TAi_{OUT} pin outputs a pulse of which polarity is inverted each time a counter underflow occurs. When the count start bit (addresses 40_{16}, 41_{16}) is “0” (count stopped), the TAi_{OUT} pin outputs “L” level. Figure 7.3.5 shows an example of operation with the pulse output function selected.

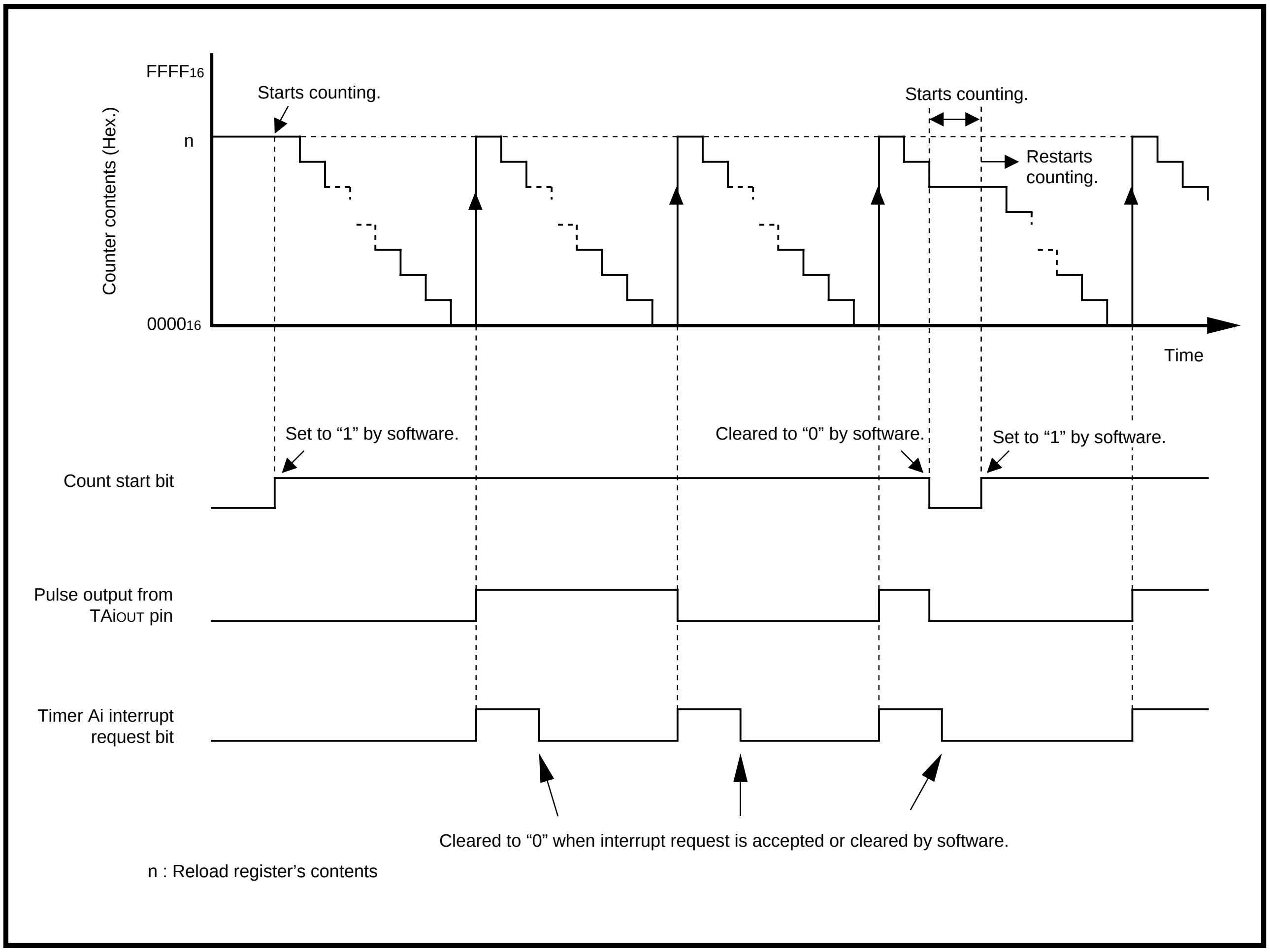


Fig. 7.3.5 Example of operation with pulse output function selected

[Precautions for timer mode]

1. By reading the timer Ai register, the counter value can be read out at arbitrary timing. However, if the timer Ai register is read at the reload timing shown in Figure 7.3.6, the value "$FFFF_{16}$" is read out. If reading is performed in the period from when a value is set into the timer Ai register with the counter stopped until the counter starts counting, the set value is correctly read out.

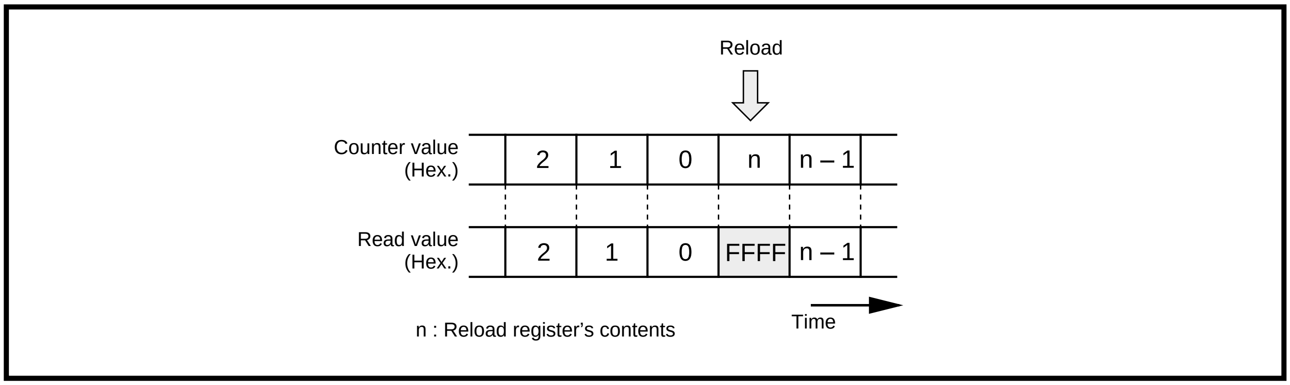


Fig. 7.3.6 Reading timer Ai register

7.4 Event counter mode

In this mode, the timer counts an external signal.
Tables 7.4.1 and 7.4.2 list the specifications of the event counter mode. Figure 7.4.1 shows the structures of the timer Ai register and timer Ai mode register in the event counter mode.

Table 7.4.1 Specifications of event counter mode (when not using two-phase pulse signal processing function)

Item	Specifications
Count source	● External signal input to the TAi_{IN} pin ● The count source's valid edge can be selected from the falling edge and the rising edge by software.
Count operation	● Countup or countdown can be switched by external signal or software. ● When a counter overflow or underflow occurs, reload register's contents are reloaded, and counting continues.
Division ratio	● For countdown $\frac{1}{(n + 1)}$ ● For countup $\frac{1}{(FFFF_{16} - n + 1)}$ n: Timer Ai register's set value
Count start condition	When the count start bit is set to "1."
Count stop condition	When the count start bit is cleared to "0."
Interrupt request occurrence timing	When a counter overflow or underflow occurs.
TAi_{IN} pin's function	Count source input
TAi_{OUT} pin's function	Programmable I/O port pin, pulse output pin, or countup/countdown switch signal input pin
Read from timer Ai register	Counter value can be read out.
Write to timer Ai register	● While counting is stopped When a value is written to timer Ai register, it is written to both of the reload register and counter. ● While counting is in progress When a value is written to timer Ai register, it is written only to the reload register. (Transferred to the counter at the next reload timing.)

Table 7.4.2 Specifications of event counter mode (when using two-phase pulse signal processing function in timers A2 to A4, A7 to A9)

Item	Specifications
Count source	External signal (two-phase pulse) input to the following pins: TAj_{IN}, TAj_{OUT} (j = 2 to 4, 7 to 9)
Count operation	● Countup or countdown can be switched by external signal (two-phase pulse). ● When a counter overflow or underflow occurs, reload register's contents are reloaded, and counting continues.
Division ratio	● For countdown $\frac{1}{(n + 1)}$ ● For countup $\frac{1}{(FFFF_{16} - n + 1)}$ n: Timer Aj register's set value
Count start condition	When the count start bit is set to "1."
Count stop condition	When the count start bit is cleared to "0."
Interrupt request occurrence timing	When a counter overflow or underflow occurs.
Function of the following pins: TAj_{IN}, TAj_{OUT} (j = 2 to 4, 7 to 9)	Two-phase pulse input
Read from timer Aj register	Counter value can be read out by reading timer Aj register.
Write to timer Aj register	● While counting is stopped When a value is written to timer Aj register, it is written to both of the reload register and counter. ● While counting is in progress When a value is written to timer Aj register, it is written only to the reload register. (Transferred to the counter at the next reload timing.)

7.4 Event counter mode

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 — (b8) b0 | b7 — b0

Bit	Function	At reset	R/W
15 to 0	Any value in the range from "0000_{16}" to "$FFFF_{16}$" can be set. Assuming that the set value = n, the counter divides the count source frequency by (n + 1) during countdown, or by ($FFFF_{16}$ – n + 1) during countup. When reading, the register indicates the counter value.	Undefined	RW

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0
X	X	0				0	1

Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits	b1 b0 0 1 : Event counter mode	0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output (TAi_{OUT} pin functions as a programmable I/O port pin.) 1 : Pulse output (TAi_{OUT} pin functions as a pulse output pin.)	0	RW
3	Count polarity select bit	0 : Counts at falling edge of external signal 1 : Counts at rising edge of external signal	0	RW
4	Up-down switching factor select bit	0 : Contents of up-down register 1 : Input signal to TAi_{OUT} pin	0	RW
5	Fix this bit to "0" in event counter mode.		0	RW
6	These bits are invalid in event counter mode.		0	RW
7			0	RW

X : It may be either "0" or "1."

Fig. 7.4.1 Structures of timer Ai register and timer Ai mode register in event counter mode

7.4.1 Setting for event counter mode

Figures 7.4.2 and 7.4.3 show an initial setting example for registers related to the event counter mode. Note that when using interrupts, set up to enable the interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."**

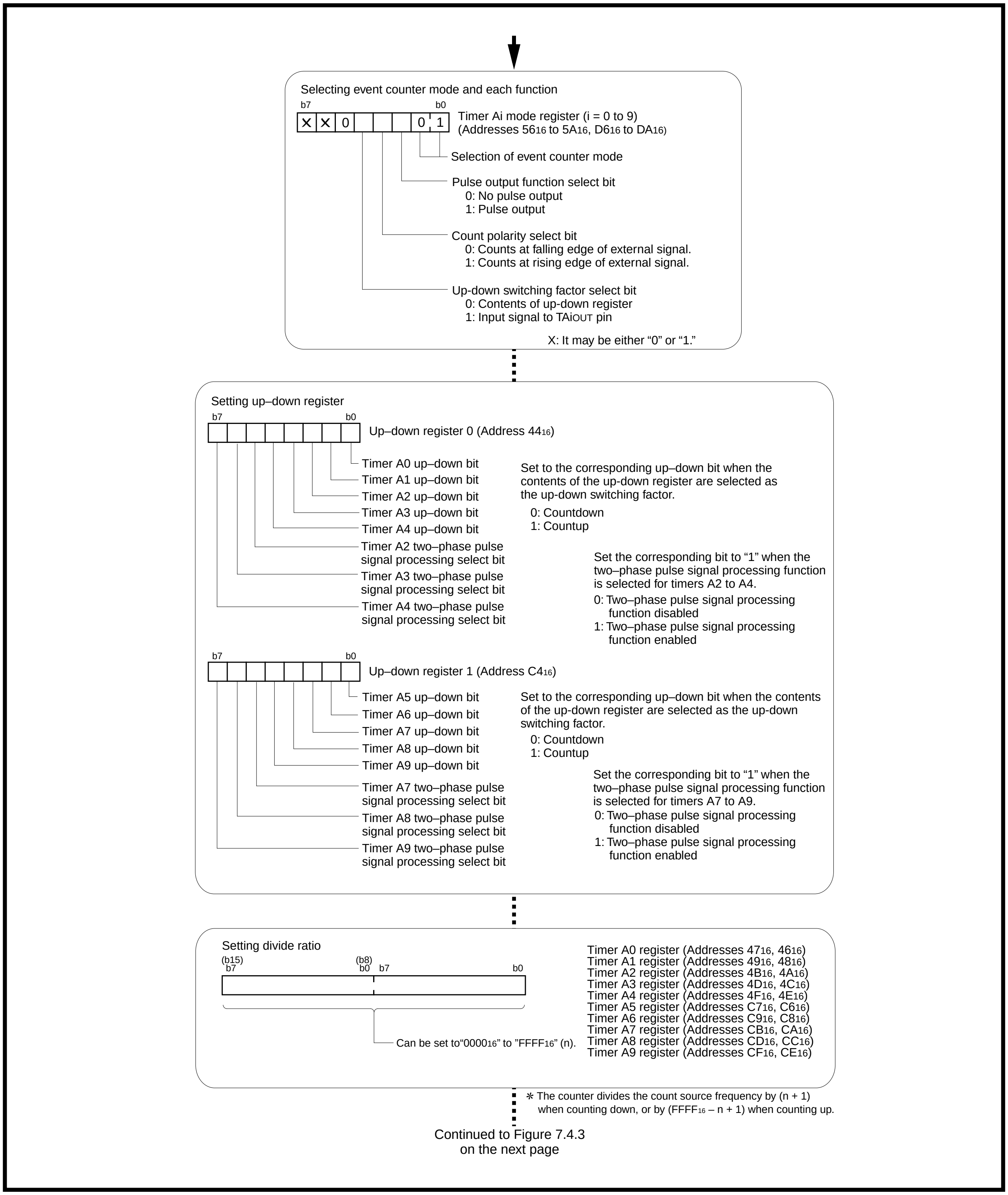


Fig. 7.4.2 Initial setting example for registers related to event counter mode (1)

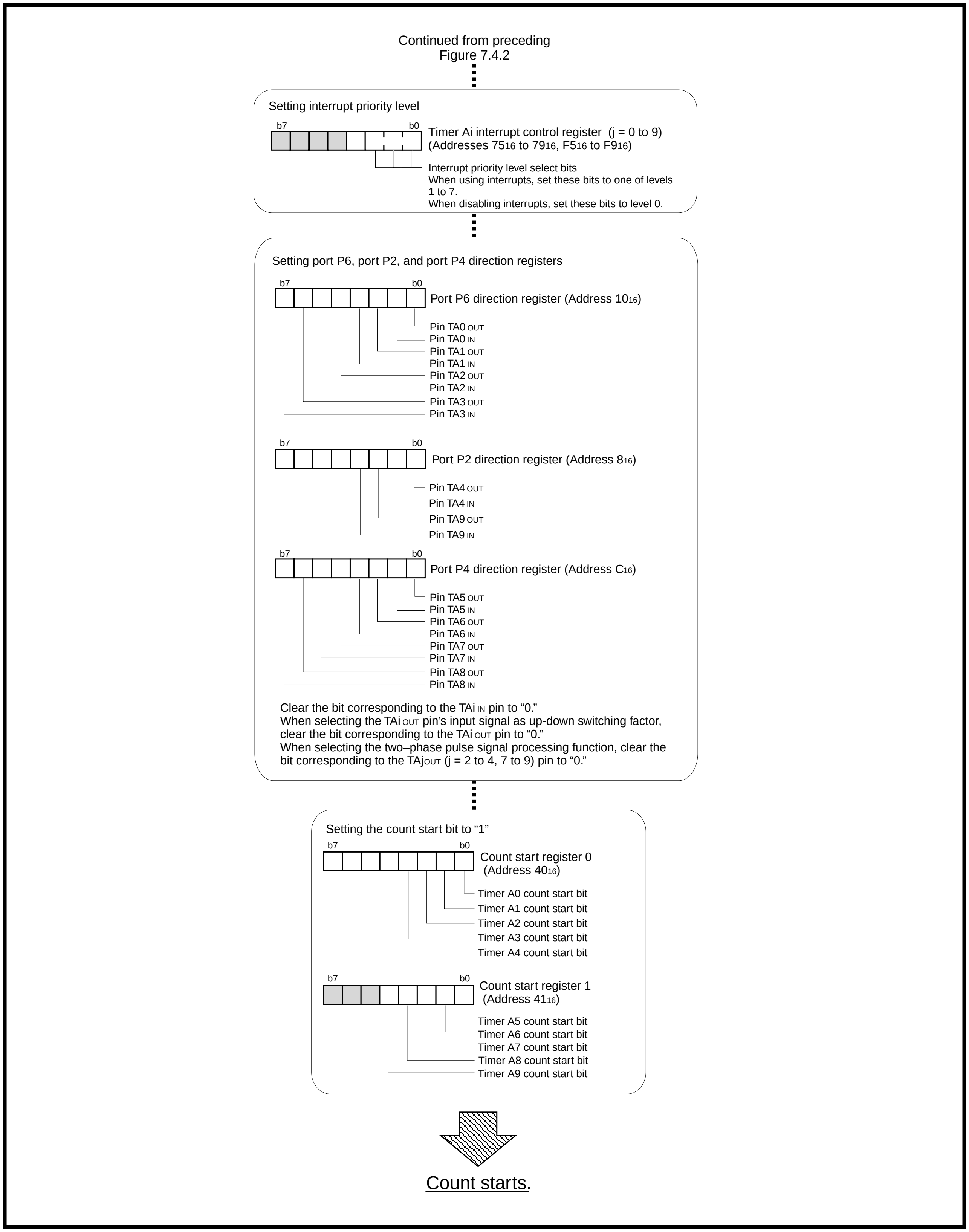
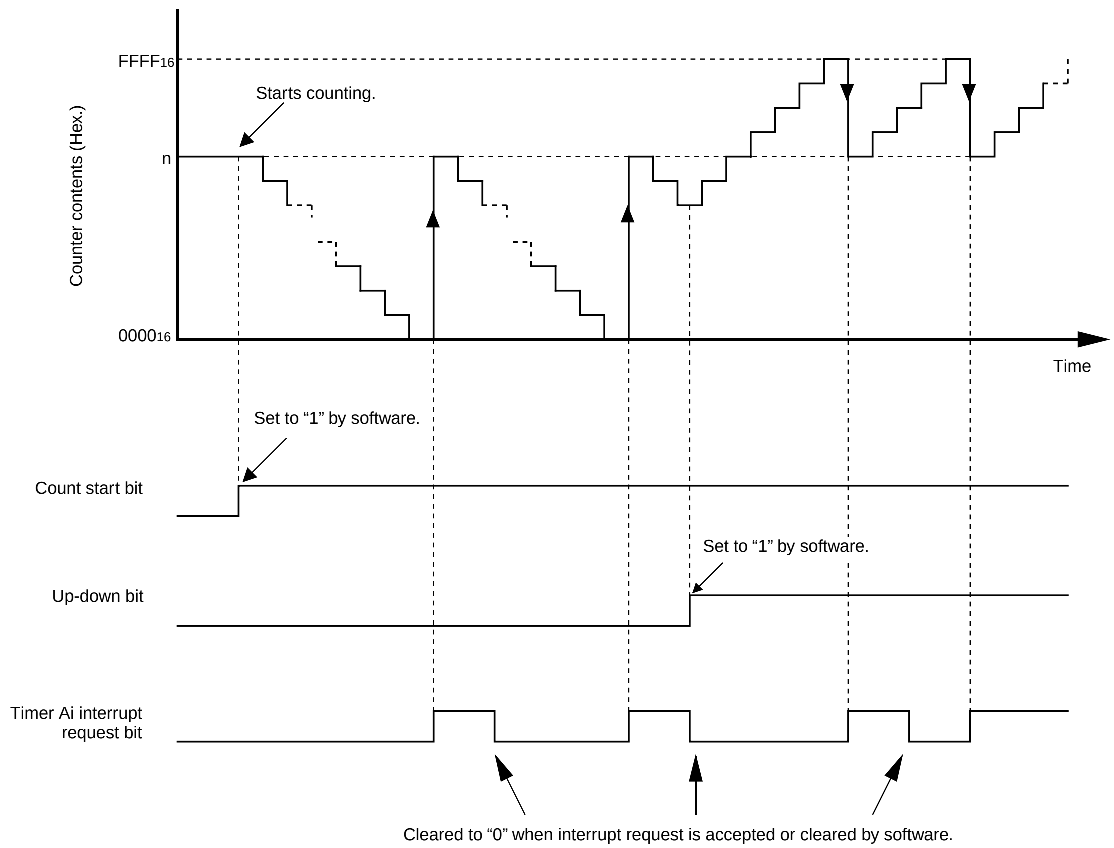


Fig. 7.4.3 Initial setting example for registers relevant to event counter mode (2)

7.4.2 Operation in event counter mode

① When the count start bit is set to “1,” the counter starts counting of the count source’s valid edge.
② When a counter underflow or overflow occurs, the reload register’s contents are reloaded, and counting continues.
③ The timer Ai interrupt request bit is set to “1” at the underflow or overflow in ②.
The interrupt request bit remains set to “1” until the interrupt request is accepted or until the interrupt request bit is cleared to “0” by software.

Figure 7.4.4 shows an example of operation in the event counter mode.



n : Reload register’s contents

Note: The above applies when the up-down bit’s contents are selected as the up-down switching factor (i.e., up-down switching factor select bit = “0”).

Fig. 7.4.4 Example of operation in event counter mode (without pulse output and two-phase pulse signal processing functions)

7.4.3 Switching between countup and countdown

Figure 7.4.5 shows structures of the up-down registers 0 and 1.
The up-down register or the input signal from the TAi_{OUT} pin is used to switch countup from and to countdown. This switching is performed by the up-down bit when the up-down switching factor select bit (bit 4 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16}) is "0," and by the input signal from the TAi_{OUT} pin when the up-down switching factor select bit is "1."
When the switching between countup and countdown is set while counting is in progress, this switching is actually performed when the count source's next valid edge is input.

(1) Switching by up-down bit

Countdown is performed when the up-down bit is "0," and countup is performed when the up-down bit is "1." Figure 7.4.5 shows the structures of the up-down registers 0 and 1.

(2) Switching by TAi_{OUT} pin's input signal

Countdown is performed when the TAi_{OUT} pin's input signal is at "L" level, and countup is performed when the TAi_{OUT} pin's input signal is at "H" level.
When using the TAi_{OUT} pin's input signal to switch countup from and to countdown, set the port P6, port P2, and port P4 direction registers' bits which correspond to the TAi_{OUT} pin for the input mode.

Up-down register 0 (Address 44_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	Timer A0 up-down bit	0 : Countdown 1 : Countup This function is valid when the contents of the up-down register is selected as the up-down switching factor.	0	RW
1	Timer A1 up-down bit		0	RW
2	Timer A2 up-down bit		0	RW
3	Timer A3 up-down bit		0	RW
4	Timer A4 up-down bit		0	RW
5	Timer A2 two-phase pulse signal processing select bit	0 : Two-phase pulse signal processing function disabled 1 : Two-phase pulse signal processing function enabled When not using the two-phase pulse signal processing function, clear the bit to "0." The value is "0" at reading.	0	WO **(Note)**
6	Timer A3 two-phase pulse signal processing select bit		0	WO **(Note)**
7	Timer A4 two-phase pulse signal processing select bit		0	WO **(Note)**

Note: Use the **MOVM (MOVMB)** or **STA(STAB, STAD)** instruction for writing to bits 5 to 7.

Up-down register 1 (Address $C4_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	Timer A5 up-down bit	0 : Countdown 1 : Countup This function is valid when the contents of the up-down register is selected as the up-down switching factor.	0	RW
1	Timer A6 up-down bit		0	RW
2	Timer A7 up-down bit		0	RW
3	Timer A8 up-down bit		0	RW
4	Timer A9 up-down bit		0	RW
5	Timer A7 two-phase pulse signal processing select bit	0 : Two-phase pulse signal processing function disabled 1 : Two-phase pulse signal processing function enabled When not using the two-phase pulse signal processing function, clear the bit to "0." The value is "0" at reading.	0	WO **(Note)**
6	Timer A8 two-phase pulse signal processing select bit		0	WO **(Note)**
7	Timer A9 two-phase pulse signal processing select bit		0	WO **(Note)**

Note: Use the **MOVM (MOVMB)** or **STA(STAB, STAD)** instruction for writing to bits 5 to 7.

Fig. 7.4.5 Structures of up-down registers 0 and 1

7.4 Event counter mode

7.4.4 Selectable functions

The following describes the selectable pulse output, and two-phase pulse signal processing functions.

(1) Pulse output function

The pulse output function is selected by setting the pulse output function select bit (bit 2 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16}) to “1.” When this function is selected, the TAi_{OUT} pin is forcibly set for the pulse output pin regardless of the corresponding bits of the port P6, port P2, and port P4 direction registers. The TAi_{OUT} pin outputs a pulse of which polarity is inverted each time a counter underflow or overflow occurs. (Refer to Figure 7.3.5).

When the count start bit (addresses 40_{16}, 41_{16}) is “0” (count stopped), the TAi_{OUT} pin outputs “L” level.

(2) Two-phase pulse signal processing function (Timers Aj)

For timer Aj (j = 2 to 4, 7 to 9), the two-phase pulse signal processing function is selected by setting the timer Aj two-phase pulse signal processing select bits (bits 5 to 7 at addresses 44_{16} and $C4_{16}$) to “1.” (See Figure 7.4.5.) Figure 7.4.6 shows the timer Aj mode register when the two-phase pulse signal processing function is selected.

For timers with two-phase pulse signal processing function selected, the timer counts two kinds of pulses of which phases differ by 90 degrees. There are two types of the two-phase pulse signal processing: normal processing and quadruple processing. In timer Am (m = 2, 3, 7, 8), normal processing is performed; in timer An (n = 4, 9), quadruple processing is performed.

For the port P6, port P2, and P4 direction registers' bits corresponding to the pins used for two-phase pulse input, be sure to set these bits for the input mode.

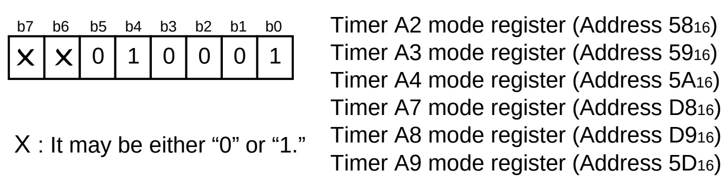


Fig. 7.4.6 Timer Aj (j = 2 to 4, 7 to 9) mode register when two-phase pulse signal processing function is selected

<Normal processing>

Countup is performed at the rising edges input to the TAmIN pin when the TAmIN (m = 2, 3, 7, 8) and TAmOUT have the relationship that the TAmIN pin's input signal goes from "L" to "H" while the TAmOUT pin's input signal is at "H" level.

Countdown is performed at the falling edges input to the TAmIN pin when the TAmIN and TAmOUT have the relationship that the TAmIN pin's input signal goes from "H" to "L" while the TAmOUT pin's input signal is "H." (See Figure 7.4.7.)

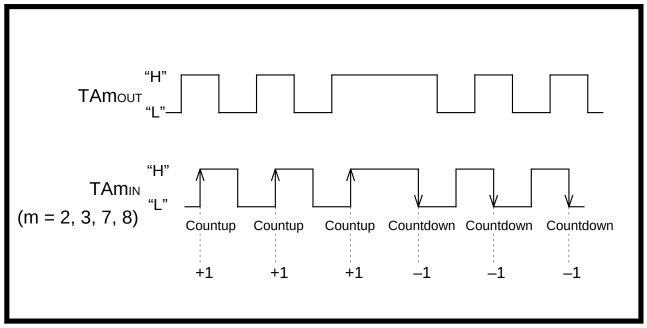


Fig. 7.4.7 Normal processing

<Quadruple processing>

Countup is performed at all rising and falling edges input to the TAnOUT (n = 4, 9) and TAnIN pins when the TAnIN and TAnOUT have the relationship that the TAnIN pin's input signal level goes from "L" to "H" while the TAnOUT pin's input signal is at "H" level.

Countdown is performed at all rising and falling edges input to the TAnOUT and TAnIN pins when the TAnIN and TAnOUT have the relationship that the TAnIN pin's input signal level goes from "H" to "L" while the TAnOUT pin's input signal is at "H" level. (See Figure 7.4.8.)

Table 7.4.3 lists the input signals on the TAnOUT and TAnIN pins when the quadruple processing is selected.

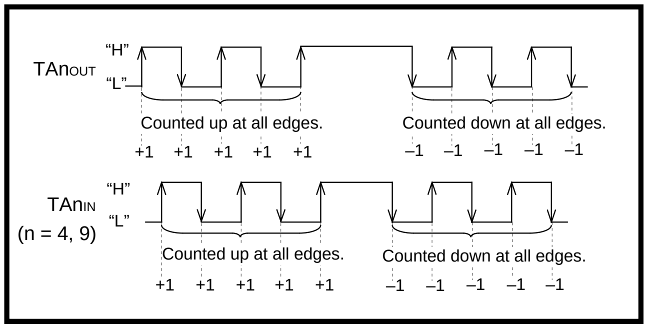


Fig. 7.4.8 Quadruple processing

Table 7.4.3 TAnOUT and TAnIN pin's input signals when quadruple processing is selected (n = 4, 9)

	Input signal to TAnOUT pin	Input signal to TAnIN pin
Countup	"H" level	Rising edge
	"L" level	Falling edge
	Rising edge	"L" level
	Falling edge	"H" level
Countdown	"H" level	Falling edge
	"L" level	Rising edge
	Rising edge	"H" level
	Falling edge	"L" level

[Precautions for event counter mode]

1. While counting is in progress, by reading the timer Ai (i = 0 to 9) register, the counter value can be read out at any timing. However, if the timer Ai register is read at the reload timing shown in Figure 7.4.9, the value "$FFFF_{16}$" (at an underflow) or "0000_{16}" (at the overflow) is read out. If reading is performed in the period from when a value is set into the timer Ai register with the counter stopped until the counter starts counting, the set value is correctly read out.

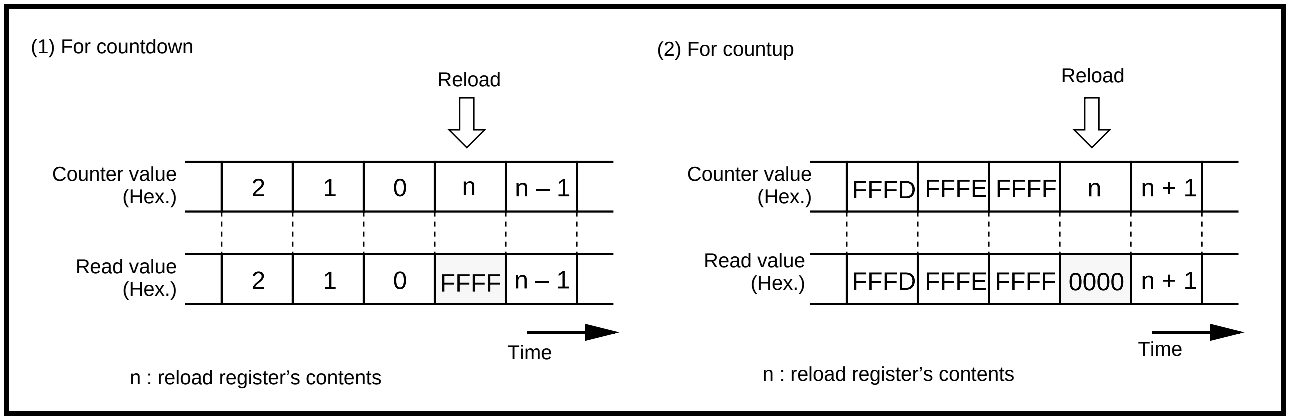


Fig. 7.4.9 Reading timer Ai register

2. The TAi_{OUT} pin is used for all functions listed below. Accordingly, only one of these functions can be selected for each timer.
   - Switching between countup and countdown by TAi_{OUT} pin's input signal
   - Pulse output function
   - Two-phase pulse signal processing function (Timers A2 to A4, A7 to A9)

7.5 One-shot pulse mode

In this mode, the timer outputs a pulse which has an arbitrary width once.
When a trigger occurs, the timer outputs "H" level from the TAi_{OUT} pin for an arbitrary time. Table 7.5.1 lists the specifications of the one-shot pulse mode. Figure 7.5.1 shows the structures of the timer Ai register and timer Ai mode register in the one-shot pulse mode.

Table 7.5.1 Specifications of one-shot pulse mode

Item	Specifications
Count source f_i	f_1, f_2, f_{16}, f_{64}, f_{512}, or f_{4096}
Count operation	● Countdown ● When the counter value becomes "0000_{16}," reload register's contents are reloaded, and counting stops. ● If a trigger occurs during counting, reload register's contents are reloaded, and counting continues.
Output pulse width ("H")	$\frac{n}{f_i}$ [s] n : Timer Ai register's set value
Count start condition	● When a trigger occurs. **(Note)** ● Internal or external trigger can be selected by software.
Count stop condition	● When the counter value becomes "0000_{16}" ● When the count start bit is cleared to "0"
Interrupt request occurrence timing	When counting stops.
TAi_{IN} pin's function	Programmable I/O port pin or trigger input pin
TAi_{OUT} pin's function	One-shot pulse output
Read from timer Ai register	An undefined value is read out.
Write to timer Ai register	● While counting is stopped When a value is written to timer Ai register, it is written to both of the reload register and counter. ● While counting is in progress When a value is written to timer Ai register, it is written only to the reload register. (Transferred to counter at the next reload timing.)

Note: The trigger is generated with the count start bit = "1."

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 (b8) b0 b7 b0

Bit	Function	At reset	R/W
15 to 0	Any value in the range from "0000_{16}" to "$FFFF_{16}$" can be set. Assuming that the set value = n, the "H" level width of the one-shot pulse which is output from the TAi_{OUT} pin is expressed as follows : $\frac{n}{f_i}$.	Undefined	WO

f_i: Frequency of count source

Note: Use the **MOVM** or **STA(STAD)** instruction for writing to this register.
Writing to this register must be performed in a unit of 16 bits.

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0
		0			1	1	0

Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits	b1 b0 1 0 : One-shot pulse mode	0	RW
1			0	RW
2	Fix this bit to "1" in one-shot pulse mode.		0	RW
3	Trigger select bits	b4 b3 0 0 : } Writing "1" to one-shot start bit 0 1 : } (TAi_{IN} pin functions as a programmable I/O port pin.) 1 0 : Falling edge of TAi_{IN} pin's input signal 1 1 : Rising edge of TAi_{IN} pin's input signal	0	RW
4			0	RW
5	Fix this bit to "0" in one-shot pulse mode.		0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 7.5.1 Structures of timer Ai register and timer Ai mode register in one-shot pulse mode

7.5.1 Setting for one-shot pulse mode

Figures 7.5.2 and 7.5.3 show an initial setting example for registers related to the one-shot pulse mode. Note that when using interrupts, set up to enable the interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."**

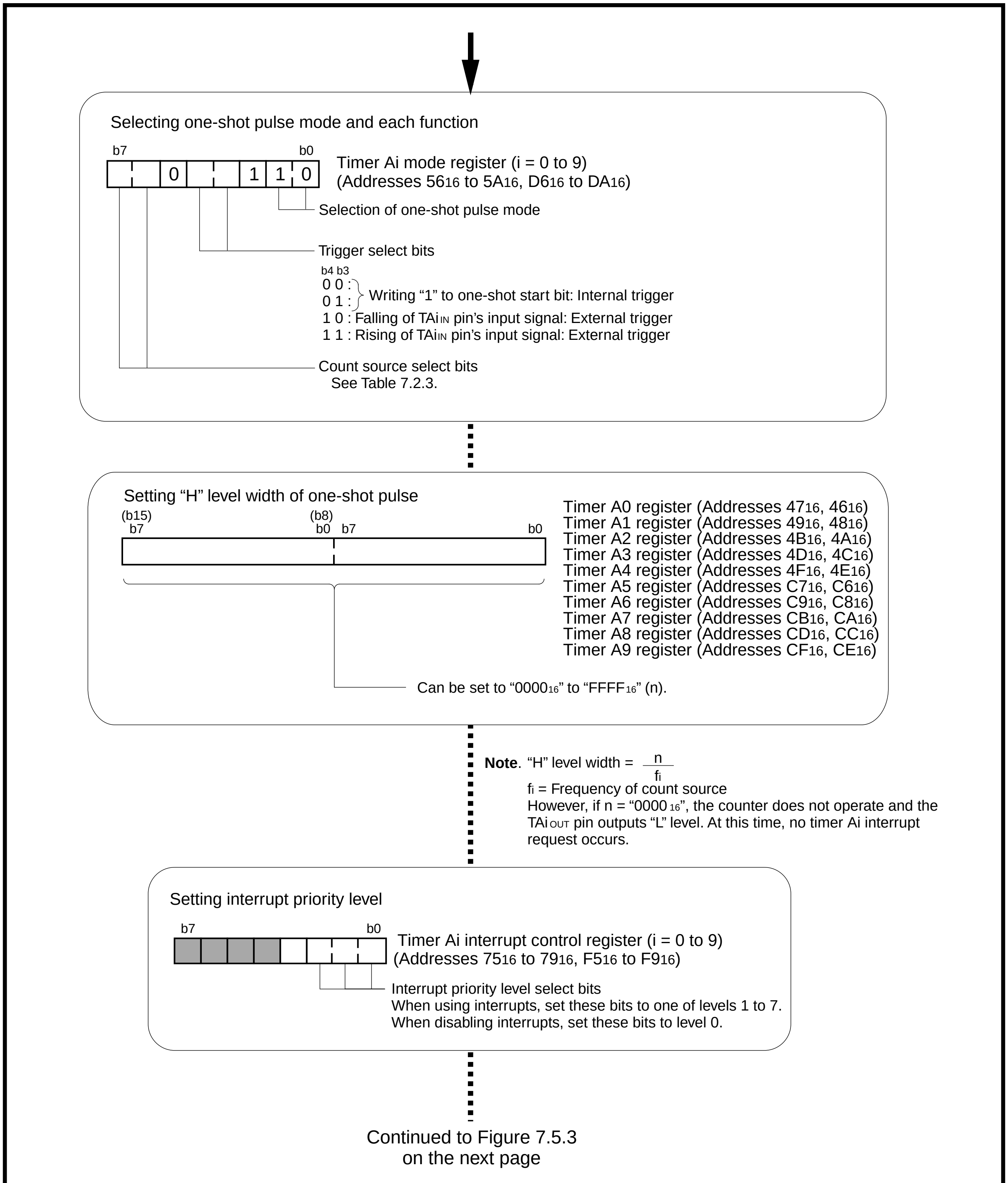


Fig. 7.5.2 Initial setting example for registers related to one-shot pulse mode (1)

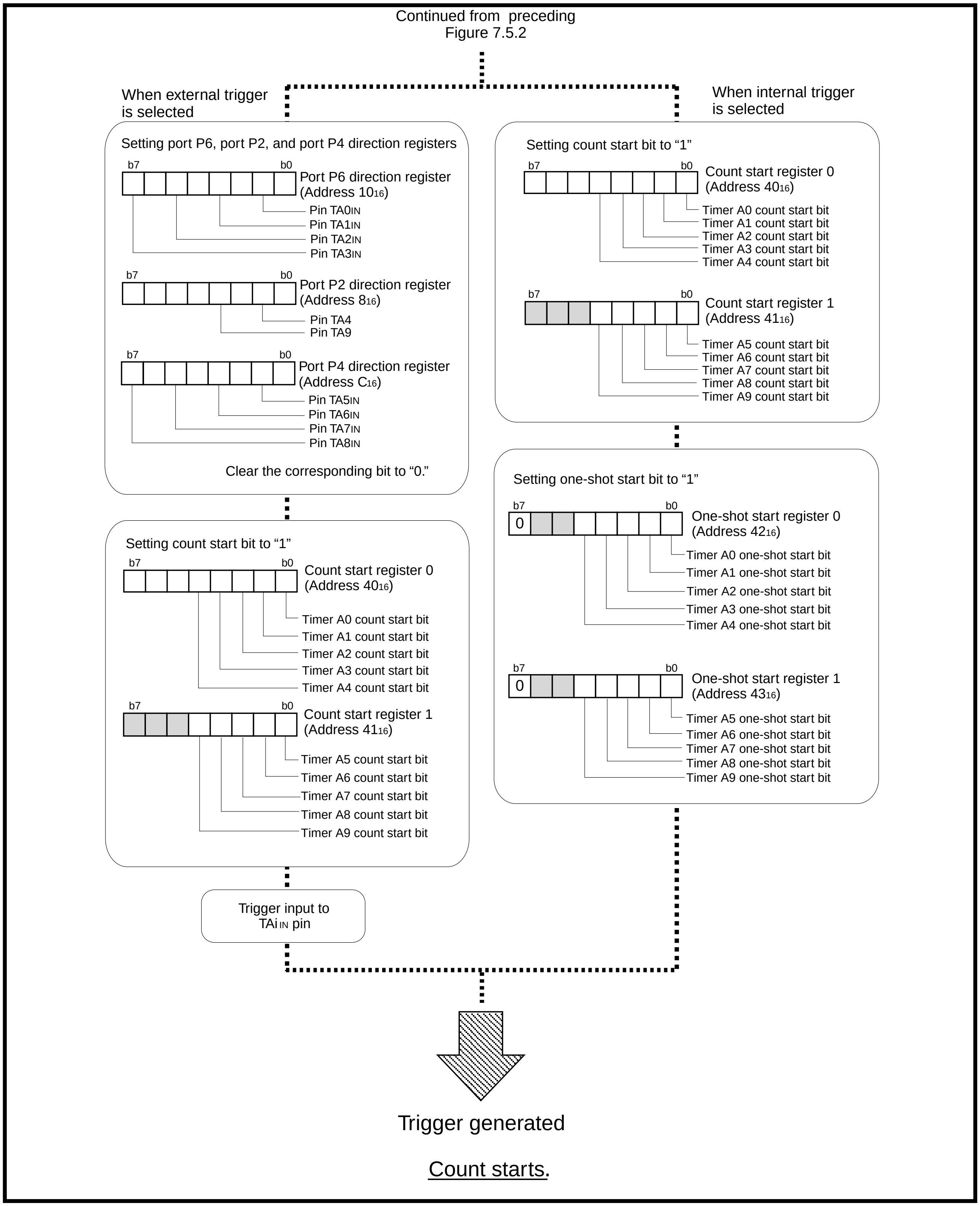


Fig. 7.5.3 Initial setting example for registers related to one-shot pulse mode (2)

7.5.2 Trigger

The counter is enabled for counting when the count start bit (addresses 40_{16}, 41_{16}) has been set to "1." The counter starts counting when a trigger is generated after counting has been enabled. An internal or external trigger can be selected as that trigger.

An internal trigger is selected when the trigger select bits (bits 4 and 3 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16}) are "00_2" or "01_2"; an external trigger is selected when the bits are "10_2" or "11_2."

If a trigger is generated during counting, the reload register's contents are reloaded and the counter continues counting. If a trigger generated during counting, make sure that a certain time which is equivalent to one cycle of the timer's count source or more has passed between the previously trigger occurrence and a new trigger occurrence.

(1) When selecting internal trigger

A trigger is generated when writing "1" to the one-shot start bit (addresses 42_{16}, 43_{16}). Figure 7.5.4 shows the structures of the one-shot start registers 0 and 1.

(2) When selecting external trigger

A trigger is generated at the falling edge of the TAi_{IN} pin's input signal when bit 3 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16} is "0," or at its rising edge when bit 3 is "1."

When using an external trigger, set the port P6, port P2, and port P4 direction registers' bits which correspond to the TAi_{IN} pins for the input mode.

One-shot start register 0 (Address 42_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0							

Bit	Bit name	Function	At reset	R/W
0	Timer A0 one-shot start bit	1 : Start outputting one-shot pulse. (Valid when an internal trigger is selected.) The value is “0” at reading.	0	WO
1	Timer A1 one-shot start bit		0	WO
2	Timer A2 one-shot start bit		0	WO
3	Timer A3 one-shot start bit		0	WO
4	Timer A4 one-shot start bit		0	WO
6, 5	Nothing is assigned.		Undefined	–
7	Fix this bit to “0.”		0	RW

One-shot start register 1 (Address 43_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0							

Bit	Bit name	Function	At reset	R/W
0	Timer A5 one-shot start bit	1 : Start outputting one-shot pulse. (Valid when an internal trigger is selected.) The value is “0” at reading.	0	WO
1	Timer A6 one-shot start bit		0	WO
2	Timer A7 one-shot start bit		0	WO
3	Timer A8 one-shot start bit		0	WO
4	Timer A9 one-shot start bit		0	WO
6, 5	Nothing is assigned.		Undefined	–
7	Fix this bit to “0.”		0	RW

Fig. 7.5.4 Structures of one-shot start registers 0 and 1

7.5.3 Operation in one-shot pulse mode

① When the one-shot pulse mode is selected with the operating mode select bits, the TAi_{OUT} pin outputs "L" level.

② When the count start bit is set to "1," the counter is enabled for counting. After that, counting starts when a trigger is generated.

③ When the counter starts counting, the TAi_{OUT} pin outputs "H" level. (When a value of "0000_{16}" is set to the timer Ai register, the counter stops operating, the output level at pin TAi_{OUT} remains "L," and no timer Ai interrupt request does not occur.)

④ When the counter value becomes "0000_{16}," the output from the TAi_{OUT} pin becomes "L" level. Additionally, the reload register's contents are reloaded and the counter stops counting there.

⑤ Simultaneously with ④, the timer Ai interrupt request bit is set to "1." This interrupt request bit remains set to "1" until the interrupt request is accepted or until the interrupt request bit is cleared to "0" by software.

Figure 7.5.5 shows an example of operation in the one-shot pulse mode.
When a trigger is generated after ④ above, the counter and TAi_{OUT} pin perform the same operations beginning from ② again. Furthermore, if a trigger is generated during counting, the counter performs countdown once after this new trigger is generated, and then, it continues counting with the reload register's contents reloaded. If generating a trigger during counting, make sure that a certain time which is equivalent to one cycle of the timer's count source or more has passed between the previously trigger occurrence and a new trigger occurrence.

The one-shot pulse output from the TAi_{OUT} pin can be disabled by clearing the timer Ai mode register's bit 2 to "0." Accordingly, timer Ai can also be used as an internal one-shot timer that does not perform the pulse output. In this case, the TAi_{OUT} pin functions as a programmable I/O port pin.

7.5 One-shot pulse mode

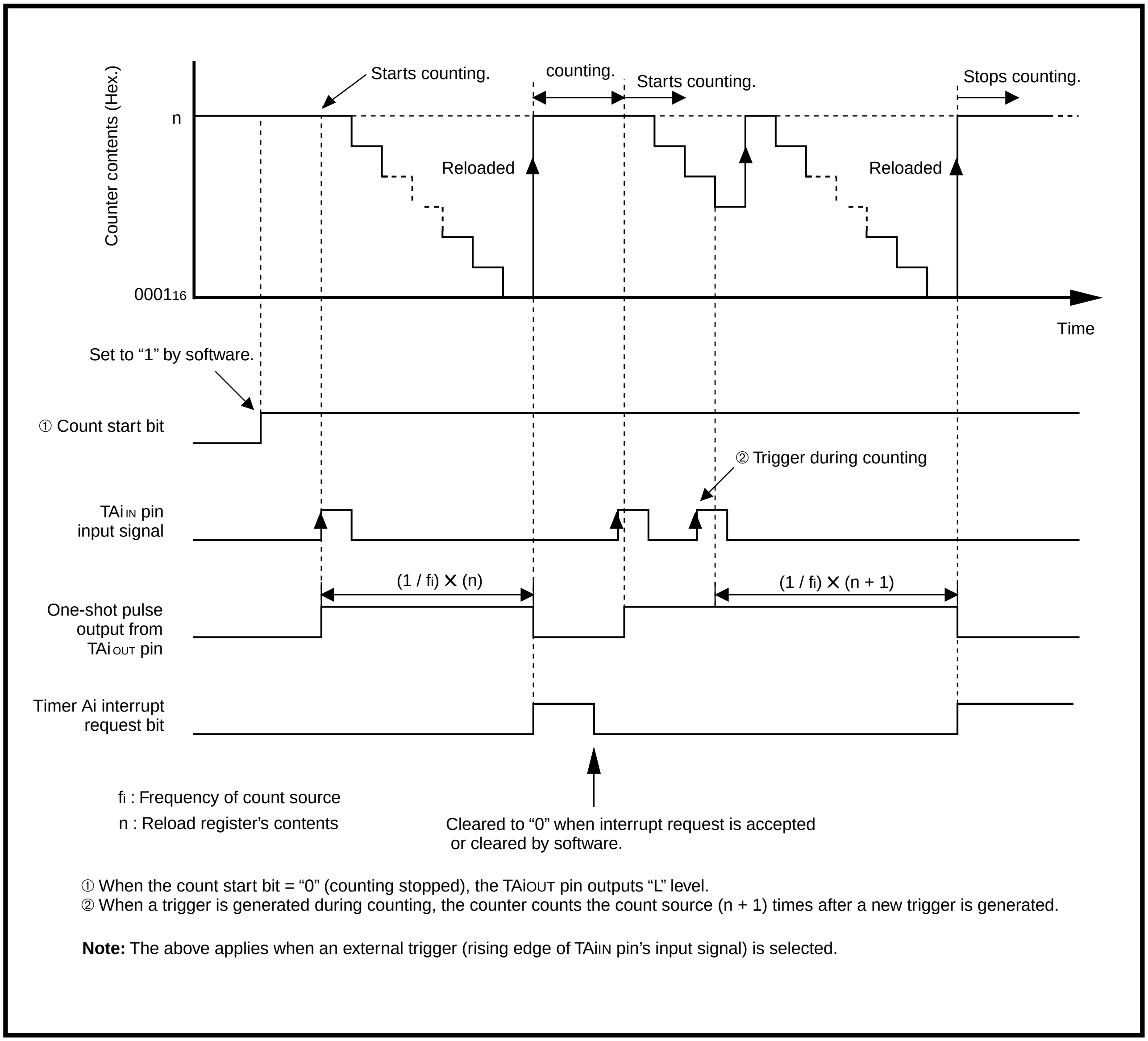


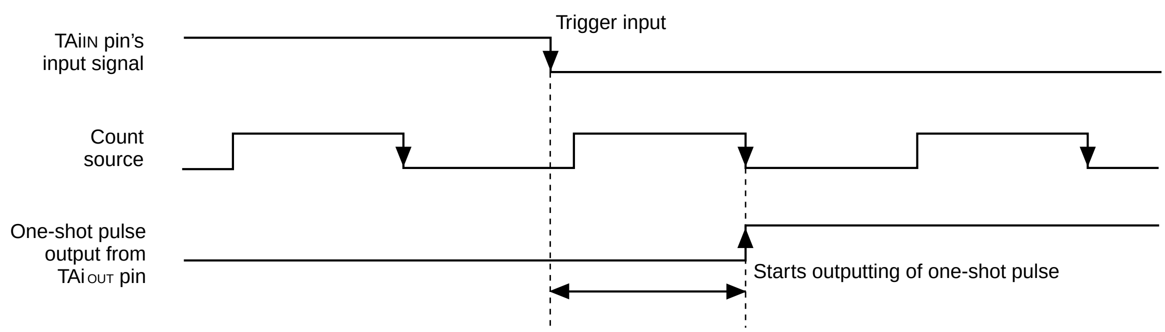
Fig. 7.5.5 Example of operation in one-shot pulse mode (selecting external trigger)

[Precautions for one-shot pulse mode]

1. If the count start bit is cleared to “0” during counting, the counter becomes as follows:
   •The counter stops counting, and the reload register’s contents are reloaded into the counter.
   •The TAi_{OUT} pin’s output level becomes “L.”
   •The timer Ai interrupt request bit is set to “1.”

2. A one-shot pulse is output synchronously with an internally generated count source. Accordingly, when selecting an external trigger, there will be a delay equivalent to one cycle of the count source at maximum, in a period from when a trigger is input to the TAi_{IN} pin until a one-shot pulse is output.

Fig. 7.5.6 Output delay in one-shot pulse output



Note: The above applies when an external trigger (falling edge of TAi_{IN} pin’s input signal) is selected.

3. When the timer’s operating mode has been set by one of the following procedures, the timer Ai interrupt request bit will be set to “1.”
   ●When the one-shot pulse mode is selected after reset
   ●When the operating mode is switched from the timer mode to the one-shot pulse mode
   ●When the operating mode is switched from the event counter mode to the one-shot pulse mode

   Accordingly, when using a timer Ai interrupt (interrupt request bit), be sure to clear the timer Ai interrupt request bit to “0” after the above setting.

7.6 Pulse width modulation (PWM) mode

In this mode, the timer continuously outputs pulses which have an arbitrary width. Table 7.6.1 lists the specifications of the PWM mode. Figure 7.6.1 shows the structure of the timer Ai register, and Figure 7.6.2 shows the structure of timer Ai mode register in the PWM mode.

Table 7.6.1 Specifications of PWM mode

Item	Specifications
Count source f_i	f_1, f_2, f_{16}, f_{64}, f_{512}, or f_{4096}
Count operation	● Countdown (operating as an 8-bit or 16-bit pulse width modulator) ● Reload register's contents are reloaded at rising edge of PWM pulse, and counting continues. ● A trigger generated during counting does not affect the counting.
PWM period/"H" level width	<16-bit pulse width modulator> Period $= \frac{(2^{16}-1)}{f_i}$ [s] "H" level width $= \frac{n}{f_i}$ [s] n : Timer Ai register's set value <8-bit pulse width modulator> Period $= \frac{(m+1)(2^8-1)}{f_i}$ [s] "H" level width $= \frac{n(m+1)}{f_i}$ [s] m : Timer Ai register's low-order 8 bits' set value n : Timer Ai register's high-order 8 bits' set value
Count start condition	● When a trigger is generated. **(Note)** ● Internal or external trigger can be selected by software.
Count stop condition	When the count start bit is cleared to "0."
Interrupt request occurrence timing	At falling edge of PWM pulse
TAi_{IN} pin's function	Programmable I/O port pin or trigger input pin
TAi_{OUT} pin's function	PWM pulse output
Read from timer Ai register	An undefined value is read out.
Write to timer Ai register	● While counting is stopped When a value is written to the timer Ai register, it is written to both of the reload register and counter. ● While counting is in progress When a value is written to the timer Ai register, it is written only to the reload register. (Transferred to the counter at the next reload time.)

Note: The trigger is generated with the count start bit = "1."

<When operating as a 16-bit pulse width modulator>

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 ... (b8) b0 | b7 ... b0

Bit	Function	At reset	R/W
15 to 0	Any value in the range from "0000_{16}" to "$FFFE_{16}$" can be set. Assuming that the set value = n, the "H" level width of the PWM pulse which is output from the TAiOUT pin is expressed as follows : $\frac{n}{f_i}$ (PWM pulse period = $\frac{2^{16}-1}{f_i}$)	Undefined	WO

f_i: Frequency of count source
Note: Use the **MOVM** or **STA(STAD)** instruction for writing to this register.
Writing to this register must be performed in a unit of 16 bits.

<When operating as an 8-bit pulse width modulator>

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 ... (b8) b0 | b7 ... b0

Bit	Function	At reset	R/W
7 to 0	Any value in the range from "00_{16}" to "FF_{16}" can be set. Assuming that the set value = m, the period of the PWM pulse which is output from the TAiOUT pin is expressed as follows: $\frac{(m + 1)(2^8 - 1)}{f_i}$	Undefined	WO
15 to 8	Any value in the range from "00_{16}" to "FF_{16}" can be set. Assuming that the set value = n, the "H" level width of the PWM pulse which is output from the TAiOUT pin is expressed as follows: $\frac{n(m + 1)}{f_i}$	Undefined	WO

f_i: Frequency of count source
Note: Use the **MOVM** or **STA(STAD)** instruction for writing to this register.
Writing to this register must be performed in a unit of 16 bits.

Fig. 7.6.1 Structures of timer Ai register in PWM mode

7.6 Pulse width modulation (PWM) mode

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

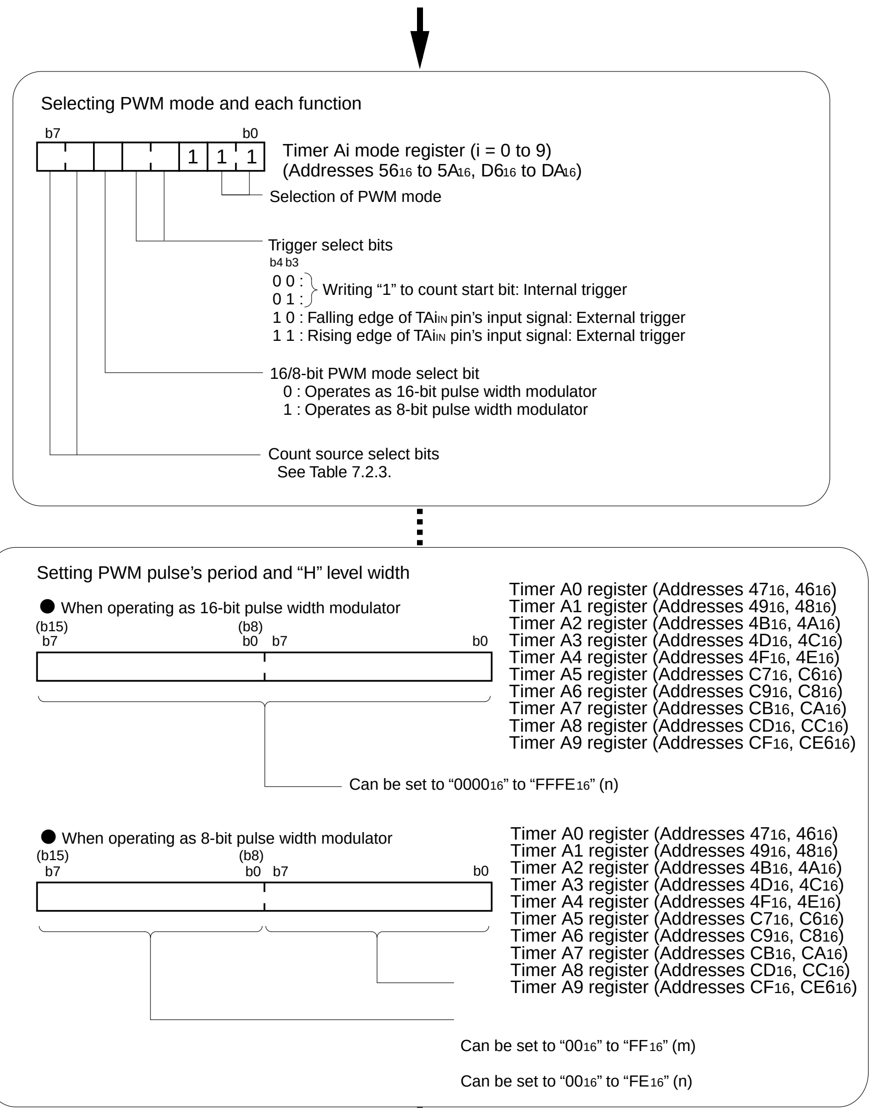
b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits	b1 b0 1 1 : PWM mode	0	RW
1			0	RW
2	Fix this bit to “1” in PWM mode.		0	RW
3	Trigger select bits	b4 b3 0 0 : } Writing “1” to count start bit 0 1 : } (TAi_{IN} pin functions as a programmable I/O port pin.) 1 0 : Falling edge of TAi_{IN} pin’s input signal 1 1 : Rising edge of TAi_{IN} pin’s input signal	0	RW
4			0	RW
5	16/8-bit PWM mode select bit	0 : 16-bit pulse width modulator 1 : 8-bit pulse width modulator	0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 7.6.2 Structures of timer Ai mode register in PWM mode

7.6.1 Setting for PWM mode

Figures 7.6.3 and 7.6.4 show an initial setting example for registers relevant to the PWM mode.
Note that when using interrupts, set up to enable the interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."**



Note. When operating as 8-bit pulse width modulator

$$\text{Period} = \frac{(m+1)(2^8-1)}{f_i} \quad (f_i : \text{Frequency of count source})$$

$$\text{"H" level width} = \frac{n(m+1)}{f_i}$$

However, if n = "00_{16}", the pulse width modulator does not operate and the TAi_{OUT} pin outputs "L" level. At this time, no timer Ai interrupt request occurs.

Note. When operating as 16-bit pulse width modulator

$$\text{Period} = \frac{2^{16}-1}{f_i} \quad (f_i : \text{Frequency of count source})$$

$$\text{"H" level width} = \frac{n}{f_i}$$

However, if n = "0000_{16}", the pulse width modulator does not operate and the TAi_{OUT} pin outputs "L" level. At this time, no timer Ai interrupt request occurs.

Continued to Figure 7.6.4
on the next page

Fig. 7.6.3 Initial setting example for registers related to PWM mode (1)

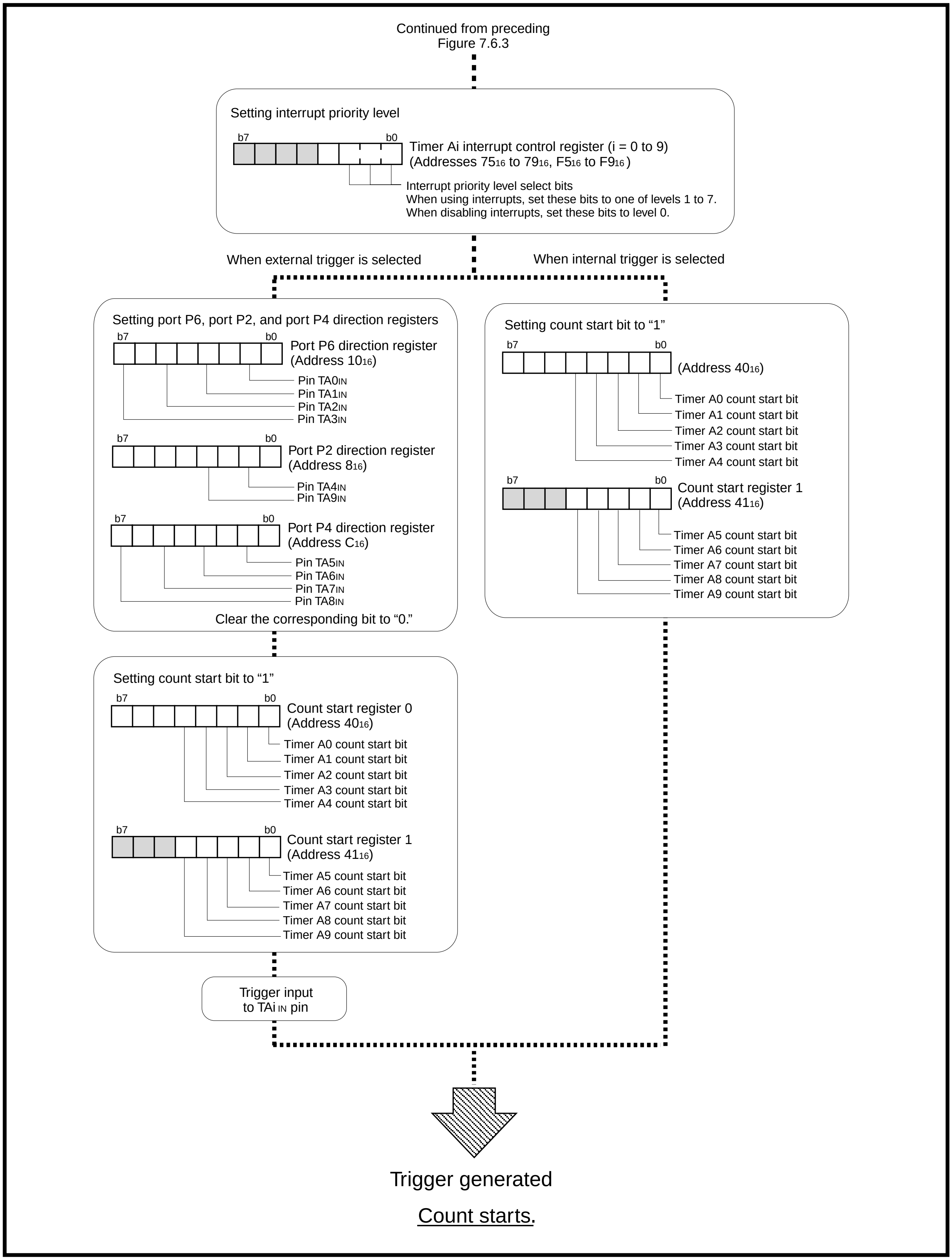


Fig. 7.6.4 Initial setting example for registers related to PWM mode (2)

7.6.2 Trigger

When a trigger is generated, the TAi_{OUT} pin starts to output PWM pulses. An internal or an external trigger can be selected as that trigger.
An internal trigger is selected when the trigger select bits (bits 4 and 3 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16}) are "00_2" or "01_2"; an external trigger is selected when these bits are "10_2" or "11_2."
A trigger generated during PWM pulse output is invalid, and it does not affect the pulse output operation.

(1) When selecting internal trigger

A trigger is generated when "1" is written to the count start bit (addresses 40_{16}, 41_{16}).

(2) When selecting external trigger

A trigger is generated at the falling edge of the TAi_{IN} pin's input signal when bit 3 at addresses 56_{16} to $5A_{16}$, $D6_{16}$ to DA_{16} is "0," or at its rising edge when bit 3 is "1." However, the trigger input is acceptableonly when the count start bit is "1."
When using an external trigger, set the port P6, port P2, and port P4 direction registers' bits which correspond to the TAi_{IN} pins for the input mode.

7.6.3 Operation in PWM mode

① When the PWM mode is selected with the operating mode select bits, the TAi_{OUT} pin outputs “L” level.
② When a trigger is generated, the counter (pulse width modulator) starts counting and the TAi_{OUT} pin outputs a PWM pulse (**Notes 1 and 2**).
③ The timer Ai interrupt request bit is set to “1” each time the PWM pulse level goes from “H” to “L.” The interrupt request bit remains set to “1” until the interrupt request is accepted or until the interrupt request bit is cleared to “0” by software.
④ Each time a PWM pulse has been output for one period, the reload register’s contents are reloaded and the counter continues counting.

The following explains operations of the pulse width modulator.

(1) 16-bit pulse width modulator
When the 16/8-bit PWM mode select bit is cleared to “0,” the counter operates as a 16-bit pulse width modulator. Figures 7.6.5 and 7.6.6 show operation examples of the 16-bit pulse width modulator.

(2) 8-bit pulse width modulator
When the 16/8-bit PWM mode select bit is set to “1,” the counter is divided into 8-bit halves. Then, the high-order 8 bits operate as an 8-bit pulse width modulator, and the low-order 8 bits operate as an 8-bit prescaler. Figures 7.6.7 and 7.6.8 show operation examples of the 8-bit pulse width modulator.

Notes 1: If a value “0000_{16}” is set into the timer Ai register when the counter operates as a 16-bit pulse width modulator, the pulse width modulator does not operate and the output from the TAi_{OUT} pin remains “L” level. The timer Ai interrupt request does not occur. Similarly, if a value “00_{16}” is set into the high-order 8 bits of the timer Ai register when the counter operates as an 8-bit pulse width modulator, the same is performed.

2: When the counter operates as an 8-bit pulse width modulator, after a trigger is generated, the TAi_{OUT} pin outputs “L” level for a period of $(1 / f_i) \times (m + 1) \times (n + 1)$. After that, the PWM pulse output will start.

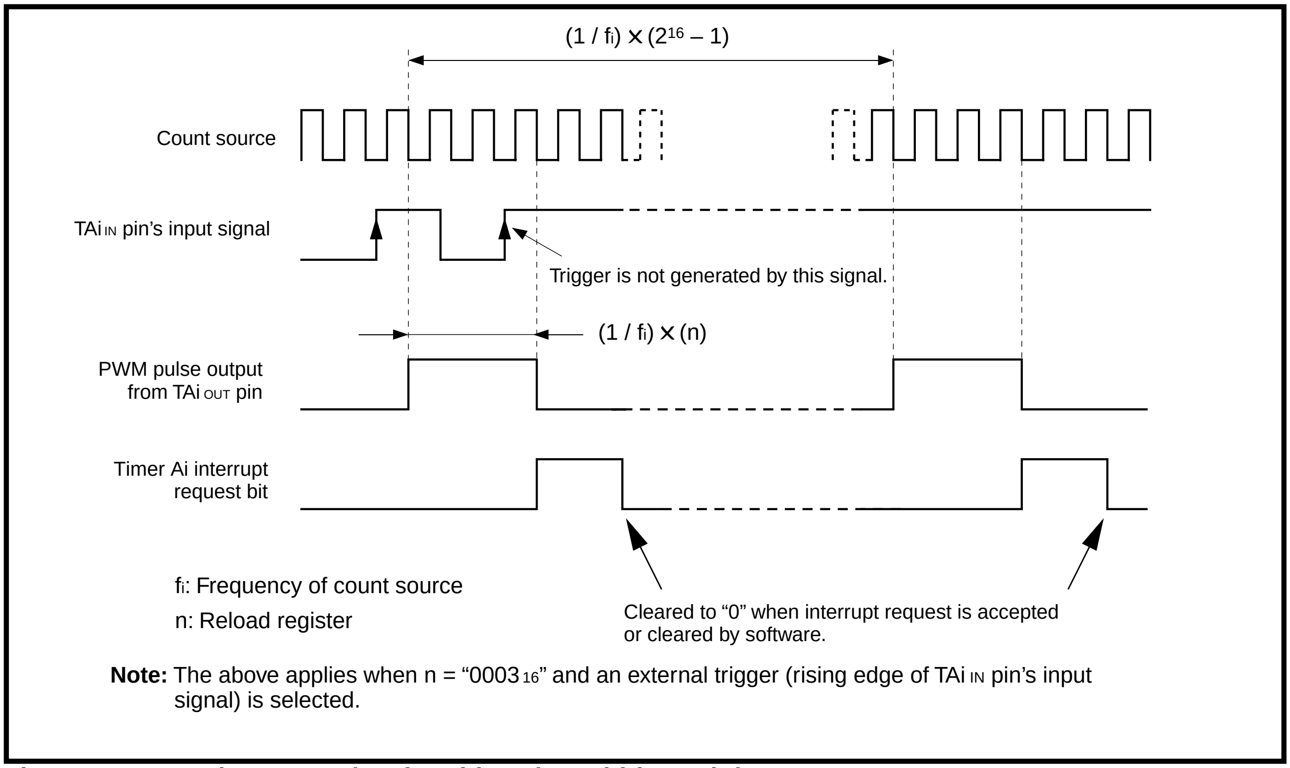


Fig. 7.6.5 Operation example of 16-bit pulse width modulator

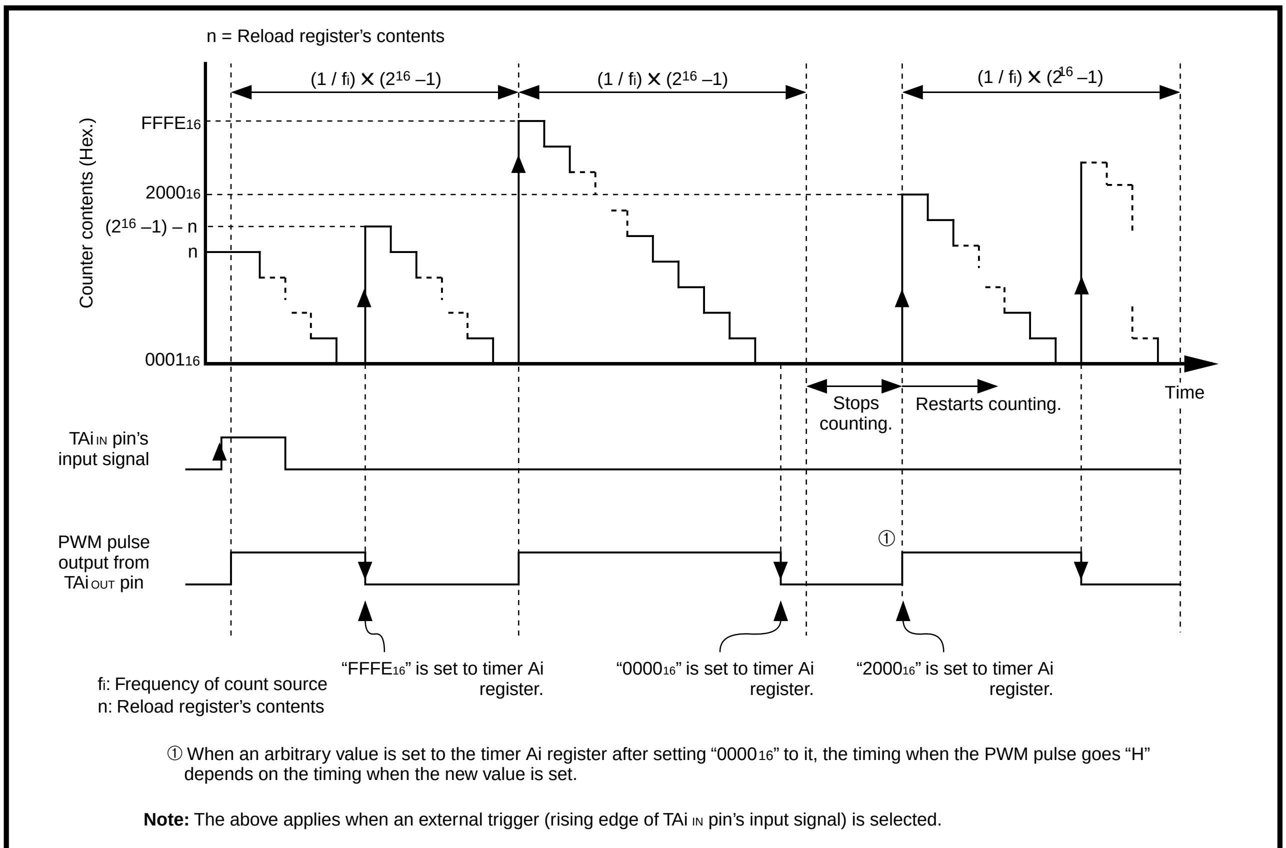
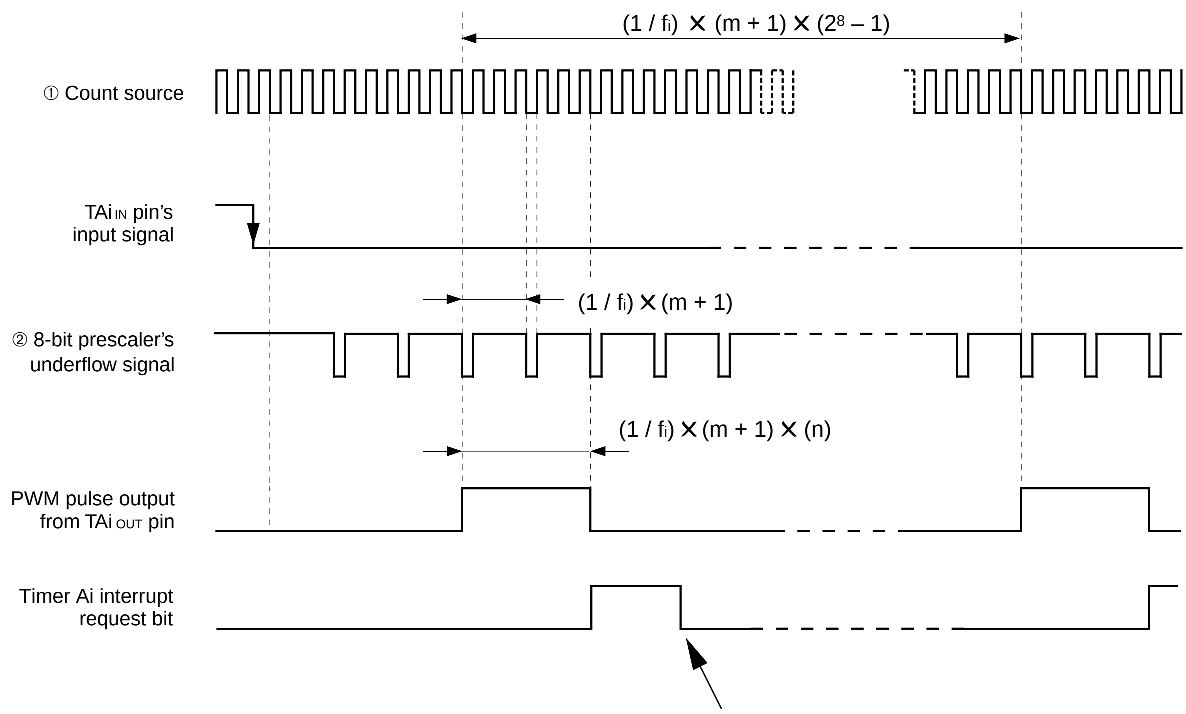


Fig. 7.6.6 Operation example of 16-bit pulse width modulator (when counter value is updated during pulse output)



f_i: Frequency of count source
n: Reload register's high-order 8 bits
m: Reload register's low-order 8 bits

① The 8-bit prescaler counts the count source.
② The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.

Note: The above applies when n = "02_{16}", m = "02_{16}", and an external trigger (falling edge of TAi_{IN} pin's input signal) is selected.

Fig. 7.6.7 Operation example of 8-bit pulse width modulator

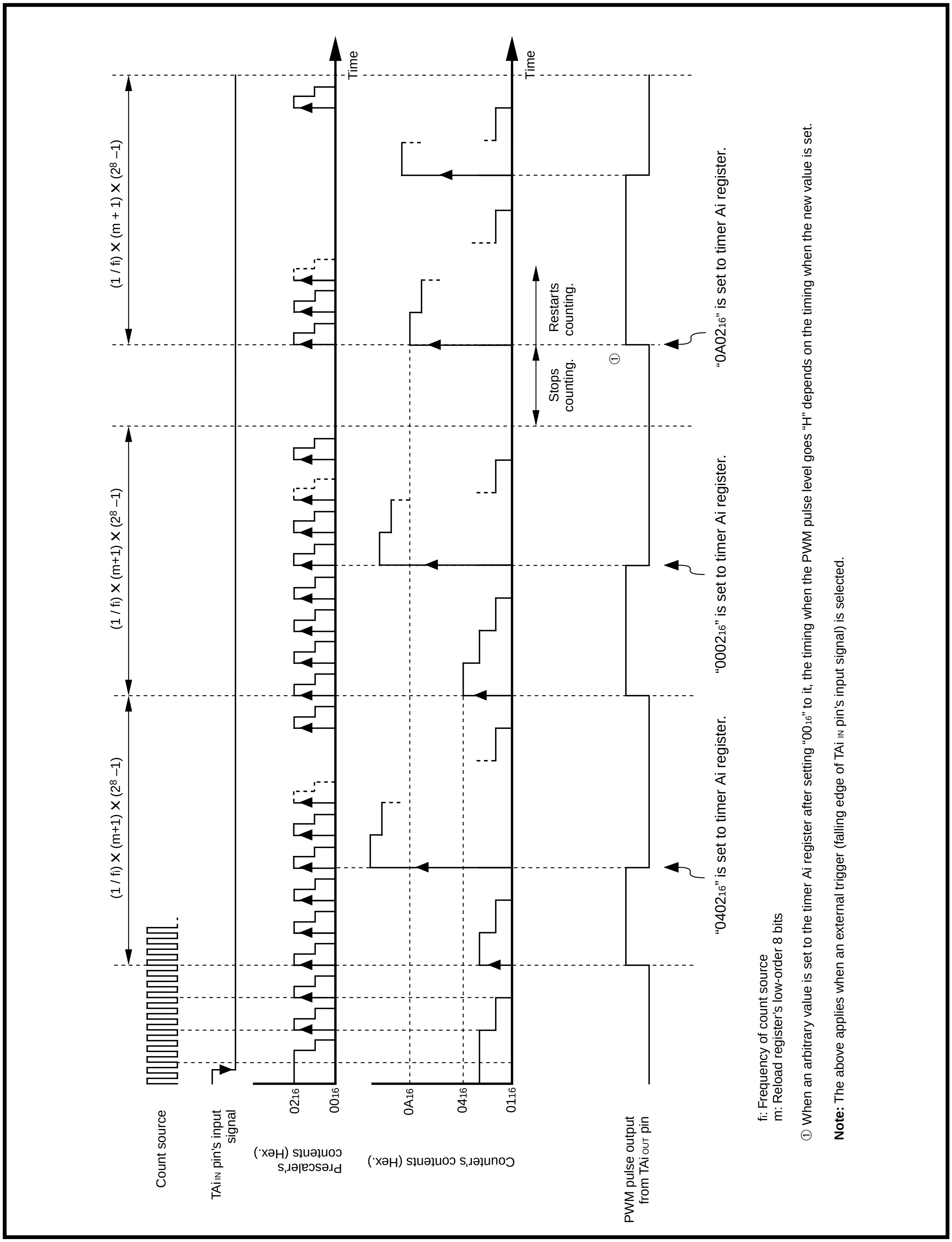


Fig. 7.6.8 Operation example of 8-bit pulse width modulator (when counter value is updated during pulse output)

[Precautions for pulse width modulation (PWM) mode]

1. If the count start bit is cleared to "0" during PWM pulse output, the counter stops counting. If the TAi_{OUT} pin outputs "H" level at that time, the output level will become "L" and the timer Ai interrupt request bit will be set to "1." When the TAi_{OUT} pin outputs "L" level at that time, the output level will not change and no timer Ai interrupt request will occur.

2. When the timer's operating mode is set by one of the following procedures, the timer Ai interrupt request bit is set to "1."
   - ●When the PWM mode is selected after reset
   - ●When the operating mode is switched from the timer mode to the PWM mode
   - ●When the operating mode is switched from the event counter mode to the PWM mode

   Accordingly, when using a timer Ai interrupt (interrupt request bit), be sure to clear the timer Ai interrupt request bit to "0" after the above setting.

CHAPTER 8

TIMER B

8.1 Overview

Timer B consists of three counters (timers B0 to B2) each equipped with a 16-bit reload function. Timers B0 to B2 have identical functions and operate independently of one other.
Timer Bi (i = 0 to 2) has three operating modes listed below.

(1) Timer mode
The timer counts an internally generated count source.

(2) Event counter mode
The timer counts an external signal.

(3) Pulse period/Pulse width measurement mode
The timer measures an external signal's pulse period or pulse width. In this mode, the following count types are available:
• Count clear type
• Free-run type

8.2 Block description

Figure 8.2.1 shows the block diagram of timer B. Explanation of registers relevant to timer B is described below.

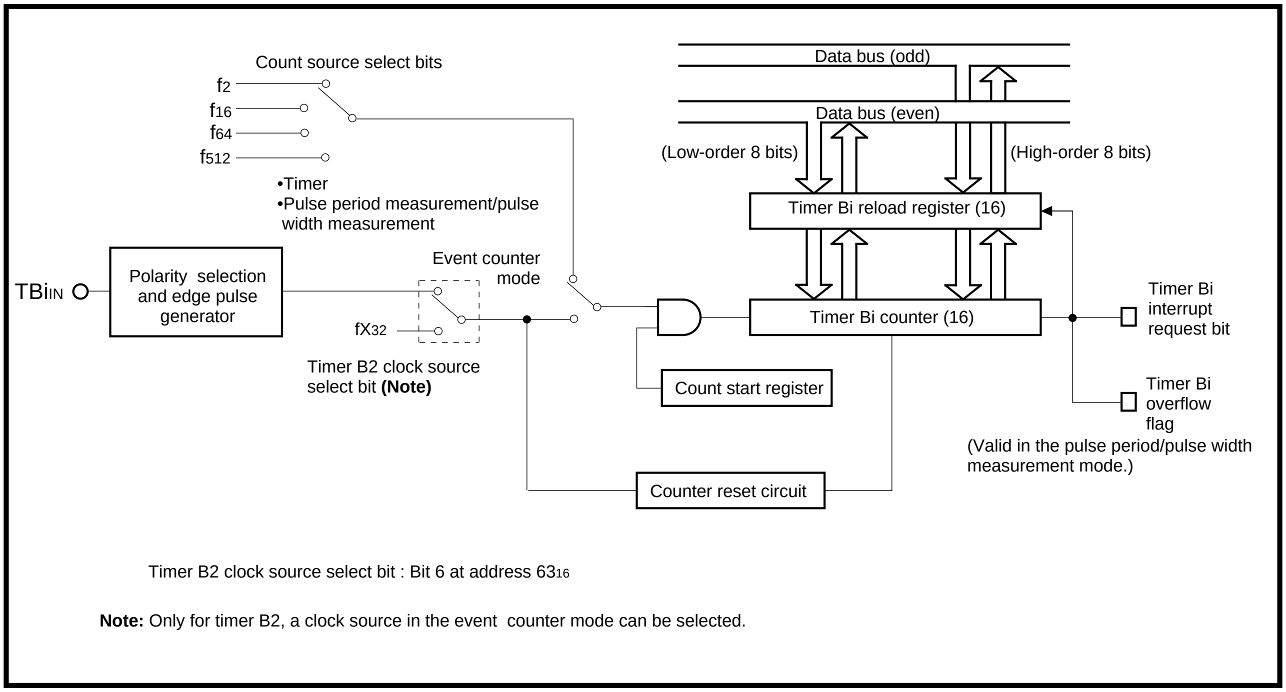


Fig. 8.2.1 Block diagram of timer B

8.2.1 Counter and Reload register (timer Bi register)

Each of timer Bi counter and reload register consists of 16 bits and has the following functions.

(1) Functions in timer mode and event counter mode

Countdown in the counter is performed each time the count source is input. The reload register is used to store the initial value of the counter. When a counter underflow occurs, the reload register's contents are reloaded into the counter.

A value is set to the counter and reload register by writing the value to the timer Bi register.

Table 8.2.1 lists the memory assignment of the timer Bi register.

The value written into the timer Bi register while counting is not in progress is set to the counter and reload register. The value written into the timer Bi register while counting is in progress is set only to the reload register. In this case, the reload register's updated contents are transferred to the counter at the next underflow. The counter value is read out by reading out the timer Bi register.

Note: When reading from or writing to the timer Bi register, perform it in a unit of 16 bits. For more information about the value obtained by reading the timer Bi register, refer to sections **"[Precautions for timer mode]"** and **"[Precautions for event counter mode]."**

(2) Functions in pulse period/pulse width measurement mode

Countup in the counter is performed each time the count source is input. The reload register is used to retain the pulse period or pulse width measurement result. When a valid edge is input to the TBi_{IN} pin, the counter value is transferred to the reload register. In this mode, the value obtained by reading the timer Bi register is the reload register's contents, so that the measurement result is obtained.

By using the count-type select bit (bit 4 at addresses $5B_{16}$ to $5D_{16}$), the count type can be selected from the counter clear type and free-run type.

The operation of the counter after the counter value is transferred to the reload register is as follows;

- In the case of the counter clear type, the counter value becomes "0000_{16}"; and counting continues.
- In the case of the free-run type, the counter value does not become "0000_{16}"; and counting continues with this counter value kept.

Note: When reading from the timer Bi register, perform it in a unit of 16 bits.

Table 8.2.1 Memory assignment of timer Bi registers

Timer Bi register	High-order byte	Low-order byte
Timer B0 register	Address 51_{16}	Address 50_{16}
Timer B1 register	Address 53_{16}	Address 52_{16}
Timer B2 register	Address 55_{16}	Address 54_{16}

Note: At reset, the contents of the timer Bi register are undefined.

8.2.2 Count start register

This register is used to start and stop counting. One bit of this register corresponds to one timer. (This is the one-to-one relationship.) Figure 8.2.2 shows the structure of the count start register 0.

Count start register 0 (Address 40_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Timer A0 count start bit	0 : Stop counting 1 : Start counting	0	RW
1	Timer A1 count start bit		0	RW
2	Timer A2 count start bit		0	RW
3	Timer A3 count start bit		0	RW
4	Timer A4 count start bit		0	RW
5	Timer B0 count start bit		0	RW
6	Timer B1 count start bit		0	RW
7	Timer B2 count start bit		0	RW

Fig. 8.2.2 Structure of count start register 0

8.2.3 Timer Bi mode register

Figure 8.2.3 shows the structure of the timer Bi mode register. The operating mode select bits are used to select the operating mode of timer Bi. Bits 2 to 7 have different functions according to the operating mode. These bits are described in the paragraph of each operating mode.

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/Pulse width measurement mode 1 1 : Do not select.	0	RW
1			0	RW
2	These bits have different functions according to the operating mode.		0	RW
3			0	RW
4			0	RW
5			Undefined	RO **(Note)**
6			0	RW
7			0	RW

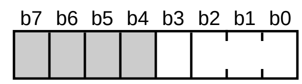
Note: Bit 5 is invalid in the timer and event counter modes; its value is undefined at reading.

Fig. 8.2.3 Structure of timer Bi mode register

8.2.4 Timer Bi interrupt control register

Figure 8.2.4 shows the structure of the timer Bi interrupt control register. For details about interrupts, refer to **"CHAPTER 6. INTERRUPTS."**

Timer Bi interrupt control register (i = 0 to 2) (Addresses $7A_{16}$ to $7C_{16}$)



Bit	Bit name	Function	At reset	R/W
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt requested 1 : Interrupt requested	0	RW **(Note)**
7 to 4	Nothing is assigned.		Undefined	—

Note: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

Fig. 8.2.4 Structure of timer Bi interrupt control register

(1) Interrupt priority level select bits (bits 2 to 0)

These bits are used to select a timer Bi interrupt's priority level. When using timer Bi interrupts, select the priority level from levels 1 through 7. When a timer Bi interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL), so that the requested interrupt is enabled only when its priority level is higher than the IPL. (However, this applies when the interrupt disable bit (I) = "0.") To disable timer Bi interrupts, set these bits to "000_2" (level 0).

(2) Interrupt request bit (bit 3)

This bit is set to "1" when a timer Bi interrupt request occurs. This bit is automatically cleared to "0" when the timer Bi interrupt request is accepted. This bit can be set to "1" or cleared to "0" by software.

8.2 Block description

8.2.5 Port P2 direction register, Port P5 direction register

The input pins of timer Bi are multiplexed with port P5 pins. By using the $TB0_{IN}/TB1_{IN}/TB2_{IN}$ pin select bit (see Figure 8.2.5.), pin $TB0_{IN}/TB1_{IN}/TB2_{IN}$ can be allocated to the corresponding port P2 pin.
When using pins $P5_5(P2_4)/TB0_{IN}$, $P5_6(P2_5)/TB1_{IN}$, $P5_7(P2_6)/TB2_{IN}$ as timer Bi's input pins, be sure to clear the corresponding bits of the port direction register, which is multiplexed, to "0" in order to set these pins to the input mode. (See Figure 8.2.6.)

Port P2 pin function control register (Address AE_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0							

Bit	Bit name	Function	At reset	R/W
0	Pin $TB0_{IN}$ select bit	0 : Allocate pin $TB0_{IN}$ to $P5_5$. 1 : Allocate pin $TB0_{IN}$ to $P2_4$.	0	RW
1	Pin $TB1_{IN}$ select bit	0 : Allocate pin $TB1_{IN}$ to $P5_6$. 1 : Allocate pin $TB1_{IN}$ to $P2_5$.	0	RW
2	Pin $TB2_{IN}$ select bit	0 : Allocate pin $TB2_{IN}$ to $P5_7$. 1 : Allocate pin $TB2_{IN}$ to $P2_6$.	0	RW
6 to 3	Nothing is assigned.		Undefined	—
7	Fix this bit to "0."		0	RW

Fig. 8.2.5 Structure of port P2 pin function control register

Port P5 direction register (Address D_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin	Functions	At reset	R/W
0	Nothing is assigned.		Undefined	–
1	Pin $\overline{INT_1}$	0 : Input mode 1 : Output mode	0	RW
2	Pin $\overline{INT_2}$/RTP_{TRG1}		0	RW
3	Pin $\overline{INT_3}$/RTP_{TRG0}		0	RW
4	Nothing is assigned.		Undefined	–
5	Pin $TB0_{IN}$ (Pin $\overline{INT_5}$/IDW) **(Note 1)**	0 : Input mode 1 : Output mode When using this pin as timer Bi's input pin, be sure to clear the corresponding bit to "0."	0	RW
6	Pin $TB1_{IN}$ (Pin $\overline{INT_6}$/IDV) **(Note 2)**		0	RW
7	Pin $TB2_{IN}$ (Pin $\overline{INT_7}$/IDU) **(Note 3)**		0	RW

Notes 1: This applies when the $TB0_{IN}$ pin select bit (bit 0 at address AE_{16}) = 0.
2: This applies when the $TB1_{IN}$ pin select bit (bit 1 at address AE_{16}) = 0.
3: This applies when the $TB2_{IN}$ pin select bit (bit 2 at address AE_{16}) = 0.
4: The pins in ( ) are I/O pins of other internal peripheral devices, which are multiplexed with the corresponding port P5 pins.

Port P2 direction register (Address 8_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin	Functions	At reset	R/W
0	Pin $TA4_{OUT}$	0 : Input mode 1 : Output mode When using this pin as timer Bi's input pin, be sure to clear the corresponding bit to "0."	0	RW
1	Pin $TA4_{IN}$		0	RW
2	Pin $TA9_{OUT}$		0	RW
3	Pin $TA9_{IN}$		0	RW
4	Pin $TB0_{IN}$ **(Note 1)**		0	RW
5	Pin $TB1_{IN}$ **(Note 2)**		0	RW
6	Pin $TB2_{IN}$ **(Note 3)**		0	RW
7	Pin $P2_7$		0	RW

Notes 1: This applies when the $TB0_{IN}$ pin select bit (bit 0 at address AE_{16}) = 1.
2: This applies when the $TB1_{IN}$ pin select bit (bit 1 at address AE_{16}) = 1.
3: This applies when the $TB2_{IN}$ pin select bit (bit 2 at address AE_{16}) = 1.

Fig. 8.2.6 Relationship between port P5 direction register, port P2 direction register, and timer Bi's input pins

8.2.6 Count source (in timer mode and pulse period/pulse width measurement mode)

In the timer mode and pulse period/pulse width measurement mode, the count source select bits (bits 6 and 7 at addresses $5B_{16}$ to $5D_{16}$) are used to select the count source (f_2, f_{16}, f_{64}, or f_{512}). (See Figures 8.3.1 and 8.5.1.)

8.3 Timer mode

In this mode, the timer counts an internally generated count source. Table 8.3.1 lists the specifications of the timer mode. Figure 8.3.1 shows the structures of the timer Bi register and timer Bi mode register in the timer mode.

Table 8.3.1 Specifications of timer mode

Item	Specifications
Count source f_i	f_2, f_{16}, f_{64}, or f_{512}
Count operation	•Countdown •When a counter underflow occurs, reload register's contents are re-loaded, and counting continues.
Division ratio	$\frac{1}{(n + 1)}$ n: Timer Bi register's set value
Count start condition	When the count start bit is set to "1."
Count stop condition	When the count start bit is cleared to "0."
Interrupt request occurrence timing	When a counter underflow occurs.
TBi_{IN} pin's function	Programmable I/O port pin
Read from timer Bi register	Counter value can be read out.
Write to timer Bi register	● While counting is stopped When a value is written to the timer Bi register, it is written to both of the reload register and counter. ● While counting is in progress When a value is written to the timer Bi register, it is written only to the reload register. (Transferred to the counter at the next reload timing.)

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

(b15) b7 … (b8) b0 b7 … b0

Bit	Function	At reset	R/W
15 to 0	Any value in the range from “0000_{16}” to “$FFFF_{16}$” can be set. Assuming that the set value = n, the counter divides the count source frequency by (n + 1). When reading, the register indicates the counter value.	Undefined	RW

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		X	X	X	X	0	0

Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits	b1 b0 0 0 : Timer mode	0	RW
1			0	RW
2	These bits are invalid in timer mode.		0	RW
3			0	RW
4			0	—
5	This bit is invalid in timer mode; its value is undefined at reading.		Undefined	RO
6	Count source select bits	b7 b6 0 0 : f_2 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW
7			0	RW

X : It may be either “0” or “1.”

Fig. 8.3.1 Structures of timer Bi register and timer Bi mode register in timer mode

8.3 Timer mode

8.3.1 Setting for timer mode

Figure 8.3.2 shows an initial setting example for registers relevant to the timer mode.
Note that when using interrupts, set up registers to enable the interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."**

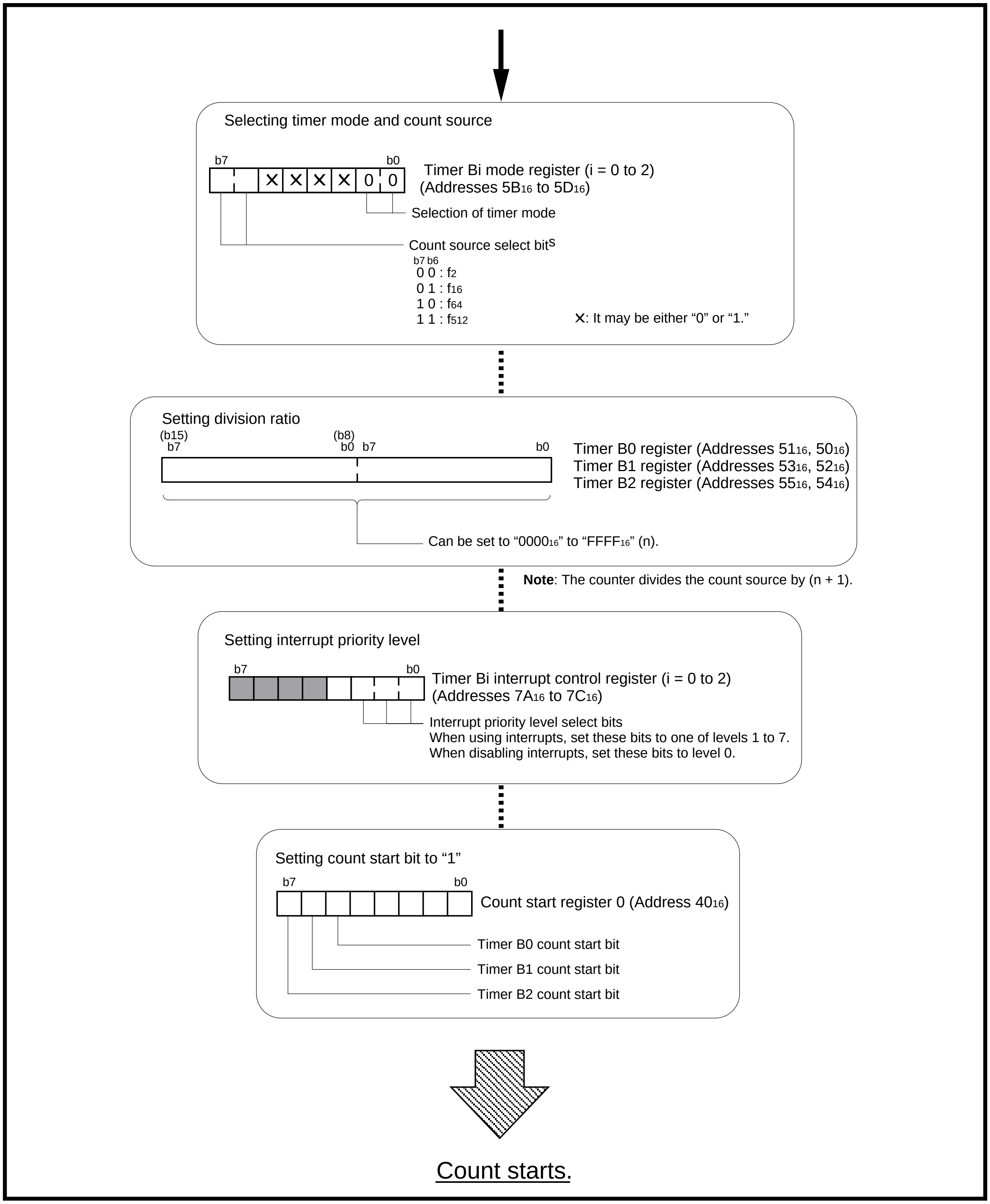


Fig. 8.3.2 Initial setting example for registers relevant to timer mode

8.3.2 Operation in timer mode

① When the count start bit is set to “1,” the counter starts counting of the count source.
② When a counter underflow occurs, the reload register’s contents are reloaded and counting continues.
③ The timer Bi interrupt request bit is set to “1” at the counter underflow in ②. The interrupt request bit remains set to “1” until the interrupt request is accepted or until the interrupt request bit is cleared to “0” by software.

Figure 8.3.3 shows an example of operation in the timer mode.

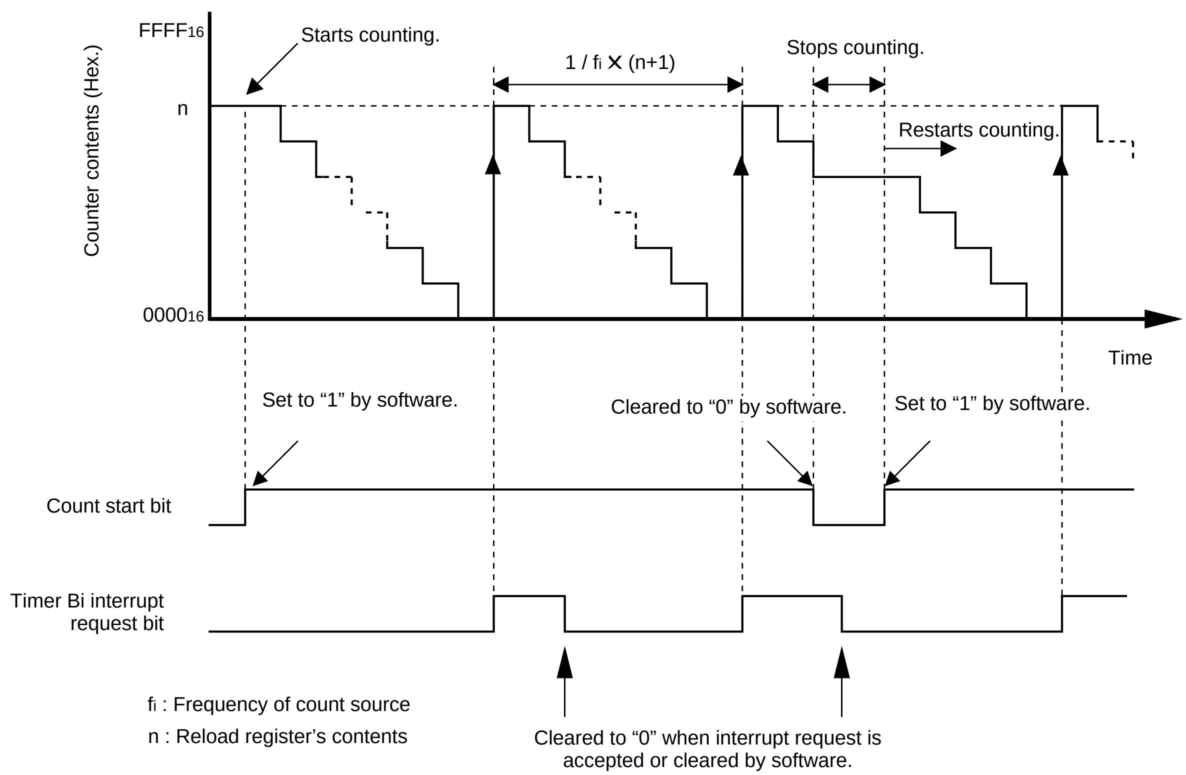


Fig. 8.3.3 Example of operation in timer mode

[Precautions for timer mode]

While counting is in progress, by reading the timer Bi register, the counter value can be read out at arbitrary timing. However, if the timer Bi register is read at the reload timing shown in Figure 8.3.4, the value "$FFFF_{16}$" is read out. If reading is performed in the period from when a value is set into the timer Bi register with the counter stopped until the counter starts counting, the set value is correctly read out.

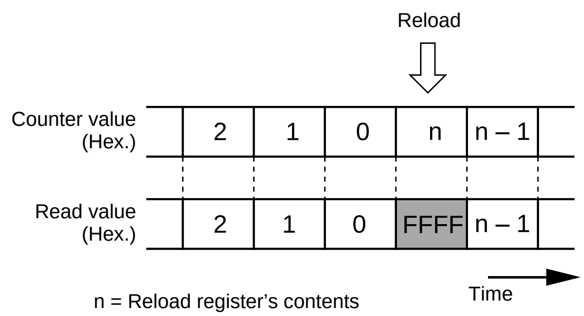


Fig. 8.3.4 Reading timer Bi register

8.4 Event counter mode

In this mode, the timer counts an external signal. Table 8.4.1 lists the specifications of the event counter mode. Figure 8.4.1 shows the structures of the timer Bi register and the timer Bi mode register in the event counter mode.

Table 8.4.1 Specifications of event counter mode

Item	Specifications
Count source	•External signal input to the TBi_{IN} pin, or fX_{32} **(Note 1)** •The count source's valid edge can be selected from the falling edge, the rising edge, and both of the falling and rising edges by software.
Count operation	•Countdown •When a counter underflow occurs, reload register's contents are reloaded, and counting continues.
Division ratio	$\frac{1}{(n + 1)}$ n: Timer Bi register's set value
Count start condition	When the count start bit is set to "1."
Count stop condition	When the count start bit is cleared to "0."
Interrupt request occurrence timing	When the counter underflow occurs.
TBi_{IN} pin's function	Count source input pin **(Note 2)**
Read from timer Bi register	Counter value can be read out.
Write to timer Bi register	● While counting is stopped When a value is written to the timer Bi register, it is written to both of the reload register and counter. ● While counting is in progress When a value is written to the timer Bi register, it is written only to the reload register. (Transferred to the counter at the next reload timing.)

Notes 1: Only for timer B2, fX_{32} can be selected.

2: When fX_{32} is selected as the count source in timer B2, the $TB2_{IN}$ pin can be used as a programmable I/O port pin or as I/O pins of other internal peripheral devices, which are multiplexed.

8.4 Event counter mode

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

(b15) b7 (b8) b0 b7 b0

Bit	Function	At reset	R/W
15 to 0	Any value in the range from "0000_{16}" to "$FFFF_{16}$" can be set. Assuming that the set value = n, the counter divides the count source frequency by (n + 1). When reading, the register indicates the counter value.	Undefined	RW

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X			0	1

Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits	b1 b0 0 1 : Event counter mode	0	RW
1			0	RW
2	Count polarity select bits	b3 b2 0 0 : Count at falling edge of external signal 0 1 : Count at rising edge of external signal 1 0 : Count at both falling and rising edges of external signal 1 1 : Do not select. **(Note)**	0	RW
3			0	RW
4	This bit is invalid in event counter mode.		0	—
5	This bit is invalid in event counter mode; its value is undefined at reading.		Undefined	RO
6	These bits are invalid in event counter mode.		0	RW
7			0	RW

X : It may be either "0" or "1."

Note: When the timer B2 clock source select bit (bit 6 at address 63_{16}) = "1," be sure to fix these bits to "01_2" (count at the rising edge of the external signal).

Fig. 8.4.1 Structures of timer Bi register and timer Bi mode register in event counter mode

8.4.1 Count source

For timer B2 in the event counter mode, a count source (an external signal into the $TB2_{IN}$ pin, or fX_{32}) can be selected by using the timer B2 clock source select bit. (See Figure 8.4.2.) Timers B0 and B1 count the external signals input to the $TB0_{IN}$ and $TB1_{IN}$ pins, respectively.

When fX_{32} is selected as the count source, the $TB2_{IN}$ pin serves as a programmable I/O port pin or as I/O pins of other internal peripheral devices, which are multiplexed.

Particular function select register 1 (Address 63_{16})

b7	b6	b5	b4	b3	b2	b1	b0
			0		0		

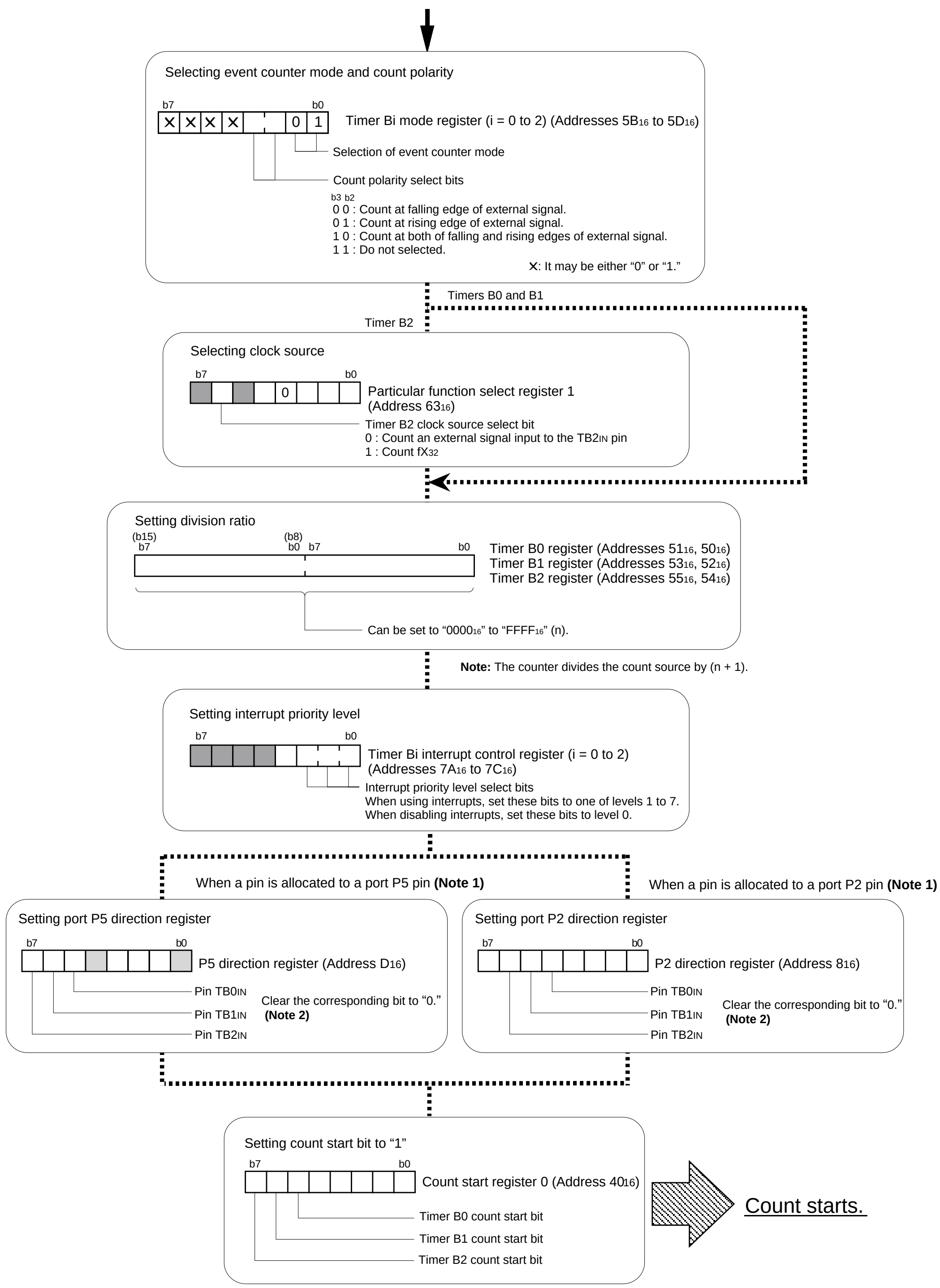
Bit	Bit name	Function	At reset	
0	STP-instruction-execution status bit	0 : Normal operation. 1 : During execution of STP instruction	**(Note 1)**	RW **(Note 2)**
1	WIT-instruction-execution status bit	0 : Normal operation. 1 : During execution of WIT instruction	**(Note 1)**	RW **(Note 2)**
2	Fix this bit to “0.”		0	RW
3	System clock stop select bit at WIT **(Note 3)**	0 : In the wait mode, system clock f_{sys} is active. 1 : In the wait mode, system clock f_{sys} is inactive.	0	RW
4	Fix this bit to “0.”		0	RW
5	The value is “0” at reading.		0	—
6	Timer B2 clock source select bit (Valid in event counter mode.) **(Note 4)**	0 : External signal input to the $TB2_{IN}$ pin is counted. 1 : fX_{32} is counted.	0	RW
7	The value is “0” at reading.		0	—

Notes 1: At power-on reset, this bit becomes “0.” At hardware reset or software reset, this bit retains the value just before reset.
2: Even when “1” is written, the bit status will not change.
3: Setting this bit to “1” must be performed just before execution of the WIT instruction. Also, after the wait state is terminated, this bit must be cleared to “0” immediately.
4: When using timer B2 in the pulse period/pulse width measurement mode, be sure to clear this bit to “0.”

Fig. 8.4.2 Structure of particular function select register 1

8.4.2 Setting for event counter mode

Figure 8.4.3 shows an initial setting example for registers relevant to the event counter mode.
Note that when using interrupts, set up to enable the interrupts. For details, refer to section **“CHAPTER 6. INTERRUPTS.”**



Notes 1: By using bits 0 to 2 of the port P2 pin function control register (address AE_{16}), be sure to set the pin allocation. (See Figure 8.2.5.)
2: When fX_{32} is selected as the count source in timer B2 (in other words, when bit 6 at address 63_{16} = 1), this setting is unnecessary.

Fig. 8.4.3 Initial setting example for registers relevant to event counter mode

8.4.3 Operation in event counter mode

① When the count start bit is set to “1,” the counter starts counting of the count source.
② When a counter underflow occurs, the reload register’s contents are reloaded, and counting continues.
③ The timer Bi interrupt request bit is set to “1” at the counter underflow in ②.
The interrupt request bit remains set to “1” until the interrupt request is accepted or until the interrupt request bit is cleared to “0” by software.

Figure 8.4.4 shows an example of operation in the event counter mode.

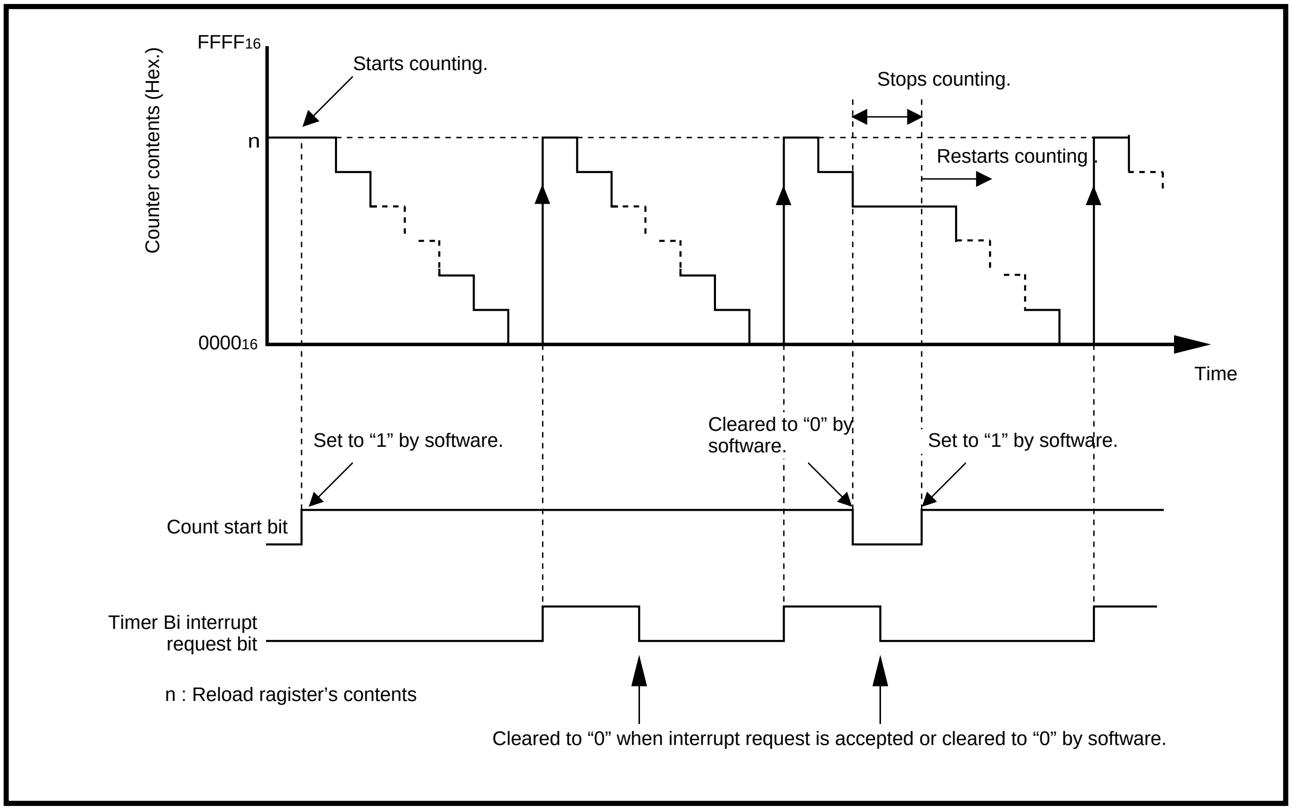


Fig. 8.4.4 Example of operation in event counter mode

[Precautions for event counter mode]

While counting is in progress, by reading the timer Bi register, the counter value can be read out at arbitrary timing. However, if the timer Bi register is read at the reload timing shown in Figure 8.4.5, a value "FFFF16" is read out. If reading is performed in the period from when a value is set into the timer Bi register with the counter stopped until the counter starts counting, the set value is correctly read out.

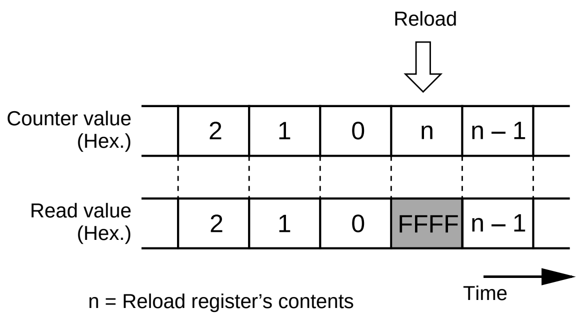


Fig. 8.4.5 Reading timer Bi register

8.5 Pulse period/Pulse width measurement mode

In this mode, the timer measures an external signal's pulse period or pulse width. Tables 8.5.1 and 8.5.2 list the specifications of the pulse period/pulse width measurement mode. Figure 8.5.1 shows the structures of the timer Bi register and timer Bi mode register in the pulse period/pulse width measurement mode.

(1) Pulse period measurement
The timer measures the pulse period of the external signal that is input to the TBi_{IN} pin.

(2) Pulse width measurement
The timer measures the pulse width ("L" level and "H" level widths) of the external signal that is input to the TBi_{IN} pin.

Table 8.5.1 Specifications of pulse period/pulse width measurement mode (when counter clear type is selected)

Item	Specifications
Count source f_i	f_2, f_{16}, f_{64}, or f_{512}
Count operation	● Countup ● Counter value is transferred to the reload register at valid edge of measurement pulse, and counting continues after clearing the counter value to "0000_{16}."
Count start condition	When the count start bit is set to "1."
Count stop condition	When the count start bit is cleared to "0."
Interrupt request occurrence timing	● When a valid edge of measurement pulse is input **(Note 1)**. ● When a counter overflow occurs (The timer Bi overflow flag is set to "1" simultaneously.)
TBi_{IN} pin's function	Measurement pulse input pin **(Note 2)**
Read from timer Bi register	The value obtained by reading the timer Bi register is the reload register's contents (Measurement result) **(Note 3)**.
Write to timer Bi register	Invalid

Timer Bi overflow flag: This bit is used to identify the source of an interrupt request occurrence.

Notes 1: No interrupt request occurs when the first valid edge is input after the counter starts counting.
2: When using timer B2, make sure that the timer B2 clock source select bit (see Figure 8.4.2.) to "0."
3: The value read out from the timer Bi register is undefined in the period after the counter starts counting until the second valid edge is input.

8.5 Pulse period/Pulse width measurement mode

Table 8.5.2 Specifications of pulse period/pulse width measurement mode (when free-run type is selected)

Item	Specifications
Count source f_i	f_2, f_{16}, f_{64}, or f_{512}
Count operation	● Countup ● Counter value is transferred to the reload register at valid edge of measurement pulse, and counting continues. ● When a counter overflow occurs, the timer Bi overflow flag is set to "1," and counting continues after clearing the counter value to "0000_{16}."
Count start condition	When the count start bit is set to "1."
Count stop condition	When the count start bit is cleared to "0."
Interrupt request occurrence timing	When a valid edge of measurement pulse is input **(Note 1)**.
TBi_{IN} pin's function	Measurement pulse input pin **(Note 2)**
Read from timer Bi register	The value obtained by reading the timer Bi register is the reload register's contents (Measurement result) **(Note 3)**.
Write to timer Bi register	Invalid

Timer Bi overflow flag: This bit is used to identify the source of an interrupt request occurrence.

Notes 1: No interrupt request occurs when the first valid edge is input after the counter starts counting.
2: When using timer B2, make sure that the timer B2 clock source select bit (see Figure 8.4.2.) = "0."
3: The value read out from the timer Bi register is undefined in the period after the counter starts counting until the second valid edge is input.

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

(b15) b7 … (b8) b0 b7 … b0

Bit	Function	At reset	R/W
15 to 0	The measurement result of pulse period or pulse width is read out.	Undefined	RO

Note: Reading from this register must be performed in a unit of 16 bits.

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0: b1 = 1, b0 = 0

Bit	Bit name	Function	At reset	R/W
0	Operating mode select bits	b1 b0 1 0 : Pulse period/Pulse width measurement mode	0	RW
1			0	RW
2	Measurement mode select bits	b3 b2 0 0 : Pulse period measurement (Interval between falling edges of measurement pulse) 0 1 : Pulse period measurement (Interval between rising edges of measurement pulse) 1 0 : Pulse width measurement (Interval from a falling edge to a rising edge, and from a rising edge to a falling edge of measurement pulse) 1 1 : Do not select.	0	RW
3			0	RW
4	Count-type select bit	0 : Counter clear type 1 : Free-run type	0	RW
5	Timer Bi overflow flag **(Note)**	0 : No overflow 1 : Overflowed	Undefined	RO
6	Count source select bits	b7 b6 0 0 : f_2 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW
7			0	RW

Note: The timer Bi overflow flag is cleared to "0" when a value is written to the timer Bi mode register with the count start bit = "1."
This flag cannot be set to "1" by software.

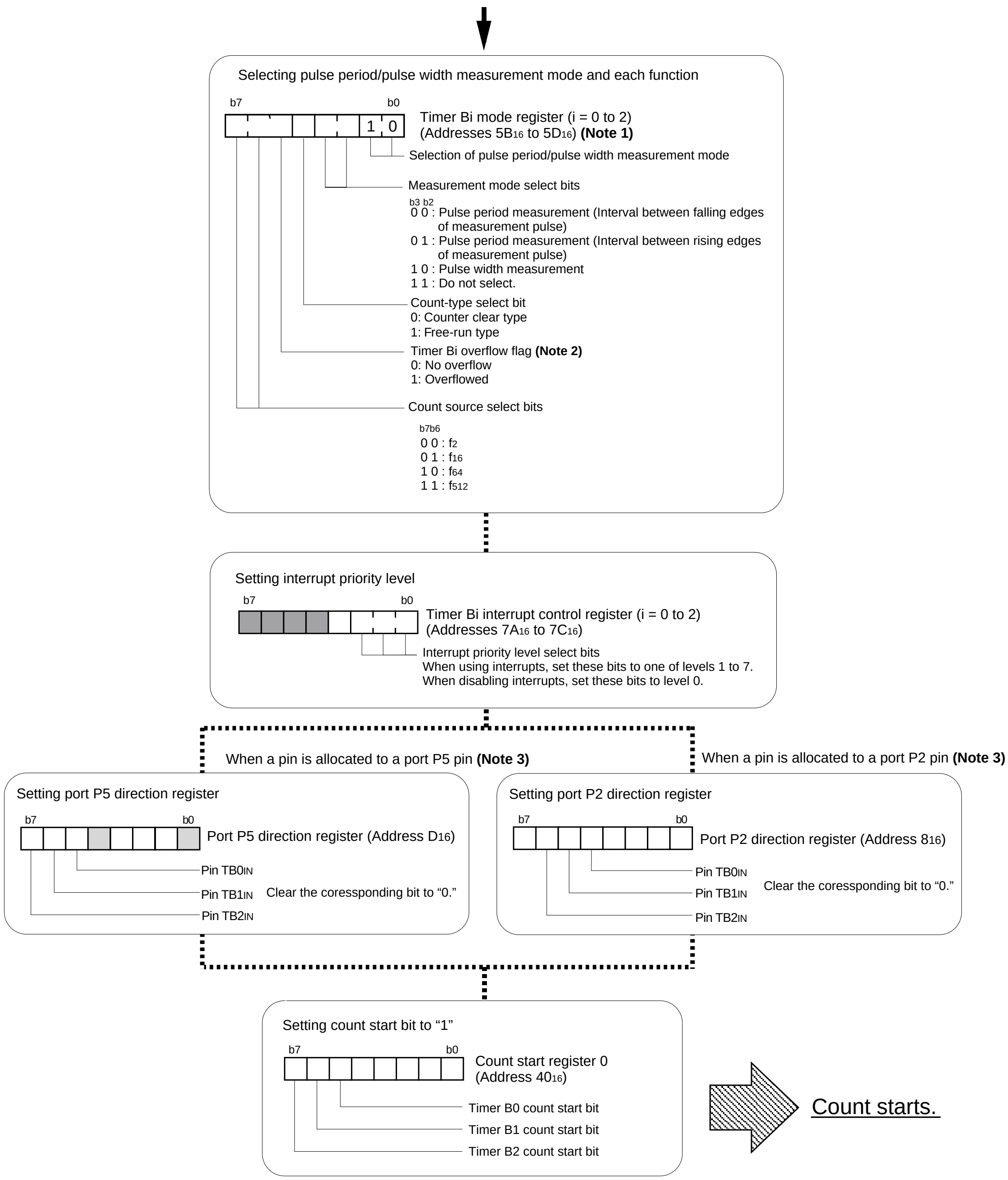
Fig. 8.5.1 Structures of timer Bi register and timer Bi mode register in pulse period/pulse width measurement mode

8.5 Pulse period/Pulse width measurement mode

8.5.1 Setting for pulse period/pulse width measurement mode

Figure 8.5.2 shows an initial setting example for registers relevant to the pulse period/pulse width measurement mode.

Note that when using interrupts, set up to enable the interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."**



Notes 1: When using timer B2, be sure to clear the timer B2 clock source select bit (See Figure 8.4.2.) to "0."

2: The timer Bi overflow flag is a read-only bit. This bit is undefined after reset. When a value is written to the timer Bi mode register with the count start bit = "1," this bit will be cleared to "0."

3: By using bits 0 to 2 of the port P2 pin function control register (address AE_{16}), be sure to set the pin allocation. (See Figure 8.2.5.)

Fig. 8.5.2 Initial setting example for registers relevant to pulse period/pulse width measurement mode

8.5.2 Operation in pulse period/pulse width measurement mode

■ When counter clear type is selected

① When the count start bit is set to "1," the counter starts counting of the count source.

② The counter value is transferred to the reload register when a valid edge of the measurement pulse is detected. (Refer to section **"(1) Pulse period/Pulse width measurement."**)

③ The counter value is cleared to "0000_{16}" after the transfer in ②, and the counter continues counting.

④ The timer Bi interrupt request bit is set to "1" when the counter value is cleared to "0000_{16}" in ③ (**Note**). The interrupt request bit remains set to "1" until the interrupt request is accepted or until the interrupt request bit is cleared to "0" by software.

⑤ The timer repeats operations ② to ④ above.

Note: No timer Bi interrupt request occurs when the first valid edge is input after the counter starts counting.

■ When free-run type is selected

① When the count start bit is set to "1," the counter starts counting of the count source.

② The counter value is transferred to the reload register when a valid edge of the measurement pulse is detected. (Refer to section **"(1) Pulse period/Pulse width measurement."**)

③ The timer Bi interrupt request bit is set to "1" after the transfer in ② **(Note)**. The interrupt request bit remains set to "1" until the interrupt request is accepted or until the interrupt request bit is cleared to "0" by software. The counter continues counting with the counter value kept.

④ When a counter overflow occurs, the timer Bi overflow flag is set to "1," and counting continues after clearing the counter value to "0000_{16} ." At this time, the timer Bi interrupt request bit does not change.

⑤ The timer repeats operations ② to ④ above.

Note: No timer Bi interrupt request occurs when the first valid edge is input after the counter starts counting.

(1) Pulse period/pulse width measurement

The measurement mode select bits (bits 3 and 2 at addresses $5B_{16}$ and $5D_{16}$) specify whether the pulse period of an external signal is measured or its pulse width is done. Table 8.5.3 lists the relationship between the measurement mode select bits and the pulse period/pulse width measurements. Make sure that the measurement pulse interval from the falling edge to the rising edge, and vice versa are two cycles of the count source or more. Additionally, use software to identify whether the measurement result indicates the "H" level width or the "L" level width.

Table 8.5.3 Relationship between measurement mode select bits and pulse period/pulse width measurements

b3	b2	Pulse period/Pulse width measurement	Measurement interval (Valid edges)
0	0	Pulse period measurement	From falling edge to falling edge (Falling edges)
0	1		From rising edge to rising edge (Rising edges)
1	0	Pulse width measurement	From falling edge to rising edge, and vice versa (Falling and rising edges)

(2) Timer Bi overflow flag

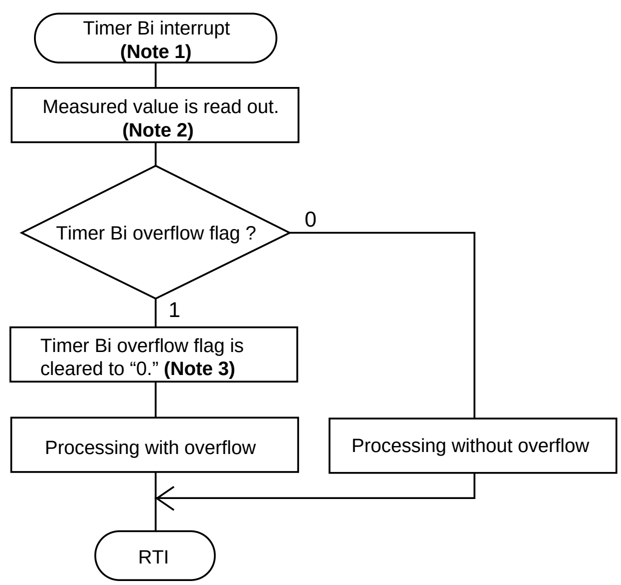
■ When counter clear type is selected

A timer Bi interrupt request occurs when a measurement pulse's valid edge is input or when a counter overflow occurs. The timer Bi overflow flag is used to identify the source of the interrupt request occurrence, that is, whether it is an overflow occurrence or a valid edge input.
The timer Bi overflow flag is set to "1" at an overflow occurrence. Accordingly, the source of the interrupt request occurrence is identified by checking the timer Bi overflow flag in the interrupt routine. When a value is written to the timer Bi mode register after the next count timing of the count source with the count start bit = "1," the timer Bi overflow flag will be cleared to "0".
The timer Bi overflow flag is a read-only bit.
Use the timer Bi interrupt request bit to detect the overflow timing. Do not use the timer Bi overflow flag for this detection.

■ When free-run type is selected

The timer Bi overflow flag is set to "1" at an overflow occurrence. (At this time, no timer Bi interrupt request is generated.) Accordingly, whether a counter overflow occurs between valid edges is identified by checking the timer Bi overflow flag in the interrupt routine owing to a valid edge input.
When a value is written to the timer Bi mode register after the next count timing of the count source with the count start bit = "1," the timer Bi overflow flag will be cleared to "0".
The timer Bi overflow flag is a read-only bit.

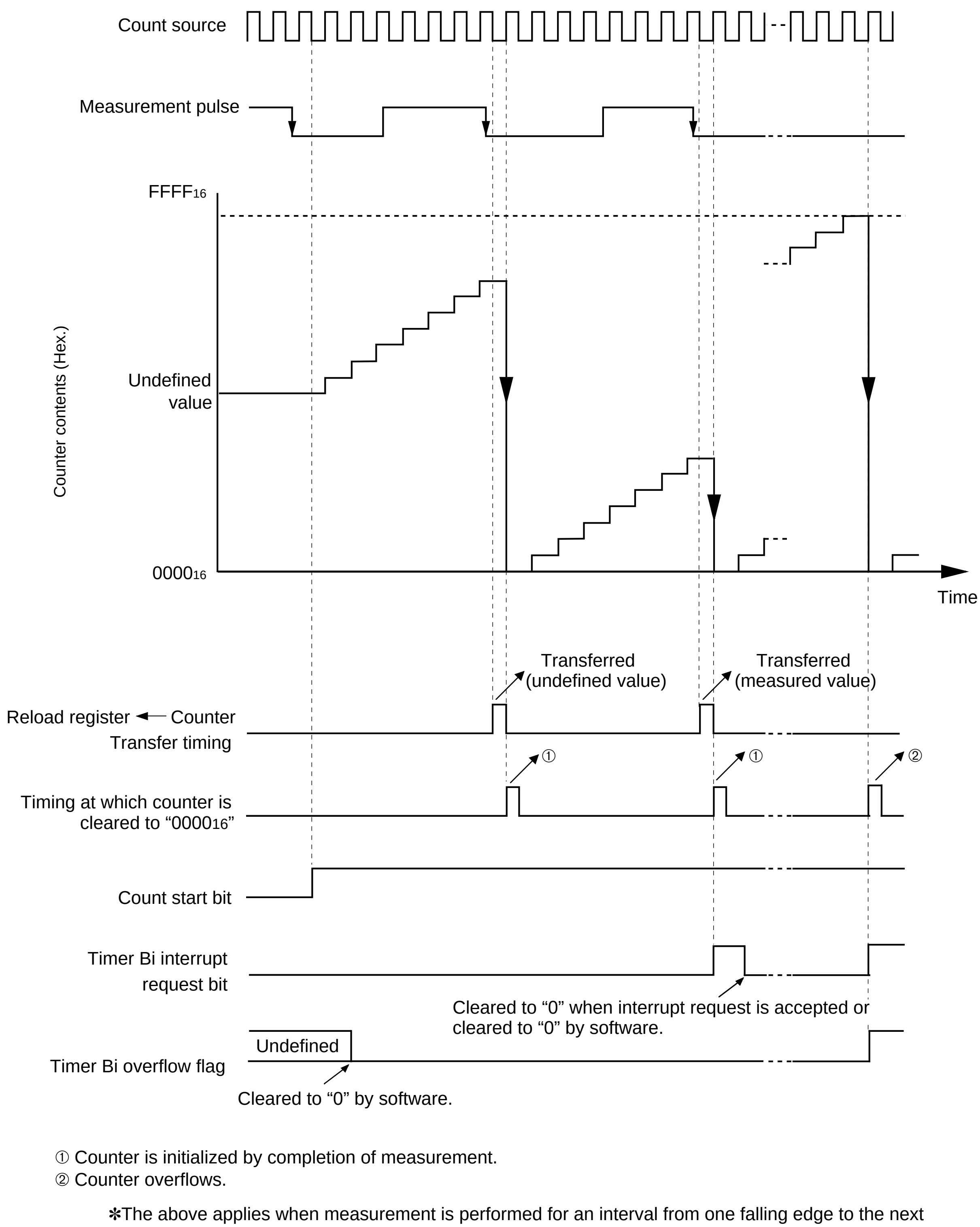
Figure 8.5.3 shows the processing example of a timer Bi interrupt when a measurement pulse's valid edge is detected by the timer Bi interrupt request.



Notes 1: The valid edge of the measurement pulse is detected.
2: Be sure to read out the timer Bi register.
3: After the timer Bi overflow flag is set to "1", be sure to wait for one cycle of the count source to elapse.
Then, write a value to the timer Bi mode register.

Fig. 8.5.3 Processing example of timer Bi interrupt when free-run count type is selected

Figures 8.5.4 and 8.5.5 show the operation examples during the pulse period measurement; Figures 8.5.6 and 8.5.7 show the operation examples during the pulse width measurement.



① Counter is initialized by completion of measurement.
② Counter overflows.

✽The above applies when measurement is performed for an interval from one falling edge to the next falling edge of the measurement pulse.

Fig. 8.5.4 Operation examples during pulse period measurement (when counter clear type is selected)

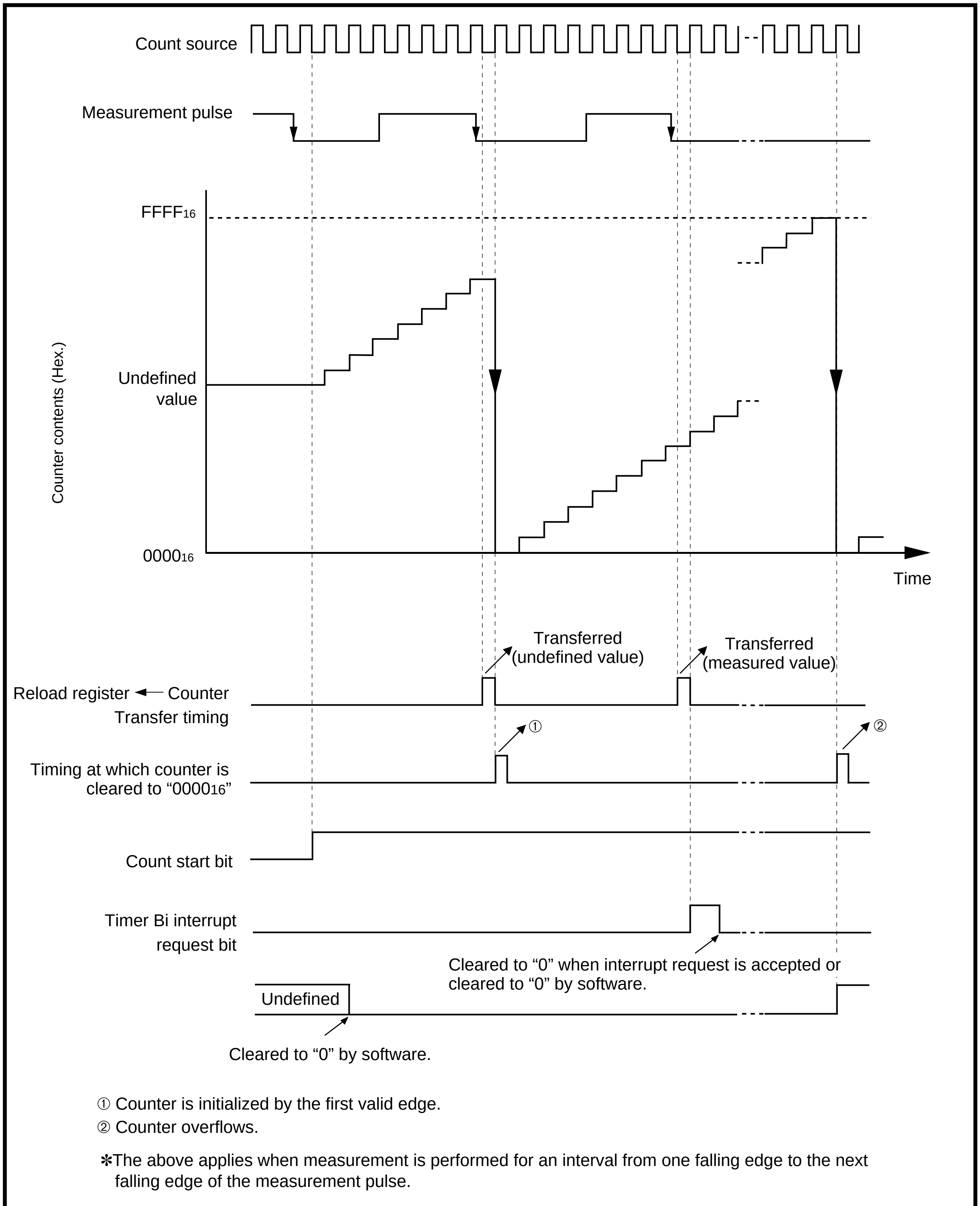


Fig. 8.5.5 Operation examples during pulse period measurement (when free-run count type is selected)

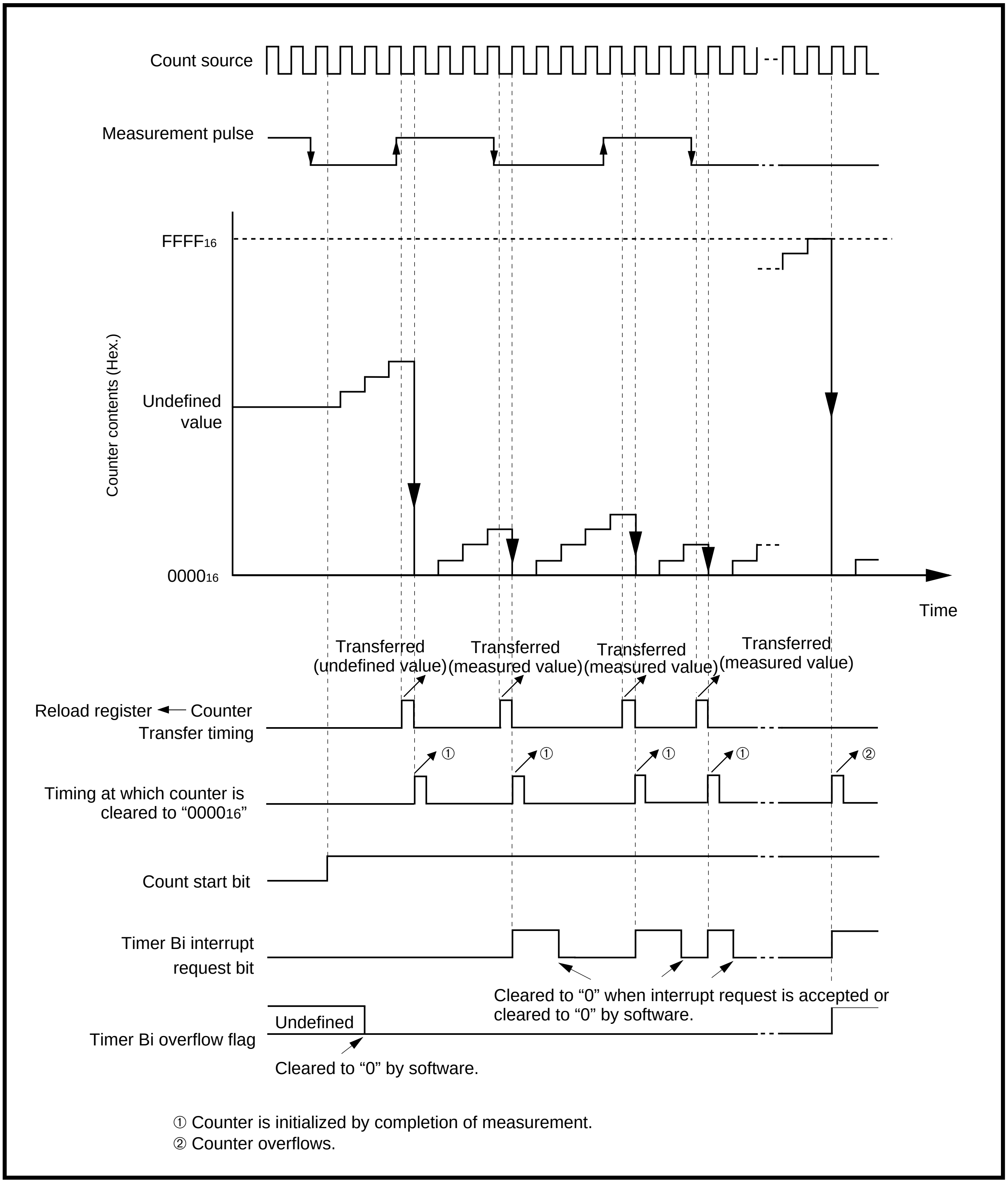


Fig. 8.5.6 Operation example during the pulse width measurement (when counter clear type is selected)

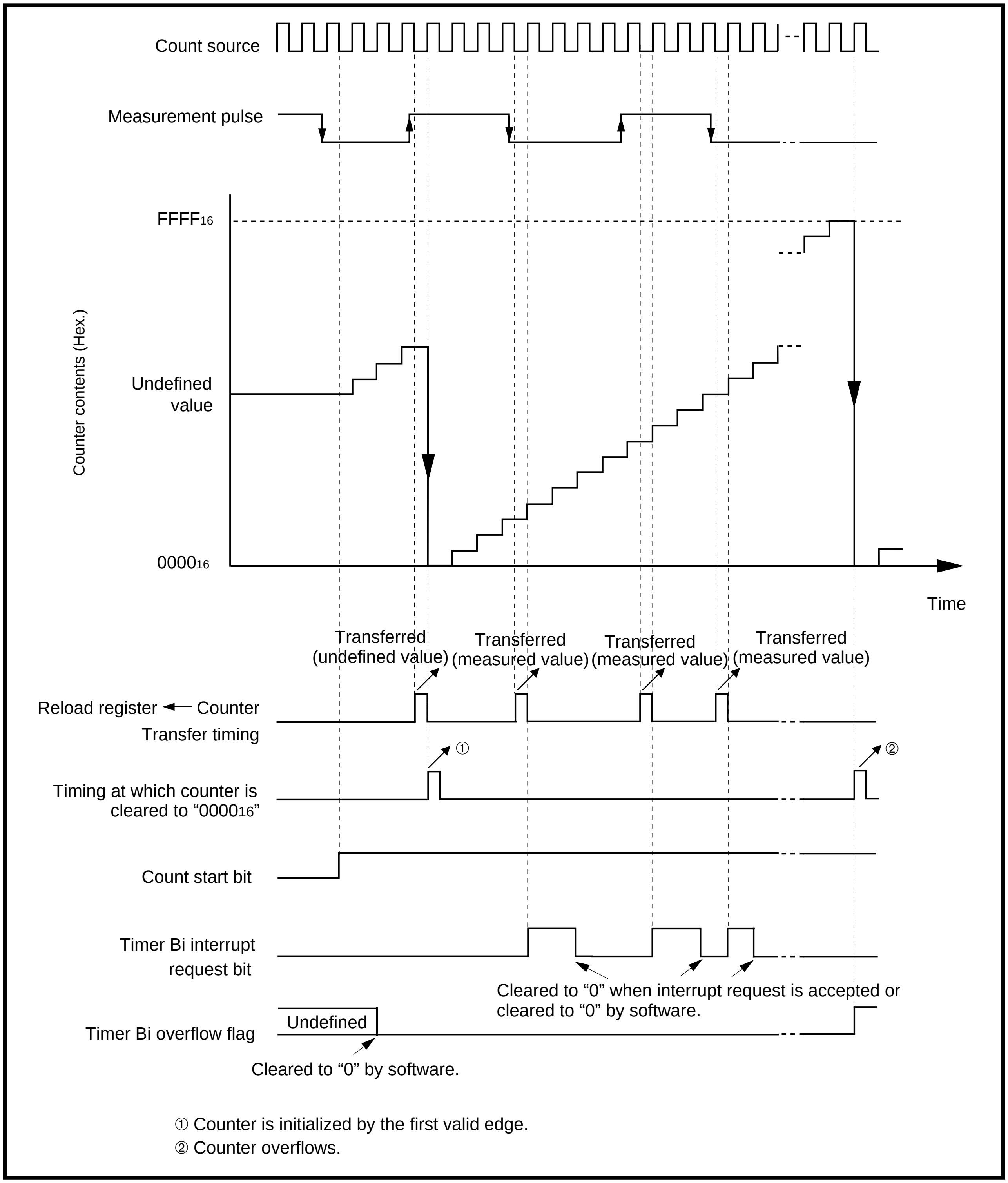


Fig. 8.5.7 Operation example during the pulse width measurement (when free-run count type is selected)

[Precautions for pulse period/pulse width measurement mode]

1. When the counter clear type is selected, a timer Bi interrupt request is generated by one of the following sources:
   - Valid edge input of measured pulse
   - Counter overflow

   When an overflow generates an interrupt request, the timer Bi overflow flag will be set to “1.”

2. When the free-run type is selected, the timer Bi interrupt request is generated only by the valid edge input of the pulse to be measured.

3. After reset, the timer Bi overflow flag is undefined. When a value is written to the timer Bi mode register after the next count timing of the count source with the count start bit = “1,” this flag will be cleared to “0.”

4. An undefined value is transferred to the reload register at the first valid edge input after the count start. In this case, no timer Bi interrupt request will occur.

5. The counter value at count start is undefined. Therefore, there is a possibility that a counter overflow occurs immediately after the counting starts. In this case, the timer Bi overflow flag becomes “1”; and when the counter clear type is selected, a timer Bi interrupt request is generated.

6. If the contents of the measurement mode select bits are changed after the count start, the timer Bi interrupt request bit is set to “1.” When the value, which has been set in these bits before, are written again, the timer Bi interrupt request bit will not change.

7. When using timer B2, be sure to clear the timer B2 clock source select bit (bit 6 at address 63_{16}) to “0.”

8. If the input signal to the TBi_{IN} pin is affected by noise, etc., there is a possibility that the counter cannot perform the exact measurement. We recommend to verify, by software, that the measurement values are within a constant range.

[Precautions for pulse period/pulse width measurement mode]

MEMORANDUM

CHAPTER 9

PULSE OUTPUT PORT MODE

9.1 Overview

The pulse output port mode function is used to change the output levels at several pins simultaneously with the following: each underflow occurrence in timer A or each valid edge input of an external signal. The pulse output port mode consists of pulse output port 0 and pulse output port 1. These two circuits have the equivalent functions and operate independently each other. Each of pulse output port 0 and pulse output port 1 has two operation modes as listed in Table 9.1.1. Table 9.1.2 lists the overview of pulse output port 0; Table 9.1.3 lists the overview of pulse output port 1.

Table 9.1.1 Overview of pulse output port mode

Function	Pulse output mode			
Circuit	Pulse output port 0		Pulse output port 1	
Operation mode	Pulse mode 0	Pulse mode 1	Pulse mode 0	Pulse mode 1

Table 9.1.2 Overview of pulse output port 0

Operation mode	Pulse mode 0		Pulse mode 1	
Pulse output pins	$RTP0_0$ to $RTP0_3$ ($P6_0$ to $P6_3$)	$RTP1_0$ to $RTP1_3$ ($P6_4$ to $P6_7$)	$RTP0_0$ to $RTP0_3$, $RTP1_0$, $RTP1_1$ ($P6_0$ to $P6_5$)	$RTP1_2$, $RTP1_3$ ($P6_4$, $P6_7$)
Pulse output trigger	Underflow occurrence in timer A0 or Valid edge of signal input to pin RTP_{TRG0}	Underflow of timer A3	Underflow of timer A0 or Valid edge of signal input to pin RTP_{TRG0}	Underflow of timer A3
Register where output data is to be set	Three-phase output data register 0 (bits 0 to 3)	Three-phase output data register 1 (bits 4 to 7)	Three-phase output data register 0 (bits 0 to 5)	Three-phase output data register 1 (bits 6, 7)
Pulse width modulation	Available (timer A1 used)	Not available	Available **(Note)** (timers A1, A2, A4 used)	Not available
Negative pulse output	Available	Available	Available	Not available
Pulse-output-cutoff signal input pin	$\overline{P6OUT_{CUT}}$ (Input of falling edge)	—	$\overline{P6OUT_{CUT}}$ (Input of falling edge)	—

Note: The pulse output pins, where pulse width modulation is to be applied, determine the timer to be used.

① 6 pins
   $RTP0_0$ to $RTP0_3$, $RTP1_0$, $RTP1_1$: timer A1

② 2 groups of 3 pins
- $RTP0_0$ to $RTP0_2$: timer A1
- $RTP0_3$, $RTP1_0$, $RTP1_1$: timer A2

③ 3 groups of 2 pins
- $RTP0_0$, $RTP0_1$: timer A1
- $RTP0_2$, $RTP0_3$: timer A2
- $RTP1_0$, $RTP1_1$: timer A4

Table 9.1.3 Overview of pulse output port 1

Operation mode	Pulse mode 0		Pulse mode 1	
Pulse output pins	$RTP2_0$ to $RTP2_3$ ($P4_0$ to $P4_3$)	$RTP3_0$ to $RTP3_3$ ($P4_4$ to $P4_7$)	$RTP2_0$ to $RTP2_3$, $RTP3_0$, $RTP3_1$ ($P4_0$ to $P4_5$)	$RTP3_2$, $RTP3_3$ ($P4_6$, $P4_7$)
Pulse output trigger	Underflow occurrence in timer A5 or Valid edge of signal input to pin RTP_{TRG1}	Underflow of timer A8	Underflow of timer A5 or Valid edge of signal input to pin RTP_{TRG1}	Underflow of timer A8
Register where output data is to be set	Pulse output data register 0 (bits 0 to 3)	Pulse output data register 1 (bits 4 to 7)	Pulse output data register 0 (bits 0 to 5)	Pulse output data register 1 (bits 6, 7)
Pulse width modulation	Available (timer A6 used)	Not available	Available **(Note)** (timers A6, A7, A9 used)	Not available
Negative pulse output	Available	Available	Available	Not available
Pulse-output-cutoff signal input pin	$\overline{P4OUT_{CUT}}$ (Input of falling edge)	—	$\overline{P4OUT_{CUT}}$ (Input of falling edge)	—

Note: The pulse output pins, where pulse width modulation is to be applied, determine the timer to be used.

① 6 pins
$RTP2_0$ to $RTP2_3$, $RTP3_0$, $RTP3_1$: timer A6

② 2 groups of 3 pins
- $RTP2_0$ to $RTP2_2$: timer A6
- $RTP2_3$, $RTP3_0$, $RTP3_1$: timer A7

③ 3 groups of 2 pins
- $RTP2_0$, $RTP2_1$: timer A6
- $RTP2_2$, $RTP2_3$: timer A7
- $RTP3_0$, $RTP3_1$: timer A9

9.2 Block description of pulse output port 0

Figure 9.2.1 shows the block diagram of pulse output port 0. Also, the pulse-output-port-0-relevant registers are described below.

In pulse output port 0 and three-phase waveform mode, the following registers are used in common: the waveform output mode register (address $A6_{16}$), three-phase output data register 0 (address $A8_{16}$), and three-phase output data register 1 (address $A9_{16}$). After pulse output port 0 is set by the waveform output select bits (bits 2 to 0 at address $A6_{16}$), be sure to set the relevant registers.

Note that, when not using pulse output port 0 and three-phase waveform mode, be sure to fix the waveform output select bits (bits 2 to 0 at address $A6_{16}$) to "000_2."

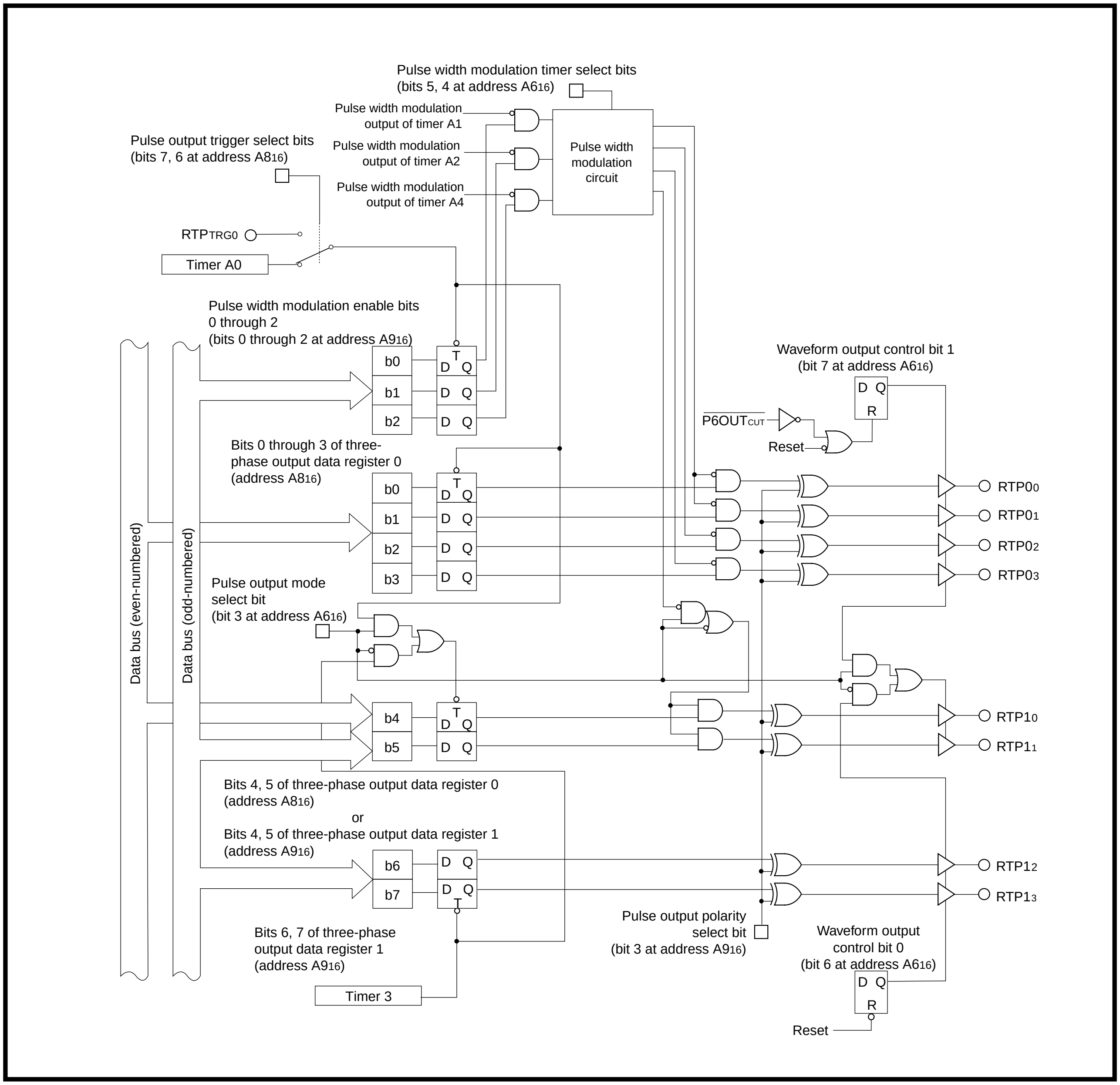


Fig. 9.2.1 Block diagram of pulse output port 0

9.2.1 Waveform output mode register

Figure 9.2.2 shows the structure of the waveform output mode register (pulse output port 0).

Waveform output mode register (Address $A6_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Waveform output select bits **(Note)**	See Table 9.2.1.	0	RW
1			0	RW
2			0	RW
3	Pulse output mode select bit	0 : Pulse mode 0 1 : Pulse mode 1	0	RW
4	Pulse width modulation timer select bits	See Table 9.2.2.	0	RW
5			0	RW
6	Waveform output control bit 0	When pulse mode 0 is selected, 0: $RTP1_0$ to $RTP1_3$: pulse outputs are disabled. 1: $RTP1_0$ to $RTP1_3$: pulse outputs are enabled. When pulse mode 1 is selected, 0: $RTP1_2$, $RTP1_3$: pulse outputs are disabled. 1: $RTP1_2$, $RTP1_3$: pulse outputs are enabled.	0	RW
7	Waveform output control bit 1	When pulse mode 0 is selected, 0 : $RTP0_0$ to $RTP0_3$: pulse outputs are disabled. 1 : $RTP0_0$ to $RTP0_3$: pulse outputs are enabled. When pulse mode 1 is selected, 0 : $RTP0_0$ to $RTP0_3$, $RTP1_0$, $RTP1_1$: pulse outputs are disabled. 1 : $RTP0_0$ to $RTP0_3$ $RTP1_0$, $RTP1_1$: pulse outputs are enabled.	0	RW

Note: When not using pulse output port 0 and three-phase waveform mode, be sure to fix these bits to "000_2."

Fig. 9.2.2 Structure of waveform output mode register (pulse output port 0)

PULSE OUTPUT PORT MODE

9.2 Block description of pulse output port 0

(1) Waveform output select bits (bits 2 to 0)

These bits are used to select whether a pin serves as a programmable I/O port pin or a pulse output pin. Table 9.2.1 lists the functions of the waveform output select bits.

Table 9.2.1 Functions of waveform output select bits

b2 b1 b0	000	001	010	011
Pulse mode 0 **(Note)**	$P6_7/RTP1_3$ $P6_6/RTP1_2$ $P6_5/RTP1_1$ $P6_4/RTP1_0$ → Port	$P6_7/RTP1_3$ $P6_6/RTP1_2$ $P6_5/RTP1_1$ $P6_4/RTP1_0$ → Port	$P6_7/RTP1_3$ $P6_6/RTP1_2$ $P6_5/RTP1_1$ $P6_4/RTP1_0$ → RTP	$P6_7/RTP1_3$ $P6_6/RTP1_2$ $P6_5/RTP1_1$ $P6_4/RTP1_0$ → RTP
	$P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → Port	$P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → RTP	$P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → Port	$P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → RTP
Pulse mode 1 **(Note)**	$P6_7/RTP1_3$ $P6_6/RTP1_2$ → Port	$P6_7/RTP1_3$ $P6_6/RTP1_2$ → Port	$P6_7/RTP1_3$ $P6_6/RTP1_2$ → Port	$P6_7/RTP1_3$ $P6_6/RTP1_2$ → RTP
	$P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → Port	$P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → RTP	$P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → Port	$P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → RTP

Port: This serves as a programmable I/O port pin or timer I/O pin.
RTP: This serves as a pulse output pin regardless of the contents of the corresponding port direction register.
Note: This is selected by the pulse output mode select bit (bit 3 at address $A6_{16}$).

(2) Pulse output mode select bit (bit 3)

This bit is used to select the operation mode of pulse output port 0: pulse mode 0 or pulse mode 1.

(3) Pulse width modulation timer select bits (bits 5 and 4)

These bits are used to select the type of the pulse width modulation of pulse output port 0. Table 9.2.2 lists the functions of the pulse width modulation timer select bits.

Table 9.2.2 Functions of pulse width modulation timer select bits

b5 b4	00	01	10	11
Pulse mode 0 **(Note 1)**	$P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → Timer A1	Do not select.	Do not select.	Do not select.
Pulse mode 1 **(Note 2)**	$P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → Timer A1	$P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ → Timer A2 $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ → Timer A1	$P6_5/RTP1_1$ $P6_4/RTP1_0$ → Timer A4 $P6_3/RTP0_3$ $P6_2/RTP0_2$ → Timer A2 $P6_1/RTP0_1$ $P6_0/RTP0_0$ → Timer A1	Do not select.

Note 1: The pulse width modulation cannot be applied to pins $RTP1_0$ to $RTP1_3$.
2: The pulse width modulation cannot be applied to pins $RTP1_2$ and $RTP1_3$.

(4) Waveform output control bits 1, 0 (bits 7, 6)

These bits are used to control the waveform output of pulse output port 0. Table 9.2.3 lists the functions of waveform output control bits 1, 0.

When a falling edge is input to pin $\overline{P6OUT_{CUT}}$, waveform output control bit 1 (bit 7) becomes "0." (See Table 9.2.7.)

Table 9.2.3 Functions of waveform output control bits 1, 0

b2 b1 b0	000	001	010	011
Pulse mode 0	$P6_7/RTP1_3$ $P6_6/RTP1_2$ $P6_5/RTP1_1$ $P6_4/RTP1_0$ } Floating state $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ } Floating state	$P6_7/RTP1_3$ $P6_6/RTP1_2$ $P6_5/RTP1_1$ $P6_4/RTP1_0$ } Pulse output enabled $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ } Floating state	$P6_7/RTP1_3$ $P6_6/RTP1_2$ $P6_5/RTP1_1$ $P6_4/RTP1_0$ } Floating state $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ } Pulse output enabled	$P6_7/RTP1_3$ $P6_6/RTP1_2$ $P6_5/RTP1_1$ $P6_4/RTP1_0$ } Pulse output enabled $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ } Pulse output enabled
Pulse mode 1	$P6_7/RTP1_3$ $P6_6/RTP1_2$ } Floating state $P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ } Floating state	$P6_7/RTP1_3$ $P6_6/RTP1_2$ } Pulse output enabled $P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ } Floating state	$P6_7/RTP1_3$ $P6_6/RTP1_2$ } Floating state $P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ } Pulse output enabled	$P6_7/RTP1_3$ $P6_6/RTP1_2$ } Pulse output enabled $P6_5/RTP1_1$ $P6_4/RTP1_0$ $P6_3/RTP0_3$ $P6_2/RTP0_2$ $P6_1/RTP0_1$ $P6_0/RTP0_0$ } Pulse output enabled

9.2.2 Three-phase output data registers 0, 1

Figure 9.2.3 shows the structures of three-phase output data registers 0, 1 (pulse output port 0).

Three-phase output data register 0 (Address $A8_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	$RTP0_0$ pulse output data bit	0 : "L" level output 1 : "H" level output	0	RW
1	$RTP0_1$ pulse output data bit		0	RW
2	$RTP0_2$ pulse output data bit		0	RW
3	$RTP0_3$ pulse output data bit		0	RW
4	$RTP1_0$ pulse output data bit (Valid in pulse mode 1.) **(Note)**		0	RW
5	$RTP1_1$ pulse output data bit (Valid in pulse mode 1.) **(Note)**		0	RW
7, 6	Pulse output trigger select bits	b7 b6 0 0 : Underflow of timer A0 0 1 : Falling edge of input signal to pin RTP_{TRG0} 1 0 : Rising edge of input signal to pin RTP_{TRG0} 1 1 : Both falling and rising edges of input signal to pin RTP_{TRG0}	0	RW

Note: Invalid in pulse mode 0.

Three-phase output data register 1 (Address $A9_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Pulse width modulation enable bit 0	0 : No pulse width modulation by timer A1 1 : Pulse width modulation by timer A1	0	RW
1	Pulse width modulation enable bit 1	0 : No pulse width modulation by timer A2 1 : Pulse width modulation by timer A2	0	RW
2	Pulse width modulation enable bit 2	0 : No pulse width modulation by timer A4 1 : Pulse width modulation by timer A4	0	RW
3	Pulse output polarity select bit	0 : Positive 1 : Negative	0	RW
4	$RTP1_0$ pulse output data bit (Valid in pulse mode 0) **(Note)**	0 : "L" level output 1 : "H" level output	0	RW
5	$RTP1_1$ pulse output data bit (Valid in pulse mode 0) **(Note)**		0	RW
6	$RTP1_2$ pulse output data bit		0	RW
7	$RTP1_3$ pulse output data bit		0	RW

Note: Invalid in pulse mode 1.

Fig. 9.2.3 Structures of three-phase output data registers 0, 1 (pulse output port 0)

(1) RTP0$_0$ to RTP0$_3$ pulse output data bits (bits 0 to 3 at address A8$_{16}$)

Each time when a pulse output trigger is generated, the contents written to these bits are output from the corresponding pulse output pins **(Note).** The pulse output trigger can be selected by the pulse output trigger select bits (bits 7, 6 at address A8$_{16}$).

(2) RTP1$_0$, RTP1$_1$ pulse output data bits (bits 4, 5 at address A8$_{16}$)

These bits are valid in pulse mode 1.
Each time when a pulse output trigger is generated, the contents written to these bits are output from the corresponding pulse output pins **(Note).** The pulse output trigger can be selected by the pulse output trigger select bits (bits 7, 6 at address A8$_{16}$).
These bits are invalid in pulse mode 0.

(3) Pulse output trigger select bits (bits 7, 6 at address A8$_{16}$)

The pulse output trigger can be selected from an internal trigger and an external trigger. When using an external trigger (input signal to pin RTP$_{TRG0}$), be sure to clear the port P5 direction register's bit, corresponding to port P5$_3$ pin, in order to set this port P5$_3$ pin for the input mode.

(4) Pulse width modulation enable bits 0 to 2 (bits 0 to 2 at Address A9$_{16}$)

These bits are used to select the pins, where the pulse width modulation is to be applied. Synchronous with a pulse output trigger, the contents of these bits become valid. Table 9.2.4 lists the pulse-width-modulation-relevant bits.

(5) Pulse output polarity select bit (bit 3 at address A9$_{16}$)

When this bit = "0," the data corresponding to the contents which have been set in the RTP0$_0$ to RTP0$_3$, RTP1$_0$ to RTP1$_3$ pulse output data bits are output from pins RTP0$_0$ to RTP0$_3$, RTP1$_0$ to RTP1$_3$. When this bit = "1," the contents which have been set in the RTP0$_0$ to RTP0$_3$, RTP1$_0$ to RTP1$_3$ pulse output data bits are reversed (in other words, pulses with the negative polarity are generated here.); and then, these pulses with the negative polarity are output from pins RTP0$_0$ to RTP0$_3$, RTP1$_0$ to RTP1$_3$.

Note that, in pulse mode 1, the pulses with the negative polarity are not output from pins RTP1$_2$ and RTP1$_3$.

(6) RTP1$_0$, RTP1$_1$ pulse output data bits (bits 4, 5 at address A9$_{16}$)

These bits are valid in pulse mode 0.
Each time when an underflow occurs in timer A3, the contents which have been written to these bits are output from the corresponding pulse output pins **(Note)**.
These bits are invalid in pulse mode 1.

(7) RTP1$_2$, RTP1$_3$ pulse output data bits (bits 6, 7 at address A9$_{16}$)

Each time when an underflow occurs in timer A3, the contents which have been written to these bits are output from the corresponding pulse output pins **(Note)**.

Note: The output level at a pulse output pin is undefined in the period from when data is written to these bits until the first occurrence of a pulse output trigger. If it is necessary to avoid this state, perform "Processing of avoiding undefined output before starting pulse output" in Figure 9.4.2.

9.2 Block description of pulse output port 0

Table 9.2.4 Pulse-width-modulation-relevant bits

Pulse output pins where pulse width modulation is to be applied (Timers used for pulse width modulation)			Pulse width modulation timer select bits (bits 5, 4 at address $A6_{16}$)	Pulse width modulation enable bit 2 (bit 2 at address $A9_{16}$)	Pulse width modulation enable bit 1 (bit 1 at address $A9_{16}$)	Pulse width modulation enable bit 0 (bit 0 at address $A9_{16}$)
Pulse mode 0	4 pins	$RTP0_3$ to $RTP0_0$ (Timer A1)	00	×	×	1
Pulse mode 1	6 pins	$RTP1_1$, $RTP1_0$, $RTP0_3$ to $RTP0_0$ (Timer A1)	00	×	×	1
	In a unit of 3 pins	$RTP1_1$, $RTP1_0$, $RTP0_3$ (Timer A2)	01	×	1	×
		$RTP0_2$ to $RTP0_0$ (Timer A1)		×	×	1
	In a unit of 2 pins	$RTP1_1$, $RTP1_0$ (Timer A4)	10	1	×	×
		$RTP0_3$, $RTP0_2$ (Timer A2)		×	1	×
		$RTP0_1$, $RTP0_0$ (Timer A1)		×	×	1

X: It may be either “0” or “1.”

9.2.3 Port P5 direction register

The pulse output trigger input pin is multiplexed with port $P5_3$ pin.

When using pin $P5_3$/RTP_{TRG0} as a pulse output trigger input pin, be sure to clear the port P5 direction register's bit, corresponding to port $P5_3$ pin, in order to set this port $P5_3$ pin for the input mode.

Figure 9.2.4 shows the relationship between port P5 direction register and a pulse output trigger input pin.

Port P5 direcition register (Address D_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Corresponding pin	Function	At reset	R/W
0	Nothing is assigned.		Undefined	—
1	Pin $\overline{INT_1}$	0 : Input mode 1 : Output mode When using this pin as a pulse output trigger input pin, be sure to clear the corresponding bit to "0."	0	RW
2	Pin $\overline{INT_2}$/RTP_{TRG1}		0	RW
3	Pin RTP_{TRG0} (Pin $\overline{INT_3}$)		0	RW
4	Nothing is assigned.		Undefined	—
5	Pin $\overline{INT_5}$/$TB0_{IN}$/IDW	0 : Input mode 1 : Output mode	0	RW
6	Pin $\overline{INT_6}$/$TB1_{IN}$/IDV		0	RW
7	Pin $\overline{INT_7}$/$TB2_{IN}$/IDU		0	RW

Note: ( ) shows the I/O pin of another internal peripheral device which is multiplexed.

Fig. 9.2.4 Relationship between port P5 direction register and pulse output trigger input pin

9.2.4 Timers A0 to A4

Timers A0 and A3 are used as control registers; each generates a pulse output trigger. When using timers A0 and A3, be sure to use them in the timer mode. (Refer to section **"7.3 Timer mode."**)
When performing the pulse width modulation, be sure to use timers A1, A2, A4 in the pulse width modulation mode. (Refer to section **"7.6 Pulse width modulation (PWM) mode."**) Note that, from pin $P2_0$/$TA4_{OUT}$, a PWM pulse by timer A4 is output. When it is unnecessary to output a PWM pulse, be sure to clear bit 2 of the timer A4 mode register (address $5A_{16}$) to "0." At this time, pin $P2_0$ can be used as a programmable I/O port pin.
Figure 9.2.5 shows the structure of the timer A0 and A3 mode registers (pulse output port 0); Figure 9.2.6 shows the structures of the timer A1, A2, A4 mode registers (pulse output port 0 with pulse width modulation used).

Timer A0 mode register (Address 56_{16})
Timer A3 mode register (Address 59_{16})

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0	0	0	0

Bit	Bit name	Functions	At reset	R/W
0	Fix these bits to "000000_2" in the pulse output port mode.		0	RW
1			0	RW
2			0	RW
3			0	RW
4			0	RW
5			0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 9.2.5 Structure of timer A0 and A3 mode registers (pulse output port 0)

Timer A1 mode register (Address 57_{16})
Timer A2 mode register (Address 58_{16})

b7	b6	b5	b4	b3	b2	b1	b0
			0	0	0	1	1

Bit	Bit name	Functions	At reset	R/W
0	Fix these bits to “00011_2” in the pulse output port mode.		0	RW
1			0	RW
2			0	RW
3			0	RW
4			0	RW
5	16/8-bit PWM mode select bit	0 : 16-bit pulse width modulator 1 : 8-bit pulse width modulator	0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Timer A4 mode register (Address $5A_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0		1	1

Bit	Bit name	Functions	At reset	R/W
0	Fix these bits to “11_2” in the pulse output port mode.		0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output (TA4OUT pin functions as a programmable I/O port pin.) 1 : Pulse output (TA4OUT pin functions as a PWM pulse output pin.)	0	RW
3	Fix these bits to “00_2” in the pulse output port mode.		0	RW
4			0	RW
5	16/8-bit PWM mode select bit	0 : 16-bit pulse width modulator 1 : 8-bit pulse width modulator	0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 9.2.6 Structures of timer A1, A2, A4 mode registers (pulse output port 0 with pulse width modulation used)

9.2 Block description of pulse output port 0

9.2.5 Pin $\overline{P6OUT_{CUT}}$ (pulse-output-cutoff signal input pin)

When a falling edge is input to pin $\overline{P6OUT_{CUT}}$, the waveform output control bit 1 (bit 7 at address $A6_{16}$) becomes "0" and the pulse output pins enter the floating state. (In other words, pulse output becomes disabled.) The pulse output pins where pulse output is to be inactive depend on the pulse output mode.

- Pulse mode 0: $RTP0_0$ to $RTP0_3$
- Pulse mode 1: $RTP0_0$ to $RTP0_3$, $RTP1_0$, $RTP1_1$

When restarting pulse output after the pulse output becomes inactive, be sure to return the input level at pin $\overline{P6OUT_{CUT}}$ to "H" level; and then, be sure to set the waveform output control bit 1 to "1." When the input level at pin $\overline{P6OUT_{CUT}}$ is "L" level, the waveform output control bit 1 cannot be "1."

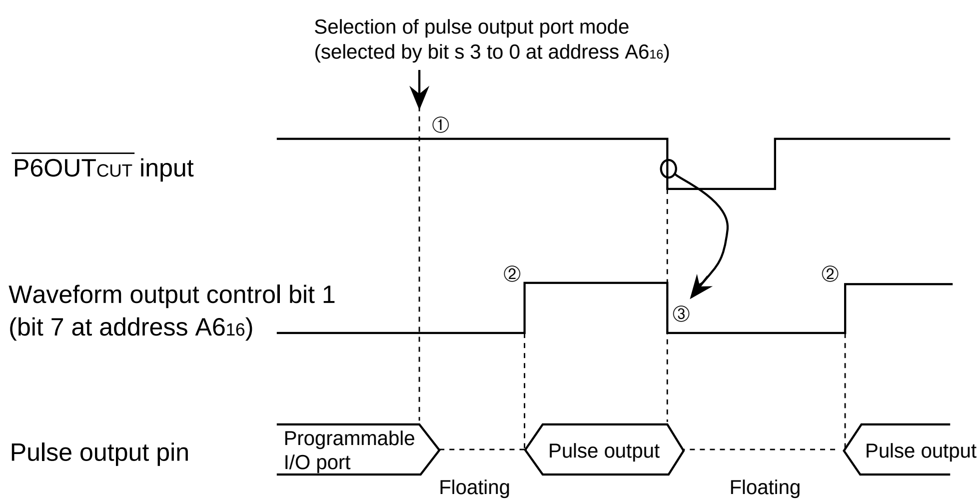
Also, at this time, bits 0 through 7 of the port P6 direction register (address 10_{16}) become "00000000_2." (Refer to section **"5.2.4 Pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$."**) Therefore, if it is necessary to switch port pins $P6_0$ through $P6_7$ to port output pins, be sure to do as follows:

① Return the input level at pin $\overline{P6OUT_{CUT}}$ to "H" level.
② Write data to the port P6 register (address E_{16})'s bits, corresponding to the port P6 pins which will output data.
③ Set the port P6 direction register's bits, corresponding to the port P6 pins in ②, to "1" in order to set these port pins to the output mode.

When the input level at pin $\overline{P6OUT_{CUT}}$ is "L" level, no bit of the port P6 direction register can be "1."

Figure 9.2.7 shows the relationship between the $\overline{P6OUT_{CUT}}$ input, waveform output control bit 1, and pulse output pin.

Note that, when not making the pulse output inactive by using pin $\overline{P6OUT_{CUT}}$, be sure to connect pin $\overline{P6OUT_{CUT}}$ to Vcc via a resistor.



① When the pulse output port mode is selected, the pulse outpit pins become floating.
② The pulse is output by writing of "1" with the input level at pin $\overline{P6OUT_{CUT}}$ = "H."
③ When a falling edge is input to pin $\overline{P6OUT_{CUT}}$, this bit becomes "0."

Fig. 9.2.7 Relationship between $\overline{P6OUT_{CUT}}$ input, waveform output control bit 1, and pulse output pin

9.3 Block description of pulse output port 1

Figure 9.3.1 shows the block diagram of pulse output port 1. Also, the pulse-output-port-1-relevant registers are described below.

After pulse output port 1 is set by the waveform output select bits (bits 2 to 0 at address $A0_{16}$), be sure to set the relevant registers.

Note that, when not using pulse output port 1, be sure to fix the waveform output select bits (bits 2 to 0 at address $A0_{16}$) to "000_2."

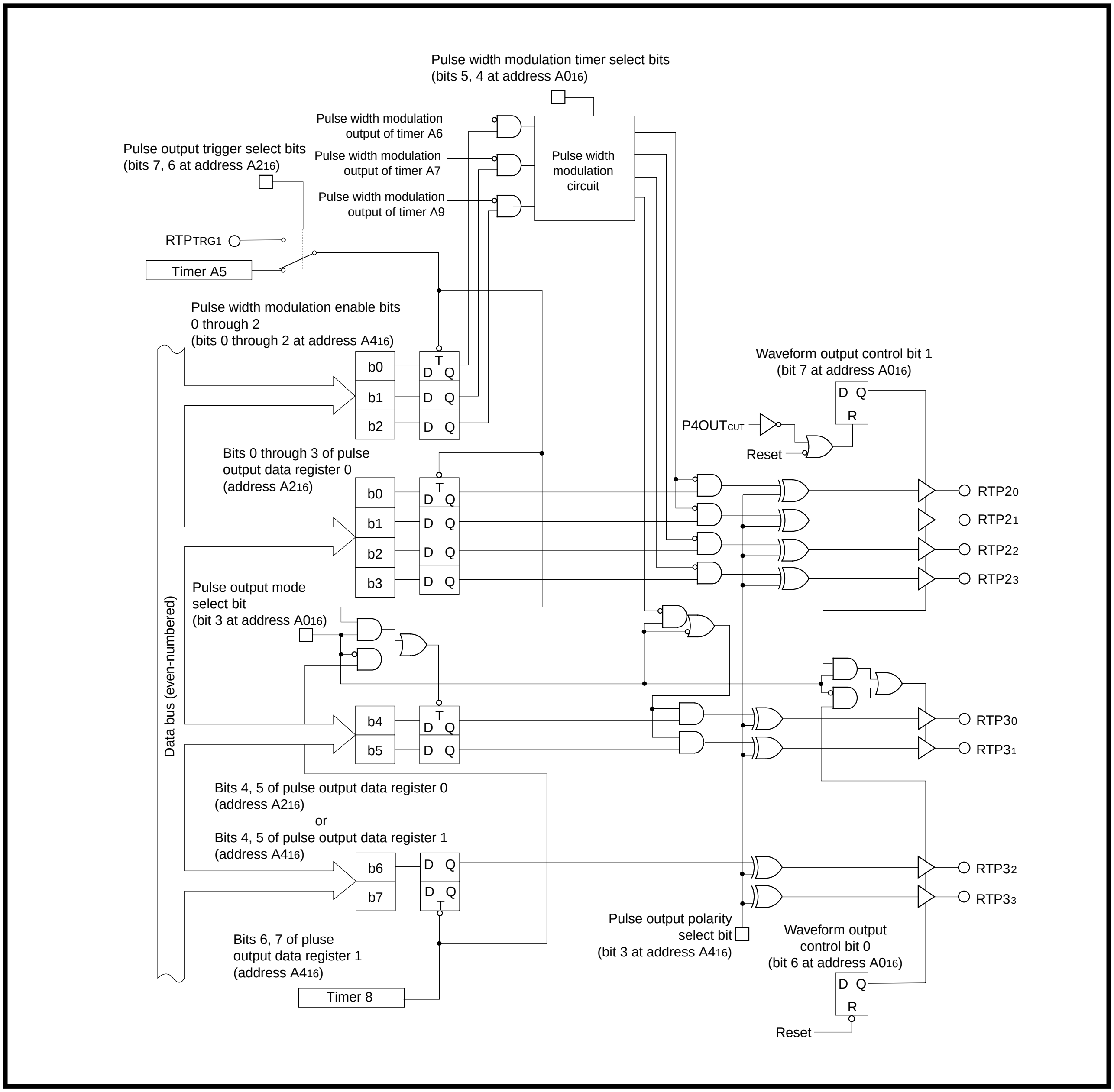


Fig. 9.3.1 Block diagram of pulse output port 1

9.3 Block description of pulse output port 1

9.3.1 Pluse output control register

Figure 9.3.2 shows the structure of the pluse output control register.

Pluse output control register (Address $A0_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Waveform output select bits **(Note)**	See Table 9.3.1.	0	RW
1			0	RW
2			0	RW
3	Pulse output mode select bit	0 : Pulse mode 0 1 : Pulse mode 1	0	RW
4	Pulse width modulation timer select bits	See Table 9.3.2.	0	RW
5			0	RW
6	Waveform output control bit 0	When pulse mode 0 is selected, 0: $RTP3_0$ to $RTP3_3$: pulse outputs are disabled. 1: $RTP3_0$ to $RTP3_3$: pulse outputs are enabled. When pulse mode 1 is selected, 0: $RTP3_2$, $RTP3_3$: pulse outputs are disabled. 1: $RTP3_2$, $RTP3_3$: pulse outputs are enabled.	0	RW
7	Waveform output control bit 1	When pulse mode 0 is selected, 0 : $RTP2_0$ to $RTP2_3$: pulse outputs are disabled. 1 : $RTP2_0$ to $RTP2_3$: pulse outputs are enabled. When pulse mode 1 is selected, 0 : $RTP2_0$ to $RTP2_3$, $RTP3_0$, $RTP3_1$: pulse outputs are disabled. 1 : $RTP2_0$ to $RTP2_3$ $RTP3_0$, $RTP3_1$: pulse outputs are enabled.	0	RW

Note: When not using pulse output port 1, be sure to fix these bits to "000_2."

Fig. 9.3.2 Structure of pluse output control register

(1) Waveform output select bits (bits 2 to 0)

These bits are used to select whether a pin serves as a programmable I/O port pin or a pulse output pin. Table 9.3.1 lists the functions of the waveform output select bits.

Table 9.3.1 Functions of waveform output select bits

b2 b1 b0	000	001	010	011
Pulse mode 0 **(Note)**	P4$_7$/RTP3$_3$ P4$_6$/RTP3$_2$ P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ } Port	P4$_7$/RTP3$_3$ P4$_6$/RTP3$_2$ P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ } Port	P4$_7$/RTP3$_3$ P4$_6$/RTP3$_2$ P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ } RTP	P4$_7$/RTP3$_3$ P4$_6$/RTP3$_2$ P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ } RTP
	P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } Port	P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } RTP	P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } Port	P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } RTP
Pulse mode 1 **(Note)**	P4$_7$/RTP3$_3$ P4$_6$/RTP3$_2$ } Port	P4$_7$/RTP3$_3$ P4$_6$/RTP3$_2$ } Port	P4$_7$/RTP3$_3$ P4$_6$/RTP3$_2$ } RTP	P4$_7$/RTP3$_3$ P4$_6$/RTP3$_2$ } RTP
	P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } Port	P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } RTP	P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } Port	P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } RTP

Port: This serves as a programmable I/O port pin or timer I/O pin.
RTP: This serves as a pulse output pin regardless of the contents of the corresponding port direction register.
Note: This is selected by the pulse output mode select bit (bit 3 at address A0$_{16}$).

(2) Pulse output mode select bit (bit 3)

This bit is used to select the operation mode of pulse output port 1: pulse mode 0 or pulse mode 1.

(3) Pulse width modulation timer select bits (bits 5 and 4)

These bits are used to select the type of the pulse width modulation of pulse output port 1. Table 9.3.2 lists the functions of the pulse width modulation timer select bits.

Table 9.3.2 Functions of pulse width modulation timer select bits

b5 b4	00	01	10	11
Pulse mode 0 **(Note 1)**	P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } Timer A6	Do not select.	Do not select.	Do not select.
Pulse mode 1 **(Note 2)**	P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } Timer A6	P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ P4$_3$/RTP2$_3$ } Timer A7 P4$_2$/RTP2$_2$ P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } Timer A6	P4$_5$/RTP3$_1$ P4$_4$/RTP3$_0$ } Timer A9 P4$_3$/RTP2$_3$ P4$_2$/RTP2$_2$ } Timer A7 P4$_1$/RTP2$_1$ P4$_0$/RTP2$_0$ } Timer A6	Do not select.

Note 1: The pulse width modulation cannot be applied to pins RTP3$_0$ to RTP3$_3$.
2: The pulse width modulation cannot be applied to pins RTP3$_2$ and RTP3$_3$.

9.3 Block description of pulse output port 1

(4) Waveform output control bits 1, 0 (bits 7, 6)

These bits are used to control the waveform output of pulse output port 1. Table 9.3.3 lists the functions of waveform output control bits 1, 0.

When a falling edge is input to pin $\overline{P4OUT_{CUT}}$, waveform output control bit 1 (bit 7) becomes “0.” (See Table 9.3.7.)

Table 9.3.3 Functions of waveform output control bits 1, 0

b2 b1 b0	00	01	10	11
Pulse mode 0	$P4_7/RTP3_3$ $P4_6/RTP3_2$ $P4_5/RTP3_1$ $P4_4/RTP3_0$ } Floating state	$P4_7/RTP3_3$ $P4_6/RTP3_2$ $P4_5/RTP3_1$ $P4_4/RTP3_0$ } Pulse output enabled	$P4_7/RTP3_3$ $P4_6/RTP3_2$ $P4_5/RTP3_1$ $P4_4/RTP3_0$ } Floating state	$P4_7/RTP3_3$ $P4_6/RTP3_2$ $P4_5/RTP3_1$ $P4_4/RTP3_0$ } Pulse output enabled
	$P4_3/RTP2_3$ $P4_2/RTP2_2$ $P4_1/RTP2_1$ $P4_0/RTP2_0$ } Floating state	$P4_3/RTP2_3$ $P4_2/RTP2_2$ $P4_1/RTP2_1$ $P4_0/RTP2_0$ } Floating state	$P4_3/RTP2_3$ $P4_2/RTP2_2$ $P4_1/RTP2_1$ $P4_0/RTP2_0$ } Pulse output enabled	$P4_3/RTP2_3$ $P4_2/RTP2_2$ $P4_1/RTP2_1$ $P4_0/RTP2_0$ } Pulse output enabled
Pulse mode 1	$P4_7/RTP3_3$ $P4_6/RTP3_2$ } Floating state	$P4_7/RTP3_3$ $P4_6/RTP3_2$ } Pulse output enabled	$P4_7/RTP3_3$ $P4_6/RTP3_2$ } Floating state	$P4_7/RTP3_3$ $P4_6/RTP3_2$ } Pulse output enabled
	$P4_5/RTP3_1$ $P4_4/RTP3_0$ $P4_3/RTP2_3$ $P4_2/RTP2_2$ $P4_1/RTP2_1$ $P4_0/RTP2_0$ } Floating state	$P4_5/RTP3_1$ $P4_4/RTP3_0$ $P4_3/RTP2_3$ $P4_2/RTP2_2$ $P4_1/RTP2_1$ $P4_0/RTP2_0$ } Floating state	$P4_5/RTP3_1$ $P4_4/RTP3_0$ $P4_3/RTP2_3$ $P4_2/RTP2_2$ $P4_1/RTP2_1$ $P4_0/RTP2_0$ } Pulse output enabled	$P4_5/RTP3_1$ $P4_4/RTP3_0$ $P4_3/RTP2_3$ $P4_2/RTP2_2$ $P4_1/RTP2_1$ $P4_0/RTP2_0$ } Pulse output enabled

9.3.2 Pulse output data registers 0, 1

Figure 9.3.3 shows the structures of pulse output data registers 0, 1.

b7 b6 b5 b4 b3 b2 b1 b0

Pulse output data register 0 (Address $A2_{16}$)

Bit	Bit name	Function	At reset	R/W
0	$RTP2_0$ pulse output data bit	0 : "L" level output 1 : "H" level output	0	RW
1	$RTP2_1$ pulse output data bit		0	RW
2	$RTP2_2$ pulse output data bit		0	RW
3	$RTP2_3$ pulse output data bit		0	RW
4	$RTP3_0$ pulse output data bit (Valid in pulse mode 1.) **(Note)**		0	RW
5	$RTP3_1$ pulse output data bit (Valid in pulse mode 1.) **(Note)**		0	RW
7, 6	Pulse output trigger select bits	b7 b6 0 0 : Underflow of timer A5 0 1 : Falling edge of input signal to pin RTP_{TRG1} 1 0 : Rising edge of input signal to pin RTP_{TRG1} 1 1 : Both falling and rising edges of input signal to pin RTP_{TRG1}	0	RW

Note: Invalid in pulse mode 0.

b7 b6 b5 b4 b3 b2 b1 b0

Pulse output data register 1 (Address $A4_{16}$)

Bit	Bit name	Function	At reset	R/W
0	Pulse width modulation enable bit 0	0 : No pulse width modulation by timer A6 1 : Pulse width modulation by timer A6	0	RW
1	Pulse width modulation enable bit 1	0 : No pulse width modulation by timer A7 1 : Pulse width modulation by timer A7	0	RW
2	Pulse width modulation enable bit 2	0 : No pulse width modulation by timer A9 1 : Pulse width modulation by timer A9	0	RW
3	Pulse output polarity select bit	0 : Positive 1 : Negative	0	RW
4	$RTP3_0$ pulse output data bit (Valid in pulse mode 0) **(Note)**	0 : "L" level output 1 : "H" level output	0	RW
5	$RTP3_1$ pulse output data bit (Valid in pulse mode 0) **(Note)**		0	RW
6	$RTP3_2$ pulse output data bit		0	RW
7	$RTP3_3$ pulse output data bit		0	RW

Note: Invalid in pulse mode 1.

Fig. 9.3.3 Structures of pulse output data registers 0, 1

9.3 Block description of pulse output port 1

(1) $RTP2_0$ to $RTP2_3$ pulse output data bits (bits 0 to 3 at address $A2_{16}$)

Each time when a pulse output trigger is generated, the contents written to these bits are output from the corresponding pulse output pins **(Note).** The pulse output trigger can be selected by the pulse output trigger select bits (bits 7, 6 at address $A2_{16}$).

(2) $RTP3_0$, $RTP3_1$ pulse output data bits (bits 4, 5 at address $A2_{16}$)

These bits are valid in pulse mode 1.
Each time when a pulse output trigger is generated, the contents written to these bits are output from the corresponding pulse output pins **(Note).** The pulse output trigger can be selected by the pulse output trigger select bits (bits 7, 6 at address $A2_{16}$).
These bits are invalid in pulse mode 0.

(3) Pulse output trigger select bits (bits 7, 6 at address $A2_{16}$)

The pulse output trigger can be selected from an internal trigger and an external trigger. When using an external trigger (input signal to pin RTP_{TRG1}), be sure to clear the port P5 direction register's bit, corresponding to port $P5_2$ pin, in order to set this port $P5_2$ pin for the input mode.

(4) Pulse width modulation enable bits 0 to 2 (bits 0 to 2 at Address $A4_{16}$)

These bits are used to select the pins, where the pulse width modulation is to be applied. Synchronous with a pulse output trigger, the contents of these bits become valid. Table 9.3.4 lists the pulse-width-modulation-relevant bits.

(5) Pulse output polarity select bit (bit 3 at address $A4_{16}$)

When this bit = "0," the data corresponding to the contents which have been set in the $RTP2_0$ to $RTP2_3$, $RTP3_0$ to $RTP3_3$ pulse output data bits are output from pins $RTP2_0$ to $RTP2_3$, $RTP3_0$ to $RTP3_3$.
When this bit = "1," the contents which have been set in the $RTP2_0$ to $RTP2_3$, $RTP3_0$ to $RTP3_3$ pulse output data bits are reversed (in other words, pulses with the negative polarity are generated here.); and then, these pulses with the negative polarity are output from pins $RTP2_0$ to $RTP2_3$, $RTP3_0$ to $RTP3_3$.
Note that, in pulse mode 1, the pulses with the negative polarity are not output from pins $RTP3_2$ and $RTP3_3$.

(6) $RTP3_0$, $RTP3_1$ pulse output data bits (bits 4, 5 at address $A4_{16}$)

These bits are valid in pulse mode 0.
Each time when an underflow occurs in timer A8, the contents which have been written to these bits are output from the corresponding pulse output pins **(Note)**.
These bits are invalid in pulse mode 1.

(7) $RTP3_2$, $RTP3_3$ pulse output data bits (bits 6, 7 at address $A4_{16}$)

Each time when an underflow occurs in timer A8, the contents which have been written to these bits are output from the corresponding pulse output pins **(Note)**.

Note: The output level at a pulse output pin is undefined in the period from when data is written to these bits until the first occurrence of a pulse output trigger. If it is necessary to avoid this state, perform "Processing of avoiding undefined output before starting pulse output" in Figure 9.4.2.

Table 9.3.4 Pulse-width-modulation-relevant bits

Pulse output pins where pulse width modulation is to be applied (Timers used for pulse width modulation)			Pulse width modulation timer select bits (bits 5, 4 at address $A0_{16}$)	Pulse width modulation enable bit 2 (bit 2 at address $A4_{16}$)	Pulse width modulation enable bit 1 (bit 1 at address $A4_{16}$)	Pulse width modulation enable bit 0 (bit 0 at address $A4_{16}$)
Pulse mode 0	4 pins	$RTP2_3$ to $RTP2_0$ (Timer A6)	00	×	×	1
Pulse mode 1	6 pins	$RTP3_1$, $RTP3_0$, $RTP2_3$ to $RTP2_0$ (Timer A6)	00	×	×	1
	In a unit of 3 pins	$RTP3_1$, $RTP3_0$, $RTP2_3$ (Timer A7)	01	×	1	×
		$RTP2_2$ to $RTP2_0$ (Timer A6)		×	×	1
	In a unit of 2 pins	$RTP3_1$, $RTP3_0$ (Timer A9)	10	1	×	×
		$RTP2_3$, $RTP2_2$ (Timer A7)		×	1	×
		$RTP2_1$, $RTP2_0$ (Timer A6)		×	×	1

X: It may be either "0" or "1."

9.3 Block description of pulse output port 1

9.3.3 Port P5 direction register

The pulse output trigger input pin is multiplexed with port $P5_2$ pin.

When using pin $P5_2$/RTP_{TRG1} as a pulse output trigger input pin, be sure to clear the port P5 direction register's bit, corresponding to port $P5_2$ pin, in order to set this port $P5_2$ pin for the input mode.

Figure 9.3.4 shows the relationship between port P5 direction register and a pulse output trigger input pin.

Port P5 direcition register (Address D_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin	Function	At reset	R/W
0	Nothing is assigned.		Undefined	—
1	Pin $\overline{INT_1}$	0 : Input mode 1 : Output mode When using this pin as a pulse output trigger input pin, be sure to clear the corresponding bit to "0."	0	RW
2	Pin RTP_{TRG1} (Pin $\overline{INT_2}$)		0	RW
3	Pin $\overline{INT_3}$/RTP_{TRG0}		0	RW
4	Nothing is assigned.		Undefined	—
5	Pin $\overline{INT_5}$/$TB0_{IN}$/IDW	0 : Input mode 1 : Output mode	0	RW
6	Pin $\overline{INT_6}$/$TB1_{IN}$/IDV		0	RW
7	Pin $\overline{INT_7}$/$TB2_{IN}$/IDU		0	RW

Note: ( ) shows the I/O pin of another internal peripheral device which is multiplexed.

Fig. 9.3.4 Relationship between port P5 direction register and pulse output trigger input pin

9.3.4 Timers A5 to A9

Timers A5 and A8 are used as control registers; each generates a pulse output trigger. When using timers A5 and A8, be sure to use them in the timer mode. (Refer to section **"7.3 Timer mode."**)
When performing the pulse width modulation, be sure to use timers A6, A7, A9 in the pulse width modulation mode. (Refer to section **"7.6 Pulse width modulation (PWM) mode."**) Note that, from pin $P2_2/TA9_{OUT}$, a PWM pulse by timer A9 is output. When it is unnecessary to output a PWM pulse, be sure to clear bit 2 of the timer A9 mode register (address DA_{16}) to "0." At this time, pin $P2_2$ can be used as a programmable I/O port pin.
Figure 9.3.5 shows the structure of the timer A5 and A8 mode registers (pulse output port 1); Figure 9.3.6 shows the structures of the timer A6, A7, A9 mode registers (pulse output port 1 with pulse width modulation used).

Timer A5 mode register (Address $D6_{16}$)
Timer A8 mode register (Address $D9_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0	0	0	0

Bit	Bit name	Functions	At reset	R/W
0	Fix these bits to "000000_2" in the pulse output port mode.		0	RW
1			0	RW
2			0	RW
3			0	RW
4			0	RW
5			0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 9.3.5 Structure of timer A5 and A8 mode registers (pulse output port 1)

9.3 Block description of pulse output port 1

Timer A6 mode register (Address $D7_{16}$)
Timer A7 mode register (Address $D8_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0	0	1	1

Bit	Bit name	Functions	At reset	R/W
0	Fix these bits to "00011_2" in the pulse output port mode.		0	RW
1			0	RW
2			0	RW
3			0	RW
4			0	RW
5	16/8-bit PWM mode select bit	0 : 16-bit pulse width modulator 1 : 8-bit pulse width modulator	0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Timer A9 mode register (Address DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0
			0	0		1	1

Bit	Bit name	Functions	At reset	R/W
0	Fix these bits to "11_2" in the pulse output port mode.		0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output (TA9OUT pin functions as a programmable I/O port pin.) 1 : Pulse output (TA9OUT pin functions as a PWM pulse output pin.)	0	RW
3	Fix these bits to "00_2" in the pulse output port mode.		0	RW
4			0	RW
5	16/8-bit PWM mode select bit	0 : 16-bit pulse width modulator 1 : 8-bit pulse width modulator	0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 9.3.6 Structures of timer A6, A7, A9 mode registers (pulse output port 1 with pulse width modulation used)

9.3.5 Pin $\overline{\text{P4OUT}_{\text{CUT}}}$ (pulse-output-cutoff signal input pin)

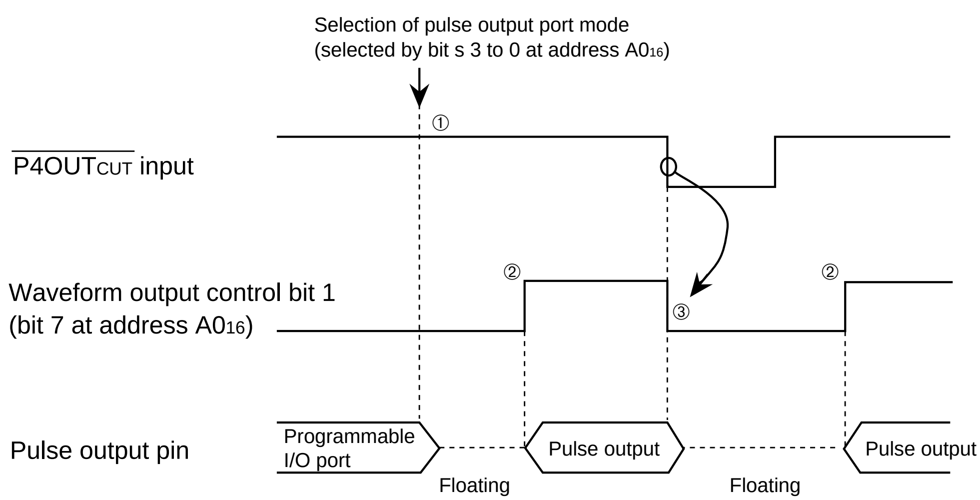
When a falling edge is input to pin $\overline{\text{P4OUT}_{\text{CUT}}}$, the waveform output control bit 1 (bit 7 at address $A0_{16}$) becomes "0" and the pulse output pins enter the floating state. (In other words, pulse output becomes disabled.) The pulse output pins where pulse output is to be inactive depend on the pulse output mode.

- Pulse mode 0: $RTP2_0$ to $RTP2_3$
- Pulse mode 1: $RTP2_0$ to $RTP2_3$, $RTP3_0$, $RTP3_1$

When restarting pulse output after the pulse output becomes inactive, be sure to return the input level at pin $\overline{\text{P4OUT}_{\text{CUT}}}$ to "H" level; and then, be sure to set the waveform output control bit 1 to "1." When the input level at pin $\overline{\text{P4OUT}_{\text{CUT}}}$ is "L" level, the waveform output control bit 1 cannot be "1."
Also, at this time, bits 0 through 7 of the port P4 direction register (address C_{16}) become "00000000_2." (Refer to section **"5.2.3 Pin $\overline{\text{P4OUT}_{\text{CUT}}}$/$\overline{\text{INT}_0}$."**) Therefore, if it is necessary to switch port pins $P6_0$ through $P6_7$ to port output pins, be sure to do as follows:

① Return the input level at pin $\overline{\text{P4OUT}_{\text{CUT}}}$ to "H" level.
② Write data to the port P4 register (address A_{16})'s bits, corresponding to the port P4 pins which will output data.
③ Set the port P4 direction register's bits, corresponding to the port P4 pins in ②, to "1" in order to set these port pins to the output mode.

When the input level at pin $\overline{\text{P4OUT}_{\text{CUT}}}$ is "L" level, no bit of the port P4 direction register can be "1."
Figure 9.3.7 shows the relationship between the $\overline{\text{P4OUT}_{\text{CUT}}}$ input, waveform output control bit 1, and pulse output pin.
Note that, when not making the pulse output inactive by using pin $\overline{\text{P4OUT}_{\text{CUT}}}$, be sure to connect pin $\overline{\text{P4OUT}_{\text{CUT}}}$ to Vcc via a resistor.



① When the pulse output port mode is selected, the pulse outpit pins become floating.
② The pulse is output by writing of "1" with the input level at pin $\overline{\text{P4OUT}_{\text{CUT}}}$ = "H."
③ When a falling edge is input to pin $\overline{\text{P4OUT}_{\text{CUT}}}$, this bit becomes "0."

Fig. 9.3.7 Relationship between $\overline{\text{P4OUT}_{\text{CUT}}}$ input, waveform output control bit 1, and pulse output pin

9.4 Setting of pulse output port mode

Figures 9.4.1 to 9.4.5 show an initial setting example for registers relevant to the pulse output port mode, where pins $RTP0_0$ to $RTP0_3$, $RTP1_0$, $RTP1_1$ are used as pulse output pins and an underflow of timer A0 is used as a pulse output trigger (pulse mode 1 of pulse output port 0). Note that when using interrupts, set up to enable the interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."** The above setting example is also applied to the case of pulse output port 1. Each right side of Figures 9.4.1 to 9.4.5 shows registers used in pulse output port 1.

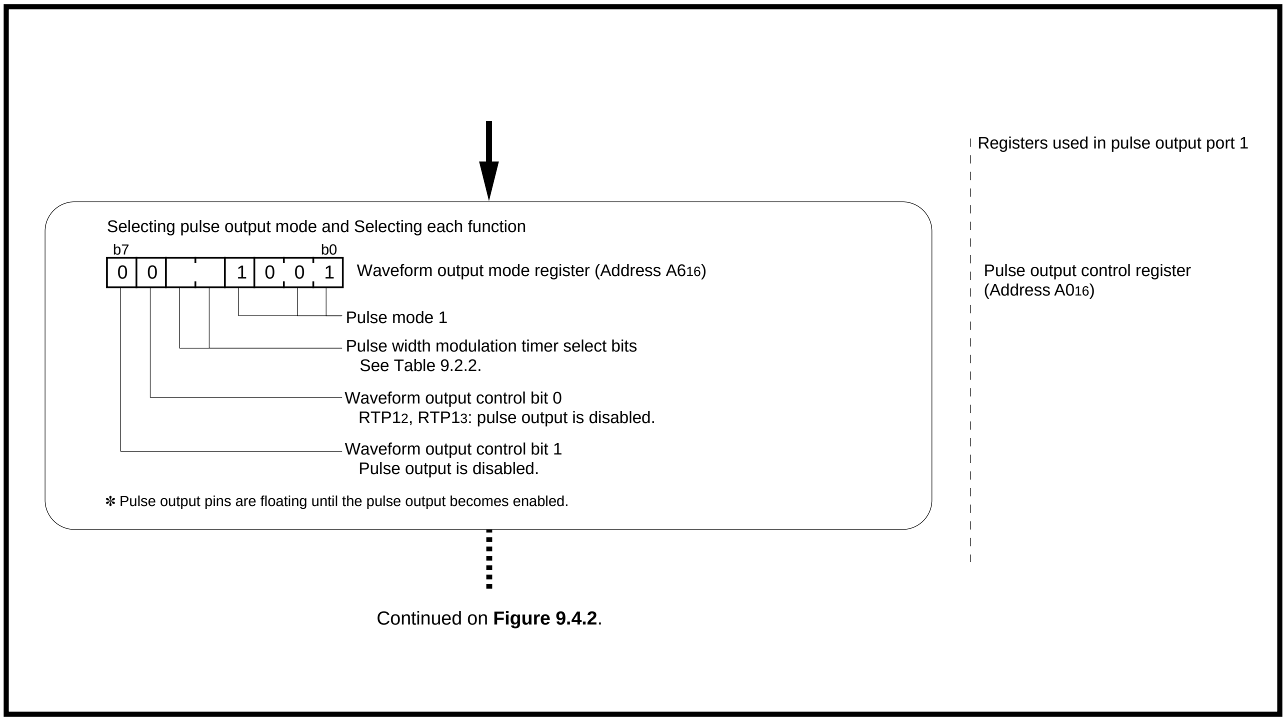


Fig. 9.4.1 Initial setting example for registers relevant to pulse output port 0 (pulse mode 1) (1)

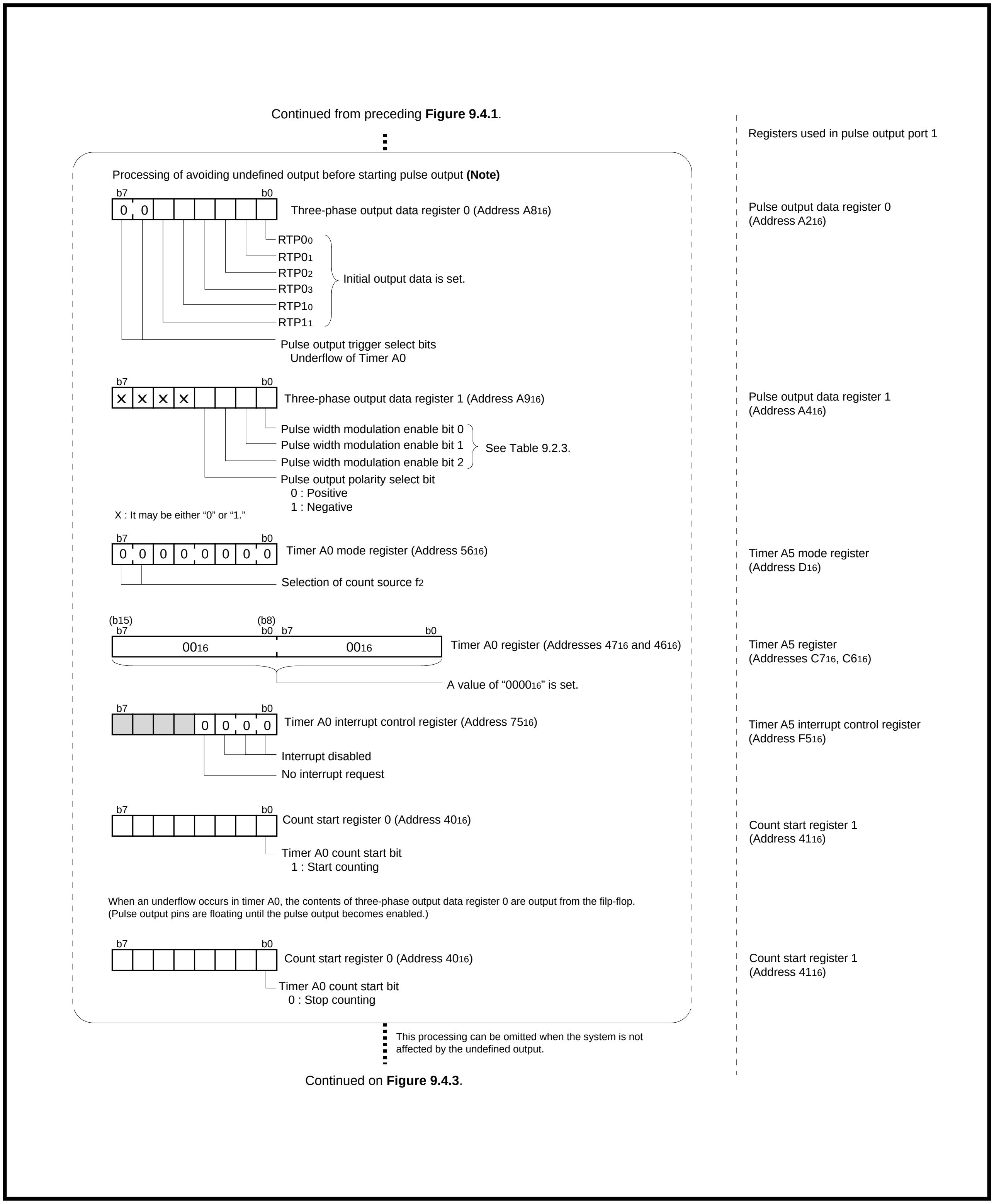


Fig. 9.4.2 Initial setting example for registers relevant to pulse output port 0 (pulse mode 1) (2)

9.4 Setting of pulse output port mode

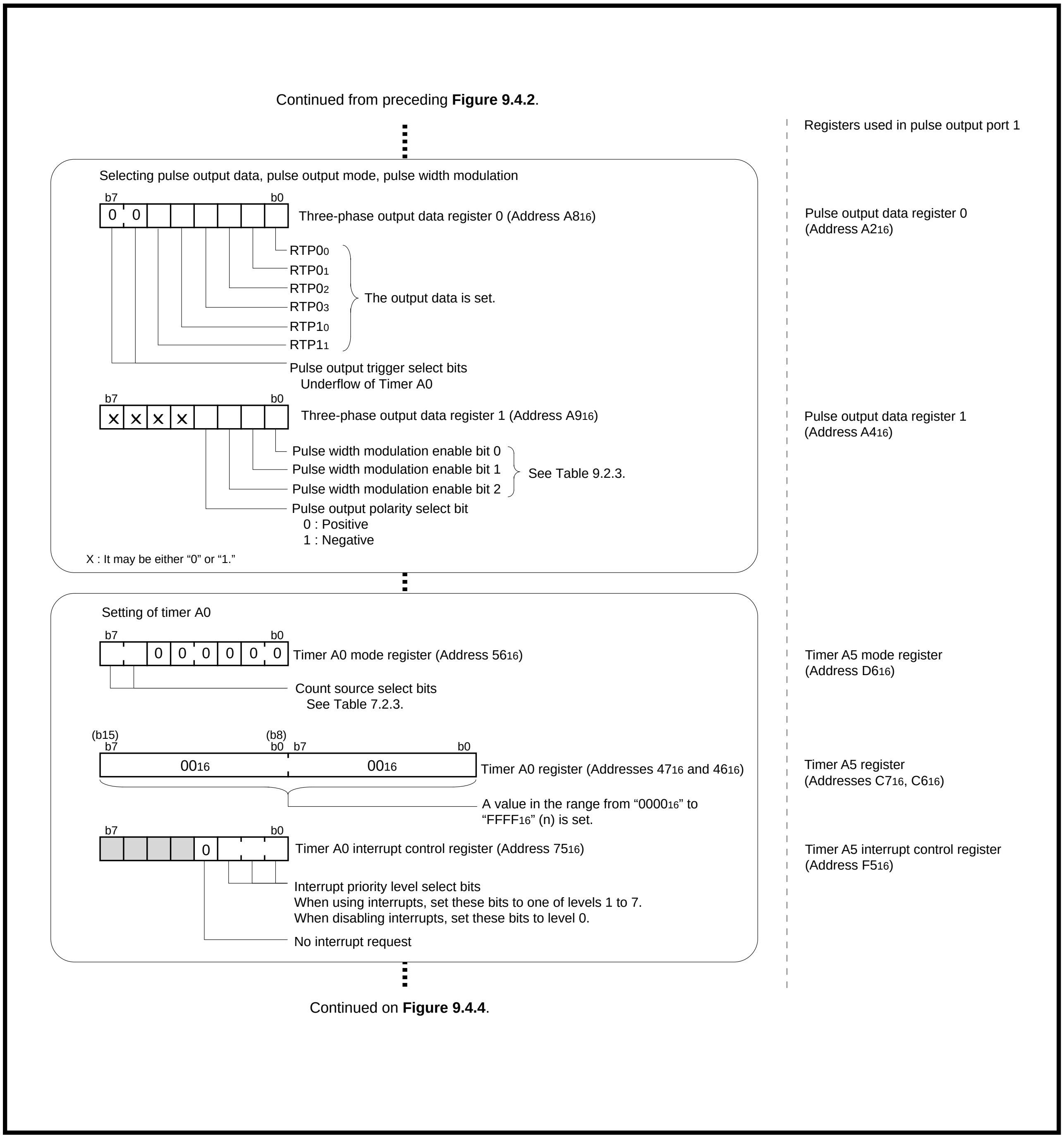


Fig. 9.4.3 Initial setting example for registers relevant to pulse output port 0 (pulse mode 1) (3)

Continued from preceding **Figure 9.4.3**.

When pulse width modulation is not performed

Registers used in pulse output port 1

When pulse width modulation is performed

b7 b0
| | | | 0 | 0 | | 1 | 1 |

Timer Aj mode register (j = 1, 2, 4) (Addresses 57_{16}, 58_{16}, $5A_{16}$)

j = 1, 2 : Fix this bit to "0.
j = 4 : When not using pin $TA4_{OUT}$ (in other words, when using pin $P2_0$ as a programmable I/O port pin), clear this bit to "0."

16/8-bit PWM mode select bit
0 : 16-bit pulse width modulator
1 : 8-bit pulse width modulator

Count source select bit
See Table 7.2.3.

Timer Ak mode register
(k = 6, 7, 9)
(Addresses $D7_{16}$, $D8_{16}$, DA_{16})

Setting of PWM pulse period and "H" level width

■When 16-bit pulse width modulator

(b15) b7 (b8) b0 b7 b0

Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

A value in the range from "0000_{16}" to "$FFFE_{16}$" (n) is set.

Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

■ When 8-bit pulse width modulator

(b15) b7 (b8) b0 b7 b0

Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

A value in the range from "00_{16}" to "FF_{16}" (m) is set.

A value in the range from "00_{16}" to "FE_{16}" (n) is set.

Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

✻ When operating as 8-bit pulse width modulator

Period = $\frac{(m + 1)(2^8 - 1)}{f_i}$

"H" level width = $\frac{n\,(m + 1)}{f_i}$

f_i : Frequency of count source

However, if n = "00_{16}," the pulse width modulator does not operate and pin TAj_{OUT} pin outputs "L" level. At this time, no timer Aj interrupt request occurs.

✻ When operating as 16-bit pulse width modulator

Period = $\frac{2^{16} - 1}{f_i}$

"H" level width = $\frac{n}{f_i}$

f_i : Frequency of count source

However, if n = "0000_{16}," the pulse width modulator does not operate and pin TAj_{OUT} pin outputs "L" level. At this time, no timer Aj interrupt request occurs.

Continued on **Figure 9.4.5**.

Fig. 9.4.4 Initial setting example for registers relevant to pulse output port 0 (pulse mode 1) (4)

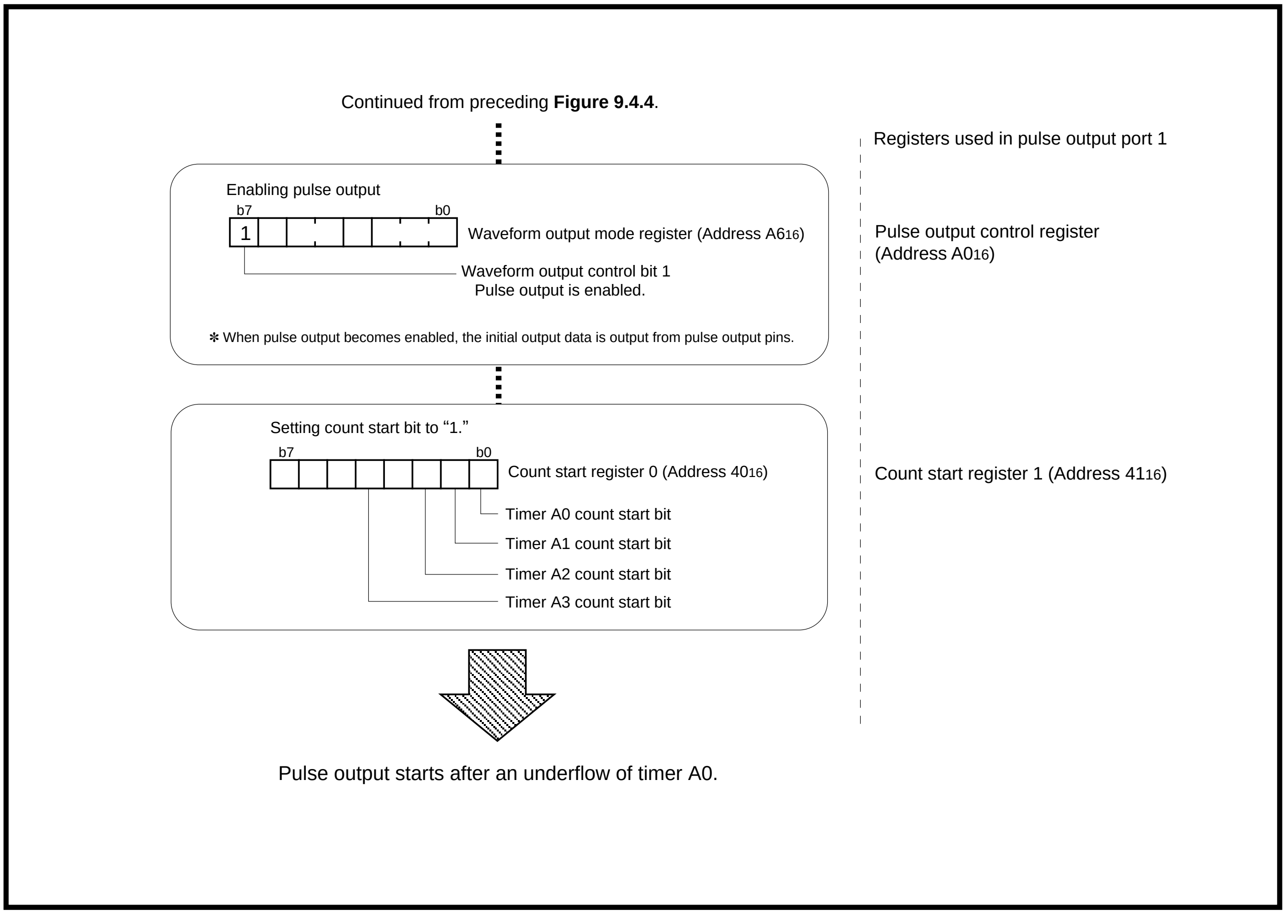


Fig. 9.4.5 Initial setting example for registers relevant to pulse output port 0 (pulse mode 1) (5)

9.5 Pulse output port mode operation

The operation of pulse output port 0 is described below and is also applied to the operation of pulse output port 1.

9.5.1 Pulse output trigger

(1) $RTP0_0$ to $RTP0_3$ in pulse mode 0; $RTP0_0$ to $RTP0_3$, $RTP1_0$, $RTP1_1$ in pulse mode 1

The pulse output trigger can be selected from an internal trigger and an external trigger.
When the pulse output trigger select bits (bits 7, 6 at address $A8_{16}$) = "00_2," an internal trigger is selected; when these bits = "01_2," "10_2," or "11_2," an external trigger is selected.

■ **Internal trigger**

A trigger occurs at an underflow of timer A0. This trigger occurrence can be confirmed by using the timer A0 interrupt request bit.

■ **External trigger**

A trigger occurs at a valid edge input to pin RTP_{TRG0} **(Note)**. This trigger occurrence can be confirmed by using the $\overline{INT_3}$ interrupt request bit. Table 9.5.1 lists the setting of $\overline{INT_3}$ according to valid edges.

When using pin $P5_3$/RTP_{TRG0} as an input pin for an external trigger, be sure to clear the port P5 direction register's bit, corresponding to port $P5_3$ pin, in order to set the port $P5_3$ pin to the input mode.

Note: This is set by the pulse output trigger select bits (bits 7, 6 at address $A8_{16}$).

Table 9.5.1 Setting of $\overline{INT_3}$ according to valid edges

Valid edge input to pin RTP_{TRG0}	Setting of $\overline{INT_3}$ **(Note)**
Falling	Falling (edge sense)
Rising	Rising (edge sense)
Falling and Rising	Falling and Rising (edge sense): used alternately

Note: Refer to section **"6.10 External interrupts."**

(2) $RTP1_0$ to $RTP1_3$ in pulse mode 0; $RTP1_2$, $RTP1_3$ in pulse mode 1

The pulse output trigger is an internal trigger.
A trigger occurs at an underflow of timer A3. This trigger occurrences can be confirmed by using the timer A3 interrupt request bit.

9.5.2 Operation at internal trigger

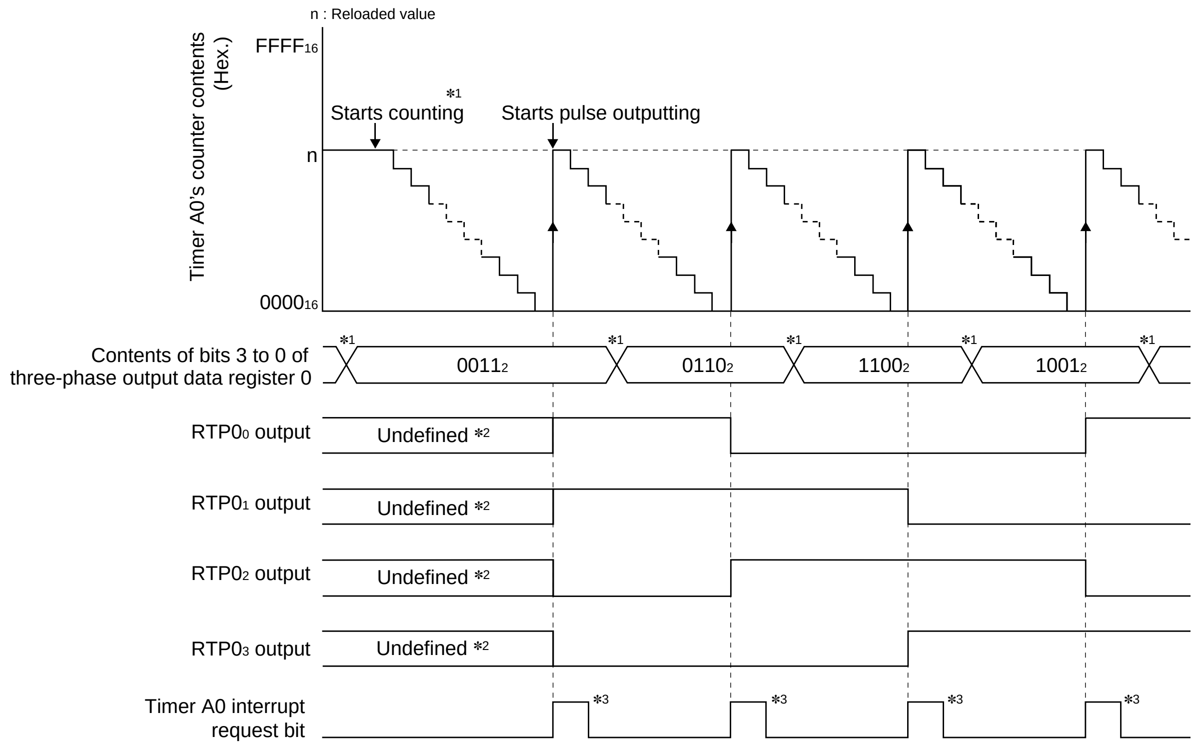
① When the timer Ai (i = 0, 3) count start bit is set to "1," the counter starts counting of a count source.

② The contents of the pulse output data bits of three-phase output data registers 0, 1 are output from the corresponding pulse output pins at each underflow of timer Ai. While the pulse width modulation is selected, the pulse width modulation is performed for "H" level output.
The timer reloads the contents of the reload register and continues counting.

③ The timer Ai interrupt request bit is set to "1" when the counter underflows in ②. The interrupt request bit retains "1" until the interrupt request is accepted or it is cleared to "0" by software.

④ Write the next output data into three-phase output data registers 0, 1 during a timer Ai interrupt routine (or after the confirmation of a timer Ai interrupt request occurrence.)

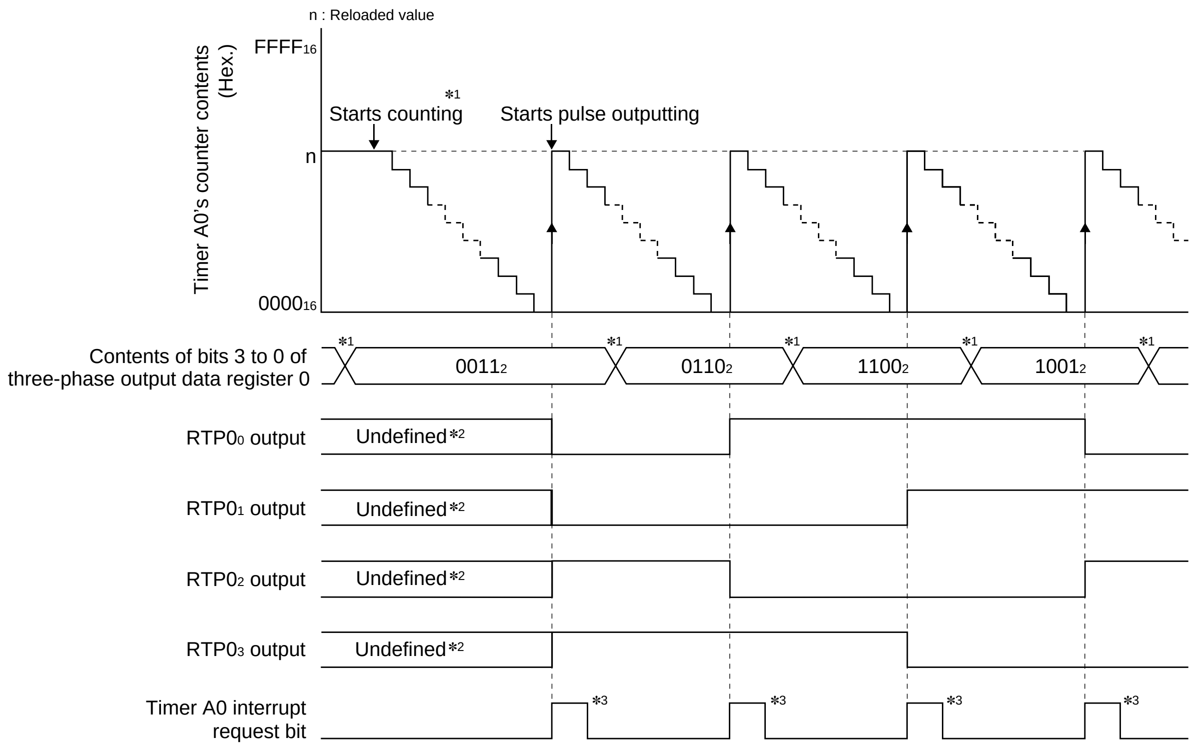
Figures 9.5.1 to 9.5.3 show examples of pulse output port mode operations.



✻1 : Written by software
✻2 : When avoiding undefined output in these terms (in other words, when stabilizing these output level), be sure to follow the procedure "Processing of avoiding undefined output before starting pulse output" in Figure 9.4.2.
✻3 : Cleared to "0" by an interrupt request acceptance or cleared by software.

The above applies when the following conditions are satisfied:
- Pulse mode 0 selected
- $RTP0_0$ to $RTP0_3$ selected
- No pulse width modulation
- Positive polarity

Fig. 9.5.1 Example of pulse output port mode operation (1)

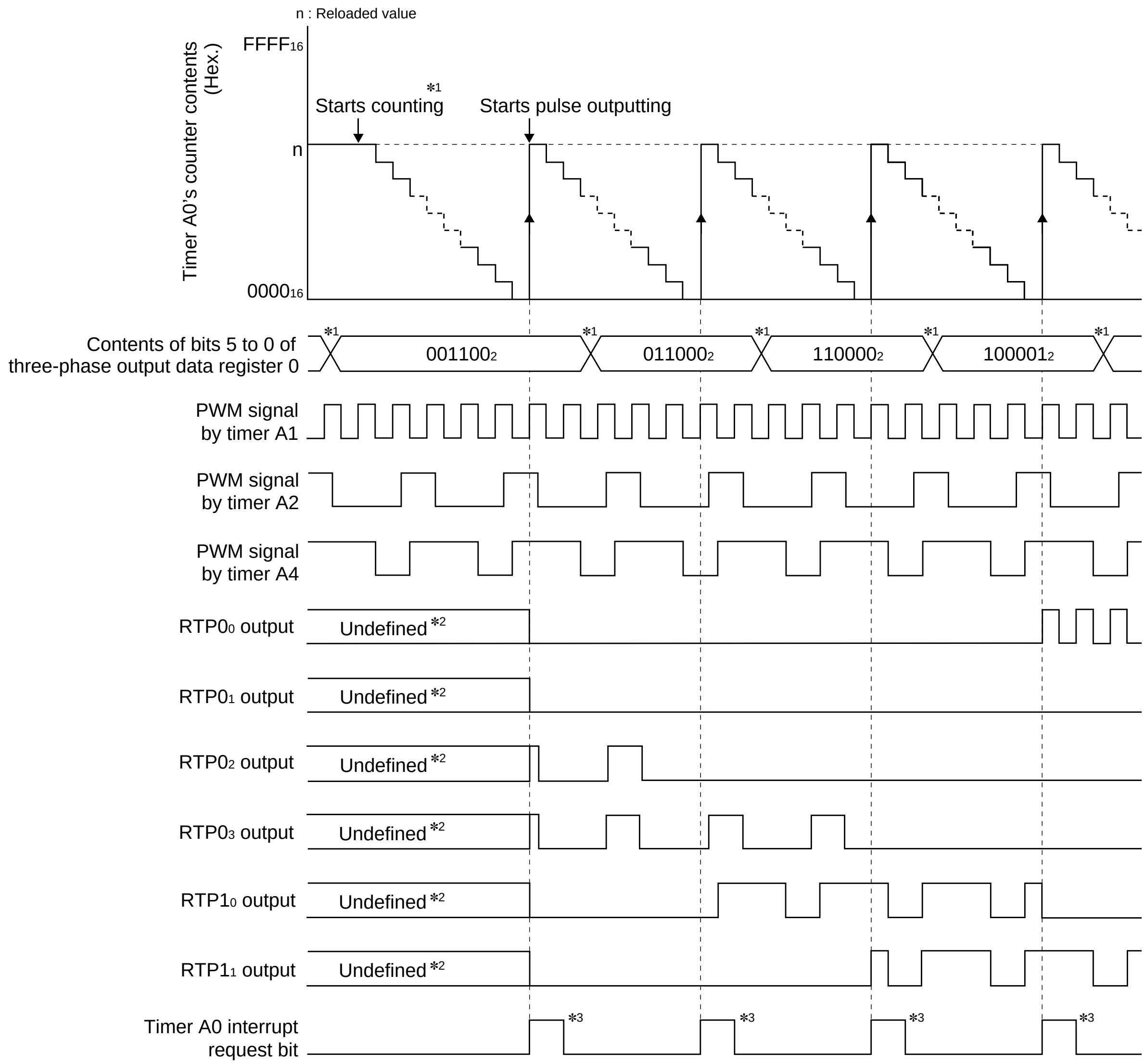


✻1 : Written by software
✻2 : When avoiding undefined output in these terms (in other words, when stabilizing these output level), be sure to follow the procedure "Processing of avoiding undefined output before starting pulse output" in Figure 9.4.2.
✻3 : Cleared to "0" by an interrupt request acceptance or cleared by software.

The above applies when the following conditions are satisfied:
- Pulse mode 0 selected
- $RTP0_0$ to $RTP0_3$ selected
- No pulse width modulation
- Negative polarity

Fig. 9.5.2 Example of pulse output port mode operation (2)

9.5 Pulse output port mode operation



✻1 : Written by software
✻2 : When avoiding undefined output in these terms (in other words, when stabilizing these output level), be sure to follow the procedure "Processing of avoiding undefined output before starting pulse output" in Figure 9.4.2.
✻3 : Cleared to "0" by an interrupt request acceptance or cleared by software.

The above applies when the following conditions are satisfied:
- Pulse mode 1 selected
- Pulse width modulation applied (in a unit of 2 pins; timers A1, A2, and A4 are used.)
- Positive polarity

Fig. 9.5.3 Example of pulse output port mode operation (3)

9.5.3 Operation at external trigger

① Each time when a valid edge of a signal input to pin RTP_{TRG0} **(Note)** is input, the contents of the pulse output data bits of three-phase output data register 0 are output from the corresponding pulse output pins. When the pulse width modulation is selected, the pulse width modulation is applied to "H" level output.

② The $\overline{INT_3}$ interrupt request bit is set to "1" when a valid edge (①) is input. (Refer to section **"9.5.1 Pulse output trigger."**) The interrupt request bit retains "1" until the interrupt request is accepted or it is cleared by software.

③ Write the next output data into three-phase output data register 0 during an $\overline{INT_3}$ interrupt routine (or after the confirmation of an $\overline{INT_3}$ interrupt request occurrence).

Note: This is set by the pulse output trigger select bits (bits 7, 6 at address $A8_{16}$).

[Precautions for pulse output port mode]

1. When using pulse output port 0, be sure to set the relevant registers after setting the waveform output select bits (bits 2 to 0 at address $A6_{16}$).
   When not using pulse output port 0 and three-phase waveform mode, be sure to fix the waveform output select bits (bits 2 to 0 at address $A6_{16}$) to “000_2.”

2. When using pulse output port 1, be sure to set the relevant registers after setting the waveform output select bits (bits 2 to 0 at address $A0_{16}$).
   When not using pulse output port 1, be sure to fix the waveform output select bits (bits 2 to 0 at address $A0_{16}$) to “000_2.”

3. When performing the pulse width modulation in pulse output port 0, be sure to use timers A1, A2, A4 in the pulse width modulation mode. (Refer to section **“7.6 Pulse width modulation (PWM) mode.”**) Note that, from pin $P2_0$/$TA4_{OUT}$, a PWM pulse by timer A4 is output. When it is unnecessary to output a PWM pulse, be sure to clear bit 2 of the timer A4 mode register (address $5A_{16}$) to “0.” At this time, pin $P2_0$ can be used as a programmable I/O port pin.

4. When performing the pulse width modulation in pulse output port 1, be sure to use timers A6, A7, A9 in the pulse width modulation mode. (Refer to section **“7.6 Pulse width modulation (PWM) mode.”**) Note that, from pin $P2_2$/$TA9_{OUT}$, a PWM pulse by timer A9 is output. When it is unnecessary to output a PWM pulse, be sure to clear bit 2 of the timer A9 mode register (address DA_{16}) to “0.” At this time, pin $P2_2$ can be used as a programmable I/O port pin.

5. Note that, when not making the pulse output inactive by input of a falling edge to pin $\overline{P6OUT_{CUT}}$ or $\overline{P4OUT_{CUT}}$, be sure to connect pin $\overline{P6OUT_{CUT}}$ or $\overline{P4OUT_{CUT}}$ to Vcc via a resistor.

CHAPTER 10

THREE-PHASE WAVEFORM MODE

10.1 Overview

The three-phase waveform mode serves as follows: three-phase waveforms (3 positive waveforms and 3 negative waveforms) are output from the three-phase waveform output pins. The three-phase waveform mode consists of “three-phase mode 0” and “three-phase mode 1.”
Table 10.1.1 lists the specifications of the three-phase waveform mode, Table 10.1.2 lists the comparison of operations in three-phase mode 0 and 1, and Figure 10.1.1 shows the comparison of waveforms in three-phase mode 0 and 1.

Table 10.1.1 Specifications of three-phase waveform mode

Item	Specifications	
Three-phase waveform output pins	6 pins (U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$)	
Three-phase-waveform-output-forcibly-cutoff signal input pin	$\overline{P6OUT_{CUT}}$ (Input of falling edge)	
Operation modes	Three-phase mode 0	A timer A3 interrupt request occurs at each timer A3 underflow.
	Three-phase mode 1	A timer A3 interrupt request occurs at each second timer A3 underflow or forth one.
Timer to be used	Timers A0 through A2 (Used in the one-shot pulse mode) • Timer A0 : W- and $\overline{W}$-phase waveform control • Timer A1 : V- and $\overline{V}$-phase waveform control • Timer A2 : U- and $\overline{U}$-phase waveform control Timer A3 (Used in the timer mode) • Output period control	
Three-phase waveform period	$\frac{1}{f_1}$ to $\frac{1}{f_{4096}}$ $\times$ 65536	
Output waveform and Output width	Saw-tooth-wave modulation output	$\frac{1}{f_1}$ to $\frac{1}{f_{4096}}$ $\times$ 65535 **(Note)**
	Triangular wave modulation output	$\frac{1}{f_1}$ $\times$ 2 to $\frac{1}{f_{4096}}$ $\times$ 65535 $\times$ 2 **(Note)**
	Fixed level output	Each of the U, V, W phases is fixed to an arbitrary level. Each of the $\overline{U}$, $\overline{V}$, $\overline{W}$ phases is fixed to the reversed level of the corresponding positive phase (the U, V, W phases).
Dead time (width)	Dead-time timer is used. See Table 10.2.1.	

Note: This value does not include the dead time.

Table 10.1.2 Comparison of operations in three-phase mode 0 and 1

	Three-phase mode 0	Three-phase mode 1
Timer A3 interrupt request occurrence interval	Each timer A3 underflow	Each second timer A3 underflow or forth one is selected by software.
Timers A0 through A2	Each timer uses one register.	Each timer uses two registers alternately.
Output polarity	• By software, the output polarity can be set to the output polarity set buffer of the U, V, or W phases. • If necessary, the contents of each output polarity set buffer are reversed by software.	• By software, the output polarity can be set to the three-phase output polarity set buffer. • At each period, the contents of the three-phase output polarity set buffer are reversed by hardware.

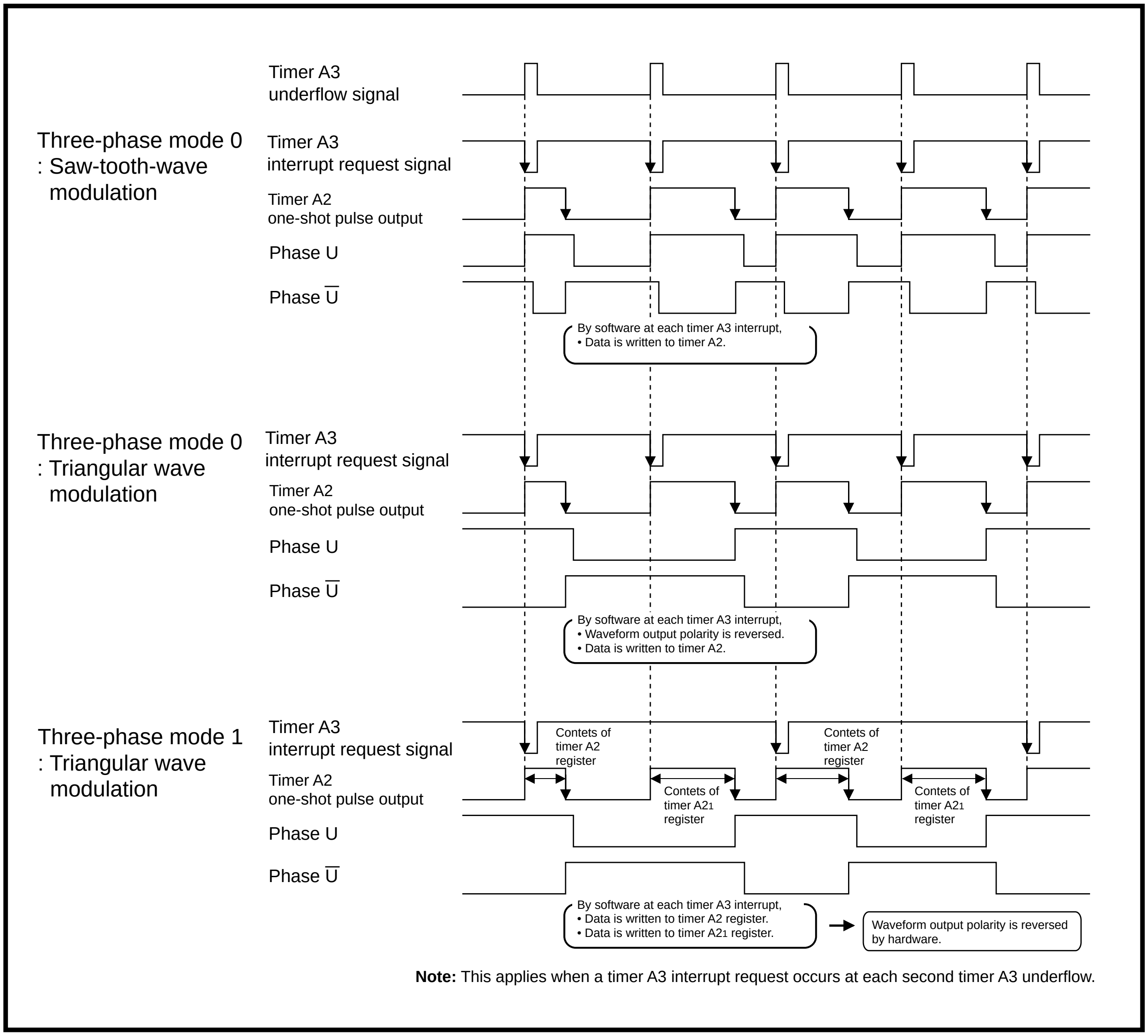


Fig. 10.1.1 Comparison of waveforms in three-phase mode 0 and 1

10.2 Block description

Figure 10.2.1 shows the block diagram of the three-phase waveform mode, and explanation of registers relevant to the three-phase waveform mode is described below.

The following registers are common to pulse output port 0 and three-phase waveform mode:

- Waveform output mode register (address $A6_{16}$)
- Three-phase output data register 0 (address $A8_{16}$)
- Three-phase output data register 1 (address $A9_{16}$)

When using the three-phase waveform mode, be sure to fix the waveform output select bits (bits 2 through 0 at address $A6_{16}$) to "100_2," and then, set the relevant registers.

When not using pulse output port 0 and three-phase waveform mode, be sure to fix the waveform output select bits (bits 2 through 0 at address $A6_{16}$) to "000_2."

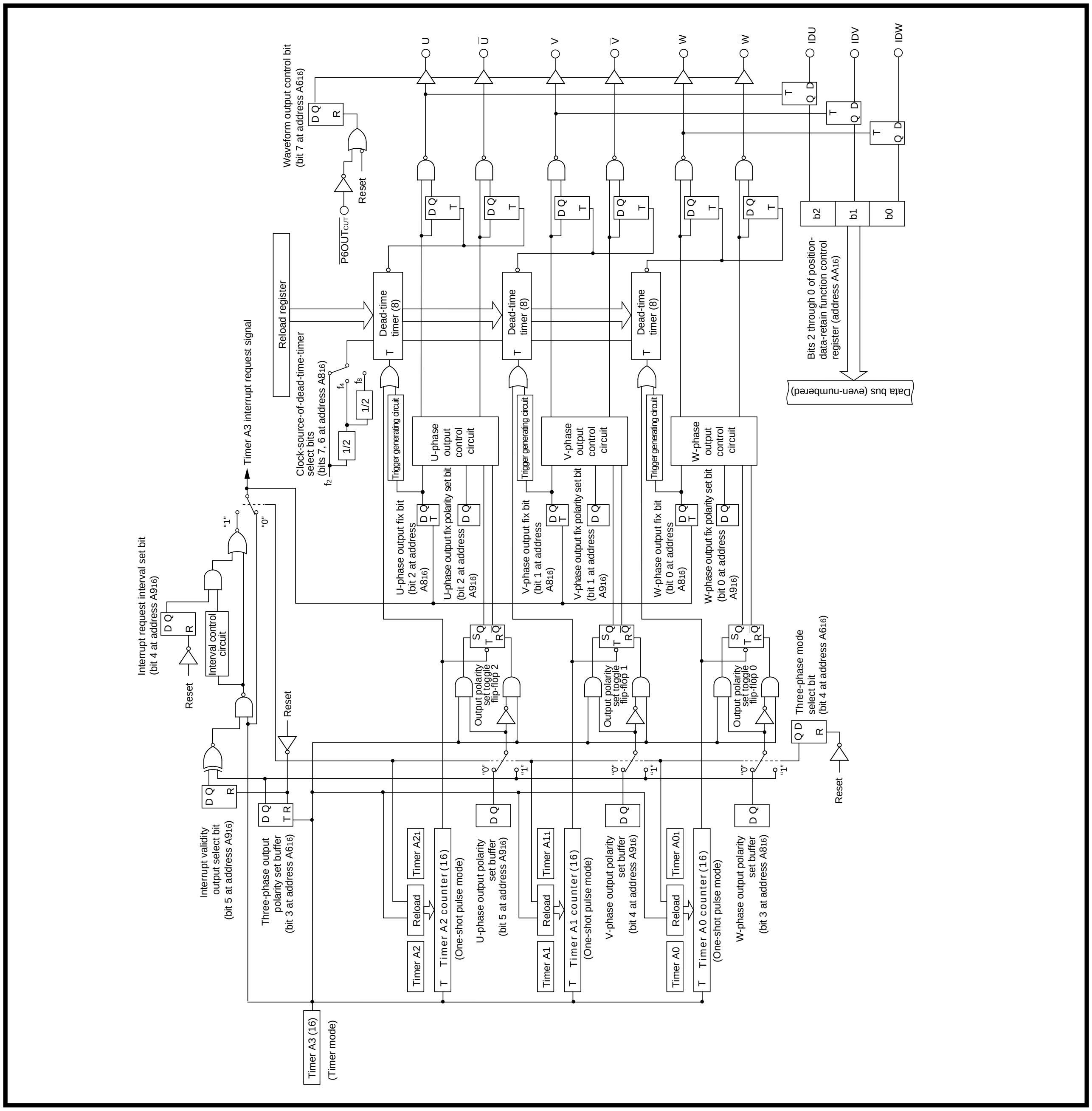


Fig. 10.2.1 Block diagram of three-phase waveform mode

10.2.1 Waveform output mode register

Figure 10.2.2 shows the structure of the waveform output mode register (the three-phase waveform mode). Note that writing to bits 0 through 6 of this register must be performed when the counting in timers A0 through A3 is halts.

Waveform output mode register (Address $A6_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		X			1	0	0

Bit	Bit name	Function	At reset	R/W
0	Waveform output select bits **(Note 1)**	b2 b1 b0 1 0 0 : Three-phase waveform mode	0	RW
1			0	RW
2			0	RW
3	Three-phase output polarity set buffer (Valid in three-phase mode 1) **(Note 2)**	0 : "H" output 1 : "L" output	0	RW
4	Three-phase mode select bit	0 : Three-phase mode 0 1 : Three-phase mode 1	0	RW
5	Invalid in the three-phase waveform mode.		0	RW
6	Dead-time timer trigger select bit **(Note 3)**	0: Both falling and rising edges of one-shot pulse for timers A0 to A2 1: Only the falling edge of one-shot pulse for timers A0 to A2	0	RW
7	Waveform output control bit	0 : Waveform output disabled 1 : Waveform output enabled	0	RW

X: It may be either "0" or "1."

Notes 1: When not using pulse output port 0 and three-phase waveform mode, be sure to fix these bits to "000_2."
2: This bit is invalid in three-phase mode 0.
3: When the saw-tooth-wave modulation output is performed, be sure to fix this bit to "0."
4: Writing to any of bits 0 to 6 must be performed while counting for timers A0 to A3 halts.

Fig. 10.2.2 Structure of waveform output mode register (three-phase waveform mode)

10.2 Block description

(1) Three-phase output polarity set buffer (bit 3)
This bit serves as the buffer to set the output polarity of the three-phase waveform and is used in three-phase mode 1. (Refer to section **"10.2.9 Output polarity set toggle flip-flop."**)

(2) Three-phase mode select bit (bit 4)
This bit is used to select three-phase mode 0 or 1.

(3) Dead-time timer trigger select bit (bit 6)
This bit is used to select a trigger of the dead-time timer.
The saw-tooth-wave modulation requires that this bit is fixed to "0."

(4) Waveform output control bit (bit 7)
Setting of this bit to "1" allows the three-phase waveform output from the three-phase waveform output pins. Clearance of this bit to "0" makes the three-phase waveform output pins floating.
When a falling edge is input to pin $\overline{P6OUT_{CUT}}$, this bit becomes "0." (See Figure 10.2.15.)

10.2.2 Dead-time timer register

Figure 10.2.3 shows the structure of the Dead-time timer register.

Dead-time timer (Address $A7_{16}$)

b7 b0

Bit	Function	At reset	R/W
7 to 0	A value in the range from "00_{16}" to "FF_{16}" can be set.	Undefined	WO

Note: Use the **MOVMB** (**MOVM** when m = 1) or **STAB** (**STA** when m = 1) instruction for writing to this register.
Additionally, make sure writing to this register does not overlap with a trigger-occurrence timing of the dead-time timer.

Fig. 10.2.3 Structure of dead-time timer register

The dead-time timer is used to count the time to prevent "L" level of positive waveform outputs from overlapping with "L" level of their negative waveform outputs. (This time is referred to as "dead time.")
Figure 10.2.4 shows the structure of the dead-time timer.

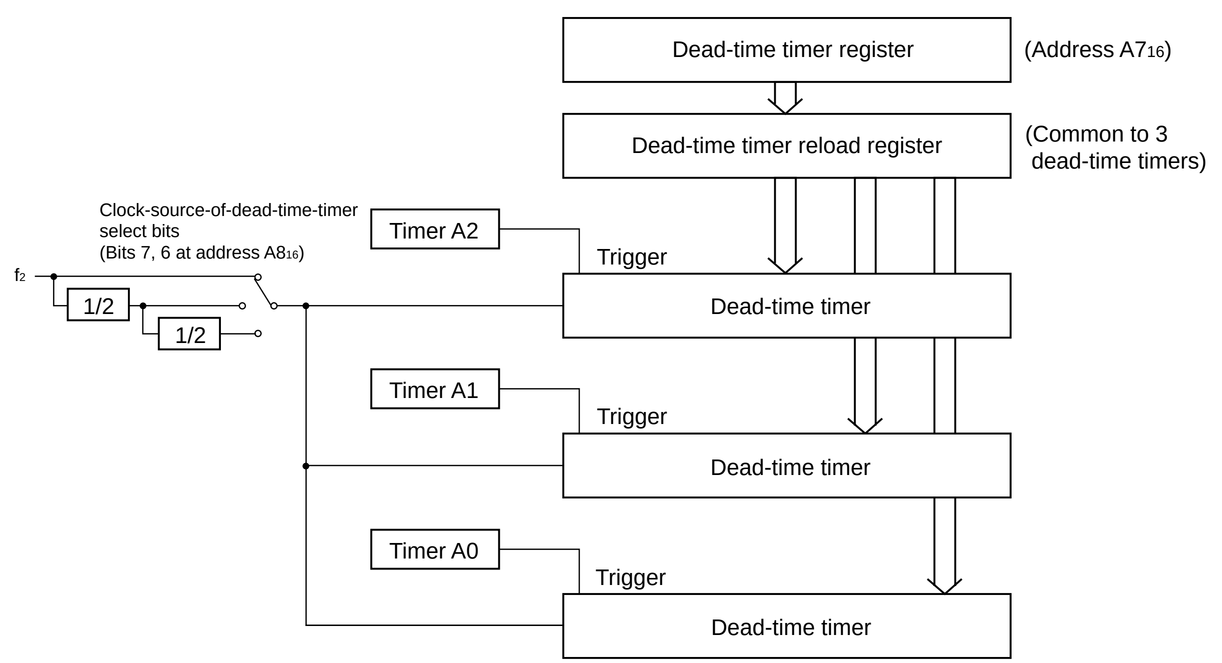


Fig. 10.2.4 Structure of dead-time timer

10.2 Block description

When a certain value is written to the dead-time timer register, this value is written to the dead-time reload register. The M37905 has three dead-time timers, and they are independent each other. When a trigger is generated due to each of timers A0 through A2, the contents of the dead-time timer reload register are reloaded; and then, the selected count source is counted down. Simultaneously, the one-shot pulse is output. A trigger is selected by the dead-time timer trigger select bit (bit 6 at address $A6_{16}$), and the count source is selected by the clock-source-of-dead-time-timer select bits (bits 7, 6 at address $A8_{16}$). When an underflow occurs, the counting becomes inactive.
Figure 10.2.5 shows the relationship between the dead-time timer's pulse and trigger, and Table 10.2.1 lists the pulse width of the dead-time timer.

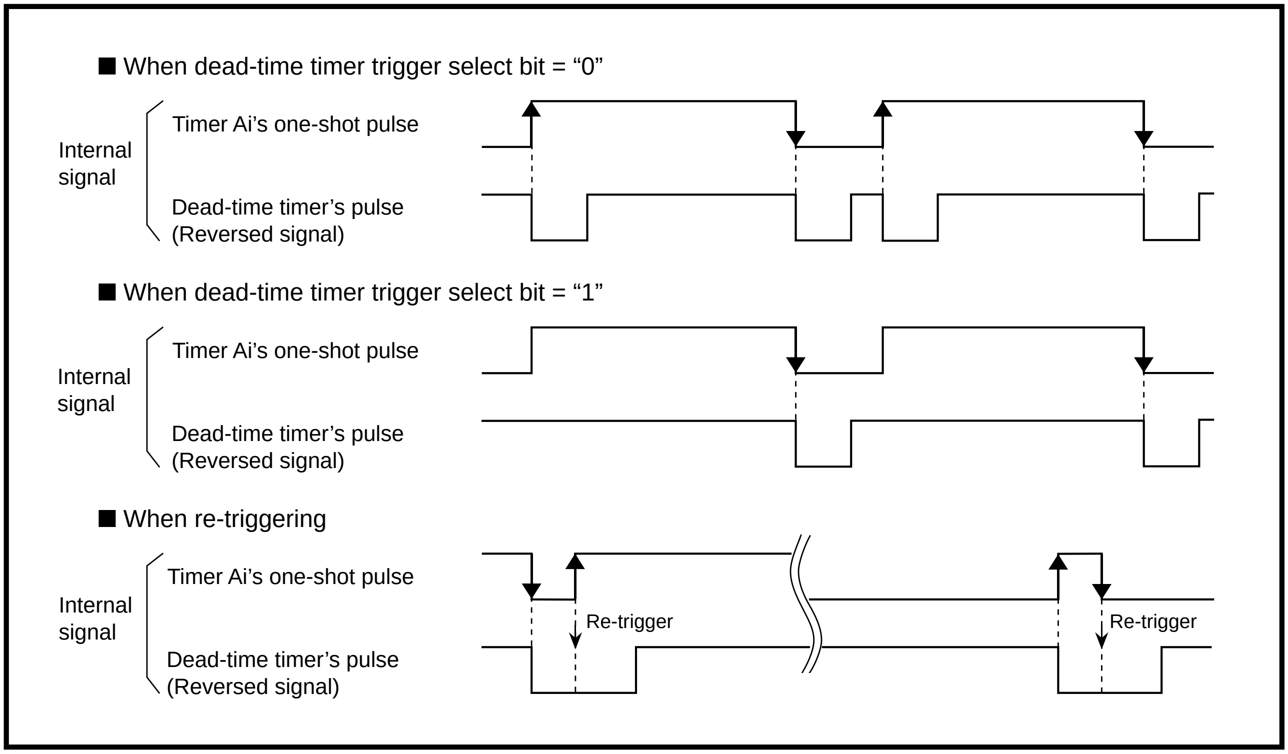


Fig. 10.2.5 Relationship between dead-time timer's pulse and trigger

Table 10.2.1 Pulse width of dead-time timer

Trigger		Pulse width	
State at trigger input	Edge	n : 00_{16}	n : 01_{16} through FF_{16}
Dead-time timer: inactive	Rising edge of timer Ai one-shot pulse	$258 \times \frac{1}{f_i}$	$(n+2) \times \frac{1}{f_i}$
	Falling edge of timer Ai one-shot pulse	$257 \times \frac{1}{f_i}$	$(n+1) \times \frac{1}{f_i}$
Dead-time timer: active	Rising edge of timer Ai one-shot pulse (Re-trigger)	$257 \times \frac{1}{f_i}$ **(Note)**	$(n+1) \times \frac{1}{f_i}$ **(Note)**
	Falling edge of timer Ai one-shot pulse (Re-trigger)		

n: A value which is set in the dead-time timer (address $A7_{16}$)
f_i: The dead-time timer's clock source (f_2, $f_2/2$, $f_2/4$)
Note: Width of pulse starting from a re-trigger occurrence timing

10.2.3 Three-phase output data register 0

Figure 10.2.6 shows the structure of the three-phase output data register 0 (the three-phase waveform mode).

For bits 7 and 6, refer to section **"10.2.2 Dead-time timer."**

Three-phase output data register 0 (Address $A8_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		X	X				

Bit	Bit name	Function	At reset	R/W
0	W-phase output fix bit	0 : Released from output fixation 1 : Output fixed	0	RW
1	V-phase output fix bit	0 : Released from output fixation 1 : Output fixed	0	RW
2	U-phase output fix bit	0 : Released from output fixation 1 : Output fixed	0	RW
3	W-phase output polarity set buffer (Valid in three-phase mode 0.) **(Note)**	0 : "H" output 1 : "L" output	0	RW
5, 4	Invalid in the three-phase waveform mode.		0	RW
6	Clock-source-of-dead-time-timer select bits	b7 b6 0 0 : f_2 0 1 : $f_2/2$ 1 0 : $f_2/4$ 1 1 : Do not select.	0	RW
7			0	RW

X: It may be either "0" or "1."

Note: This bit is invalid in three-phase mode 1.

Fig. 10.2.6 Structure of three-phase output data register 0 (three-phase waveform mode)

(1) W-phase output fix bit (bit 0)

Setting of this bit to "1" fixes the output level at the W-phase waveform output pin to the level which is selected by the W-phase fixed output's polarity set bit (bit 0 at address $A9_{16}$); vice versa, the output level at the $\overline{W}$-phase waveform output pin is reversed.

(2) V-phase output fix bit (bit 1)

Setting of this bit to "1" fixes the output level at the V-phase waveform output pin to the level which is selected by the V-phase fixed output's polarity set bit (bit 1 at address $A9_{16}$); vice versa, the output level at the $\overline{V}$-phase waveform output pin is reversed.

(3) U-phase output fix bit (bit 2)

Setting of this bit to "1" fixes the output level at the U-phase waveform output pin to the level which is selected by the U-phase fixed output's polarity set bit (bit 2 at address $A9_{16}$); vice versa, the output level at the $\overline{U}$-phase waveform output pin is reversed.

(4) W-phase output polarity set buffer (bit 3)

This bit serves as the buffer to set the W-phase output polarity and is used in three-phase mode 0. (Refer to section **"10.2.9 Output polarity set toggle flip-flop."**)

10.2.4 Three-phase output data register 1

Figure 10.2.7 shows the structure of the three-phase output data register 1 (the three-phase waveform mode).

Three-phase output data register 1 (Address $A9_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
X	X			X			

Bit	Bit name	Function	At reset	R/W
0	W-phase fixed output's polarity set bit **(Note 1)**	0 : "H" output fixed 1 : "L" output fixed	0	RW
1	V-phase fixed output's polarity set bit **(Note 2)**	0 : "H" output fixed 1 : "L" output fixed	0	RW
2	U-phase fixed output's polarity set bit **(Note 3)**	0 : "H" output fixed 1 : "L" output fixed	0	RW
3	Invalid in the three-phase waveform mode.		0	RW
4	V-phase output polarity set buffer (in three-phase mode 0)	0 : "H" output 1 : "L" output	0	RW
	Interrupt request interval set bit (in three-phase mode 1)	0 : Every second time 1 : Every forth time		
5	U-phase output polarity set buffer (in three-phase mode 0)	0 : "H" output 1 : "L" output	0	RW
	Interrupt validity output select bit (in three-phase mode 1)	0 : An interrupt request occurs at each even-numbered underflow of timer A3 1 : An interrupt request occurs at each odd-numbered underflow of timer A3		
7, 6	Invalid in the three-phase waveform mode.		0	RW

X: It may be either "0" or "1."

Notes 1: Valid when the W-phase output fix bit (bit 0 at address $A8_{16}$) = "1." Be sure not to change the value during output of a fixed value.
2: Valid when the V-phase output fix bit (bit 1 at address $A8_{16}$) = "1." Be sure not to change the value during output of a fixed value.
3: Valid when the U-phase output fix bit (bit 2 at address $A8_{16}$) = "1." Be sure not to change the value during output of a fixed value.

Fig. 10.2.7 Structure of three-phase output data register 1 (three-phase waveform mode)

(1) W-phase fixed output's polarity set bit (bit 0)

Clearance of this bit to "0" fixes the output level at the W-phase waveform output pin to "H"; vice versa, setting of this bit to "1" fixes the output level at the W-phase waveform output pin to "L."
The output level at the $\overline{W}$-phase waveform output pin is reversed.
Note that this bit is valid only when the W-phase output fix bit (bit 0 at address $A8_{16}$) = "1."

(2) V-phase fixed output's polarity set bit (bit 1)

Clearance of this bit to "0" fixes the output level at the V-phase waveform output pin to "H"; vice versa, setting of this bit to "1" fixes the output level at the V-phase waveform output pin to "L."
The output level at the $\overline{V}$-phase waveform output pin is reversed.
Note that this bit is valid only when the V-phase output fix bit (bit 1 at address $A8_{16}$) = "1."

(3) U-phase fixed output's polarity set bit (bit 2)

Clearance of this bit to "0" fixes the output level at the U-phase waveform output pin to "H"; vice versa, setting of this bit to "1" fixes the output level at the U-phase waveform output pin to "L."
The output level at the $\overline{U}$-phase waveform output pin is reversed.
Note that this bit is valid only when the U-phase output fix bit (bit 2 at address $A8_{16}$) = "1."

(4) V-phase output polarity set buffer (bit 4) (in three-phase mode 0)

This bit serves as the buffer to set the V-phase output polarity. (Refer to section **"10.2.9 Output polarity set toggle flip-flop."**)

Interrupt request interval set bit (bit 4) (in three-phase mode 1)

Clearance of this bit to "0" generates a timer A3 interrupt request at every second time; vice versa, setting of this bit to "1" generates a timer A3 interrupt request at every forth time.
(Refer to section **"10.4 Three-phase mode 1."**)

(5) U-phase output polarity set buffer (bit 5) (in three-phase mode 0)

This bit serves as the buffer to set the U-phase output polarity. (Refer to section **"10.2.9 Output polarity set toggle flip-flop."**)

Interrupt validity output select bit (bit 5) (in three-phase mode 1)

Clearance of this bit to "0" generates a timer A3 interrupt request at every even-numbered underflow of timer A3; vice versa, setting of this bit to "1" generates a timer A3 interrupt request at every odd-numbered underflow of timer A3.
(Refer to section **"10.4 Three-phase mode 1."**)

10.2.5 Position-data-retain function control register

Figure 10.2.8 shows the structure of the position-data-retain function control register.
For details of the position-data-retain function, refer to section **"10.6 Position-data-retain function."**

Position-data-retain function control register (Address AA_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	W-phase position data retain bit	Input level at pin IDW is read out. 0 : "L" level 1 : "H" level	0	RO
1	V-phase position data retain bit	Input level at pin IDV is read out. 0 : "L" level 1 : "H" level	0	RO
2	U-phase position data retain bit	Input level at pin IDU is read out. 0 : "L" level 1 : "H" level	0	RO
3	Retain-trigger polarity select bit	0 : Falling edge of positive phase 1 : Rising edge of positive phase	0	RW
7 to 4	Nothing is assigned.		Undefined	—

Note: This register is valid only in the three-phase mode.

Fig. 10.2.8 Structure of position-data-retain function control register

(1) W-phase position data retain bit (bit 0)
This bit is used to retain the input level at pin IDW.

(2) V-phase position data retain bit (bit 1)
This bit is used to retain the input level at pin IDV.

(3) U-phase position data retain bit (bit 2)
This bit is used to retain the input level at pin IDU.

(4) Retain-trigger polarity select bit (bit 3)
This bit is used to select the trigger polarity to retain the position data.
When this bit = "0," the falling edge of each positive phase is selected.
When this bit = "1," the rising edge of each positive phase is selected.

10.2.6 Port P5 direction register

The position-data input pins are multiplexed with port P5 pin.
When using these pins as position-data-input pins, clear the corresponding bits of the port P5 direction register to "0" in order to set these port pins for the input mode.
Figure 10.2.9 shows the relationship between the port P5 direction register and position-data-input pins.

Port P5 direction register (Address D_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin	Functions	At reset	R/W
0	Nothing is assigned.		Undefined	—
1	Pin $\overline{\text{INT}_1}$	0 : Input mode 1 : Output mode	0	RW
2	Pin $\overline{\text{INT}_2}$/RTP_{TRG1}		0	RW
3	Pin $\overline{\text{INT}_3}$/RTP_{TRG0}		0	RW
4	Nothing is assigned.		Undefined	—
5	Pin IDW (Pin $\overline{\text{INT}_5}$/TB0_{IN})	0 : Input mode 1 : Output mode When using this pin as a position-data input pin, be sure to clear the corresponding bit to "0."	0	RW
6	Pin IDV (Pin $\overline{\text{INT}_6}$/TB1_{IN})		0	RW
7	Pin IDU (Pin $\overline{\text{INT}_7}$/TB2_{IN})		0	RW

Note: The pins in ( ) are I/O pins of other internal peripheral devices, which are multiplexed.

Fig. 10.2.9 Relationship between port P5 direction register and position-data-input pins

10.2.7 Timers A0 through A2

Each of timers A0 through A2 is used to control the output width of each phase, and these timers are used in the one-shot pulse mode.
Figure 10.2.10 shows the structure of timer A0/A1/A2 mode register (in the three-phase waveform mode). Because the underflow signal of timer A3 serves as a trigger for timers A0 through A3, it is unnecessary to set the one-shot start bit to "1."
Note that, in three-phase mode 1, each of timers A0 through A2 has the following two registers: timer A0/A1/A2 register (addresses 46_{16} and 47_{16}, 48_{16} and 49_{16}, $4A_{16}$ and $4B_{16}$) and timer $A0_1$/$A1_1$/$A2_1$ register (addresses $D0_{16}$ and $D1_{16}$, $D2_{16}$ and $D3_{16}$, $D4_{16}$ and $D5_{16}$). These two registers are used to control the output width.
Figure 10.2.11 shows the structures of the timer A0/A1/A2 mode register and timer $A0_1$/$A1_1$/$A2_1$ register.

Timer A0/A1/A2 mode register (Addresses 56_{16} to 58_{16})

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	1	0	1	0

Bit	Bit name	Function	At reset	R/W
0	Fix these bits to "011010_2" in the three-phase waveform mode.		0	RW
1			0	RW
2			0	RW
3			0	RW
4			0	RW
5			0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 10.2.10 Structure of timer A0/A1/A2 mode register (three-phase waveform mode)

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)

(b15) b7 … (b8) b0 | b7 … b0

Bit	Function	At reset	R/W
15 to 0	Any value in the range from "0000_{16}" to "$FFFF_{16}$" can be set. Assuming that the set value = n, the "H" level width of the one-shot pulse is expressed as follows : $\frac{n}{f_i}$.	Undefined	WO

f_i: Frequency of count source
Note: Use the **MOVM** or **STA(STAD)** instruction for writing to this register.
Writing to this register must be performed in a unit of 16 bits.

Timer A0₁ register (Addresses $D1_{16}$, $D0_{16}$)
Timer A1₁ register (Addresses $D3_{16}$, $D2_{16}$)
Timer A2₁ register (Addresses $D5_{16}$, $D4_{16}$)

(b15) b7 … (b8) b0 | b7 … b0

Bit	Function	At reset	R/W
15 to 0	Any value in the range from 0000_{16} to $FFFF_{16}$ can be set. Assuming that the set value = n, the "H" level width of the one-shot pulse is expressed as follows: n/f_i.	Undefined	WO

f_i: Frequency of a count source

Notes 1: Use the **MOVM** or **STA (STAD)** instruction for writing to this register. Additionally, make sure writing to this register must be performed in a unit of 16 bits.
2: This register is valid only in three-phase mode 1 of the three-phase waveform mode.

Fig. 10.2.11 Structures of timer A0/A1/A2 register and timer $A0_1$/$A1_1$/$A2_1$ register

10.2.8 Timer A3

Timer A3 is used to control the carrier's period of the whole three-phase waveform and is used in the timer mode.

Note that a pulse is output, due to timer A3, from pin $P6_6$/$TA3_{OUT}$. (Refer to section **"7.3.3 Select function; (2) Pulse output function."**) When not outputing the pulse, be sure to clear bit 2 of the timer A3 mode register (address 59_{16}) to "0." At this time, pin $P6_6$ can be used as a programmable I/O port pin.

Figure 10.2.12 shows the structure of the timer A3 mode register (the three-phase waveform mode).

Timer A3 mode register (Address 59_{16})

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0		0	0

Bit	Bit name	Function	At reset	R/W
0	Fix these bits to "00_2" in the three-phase waveform mode.		0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output ($TA3_{OUT}$ pin functions as a programmable I/O port pin.) 1 : Pulse output ($TA3_{OUT}$ pin functions as a pulse outpt pin.)	0	RW
3	Fix these bits to "000_2" in the three-phase waveform mode.		0	RW
4			0	RW
5			0	RW
6	Count source select bits	See Table 7.2.3.	0	RW
7			0	RW

Fig. 10.2.12 Structure of timer A3 mode register (three-phase waveform mode)

10.2.9 Output polarity set toggle flip-flop

The output polarity set toggle flip-flops 0 through 2 are used to control the output polarity of the positive and negative phases of the three-phase waveform.
In three-phase mode 0, values are set into the U-, V-, W-phase output polarity set buffer (bits 5 and 4 at address $A9_{16}$ and bit 3 at address $A8_{16}$).
In three-phase mode 1, a value is set into the three-phase output polarity set buffer (bit 3 at address $A6_{16}$).
These bits are transferred to the output polarity set toggle flip-flop at an underflow of timer A3.
The contents of the output polarity set toggle flip-flop are reversed at the end of the timer A0/A1/A2 one-shot pulse.
Table 10.2.2 lists the relationship between the contents of the output polarity set toggle flip-flop and the output level, and Figure 10.2.13 shows the operations of the output polarity set buffer and output polarity set toggle flip-flop.

Table 10.2.2 Relationship between contents of output polarity set toggle flip-flop and output level

Contents of output polarity set toggle flip-flop	Output level of positive phase	Output level of negative phase
0	H	L
1	L	H

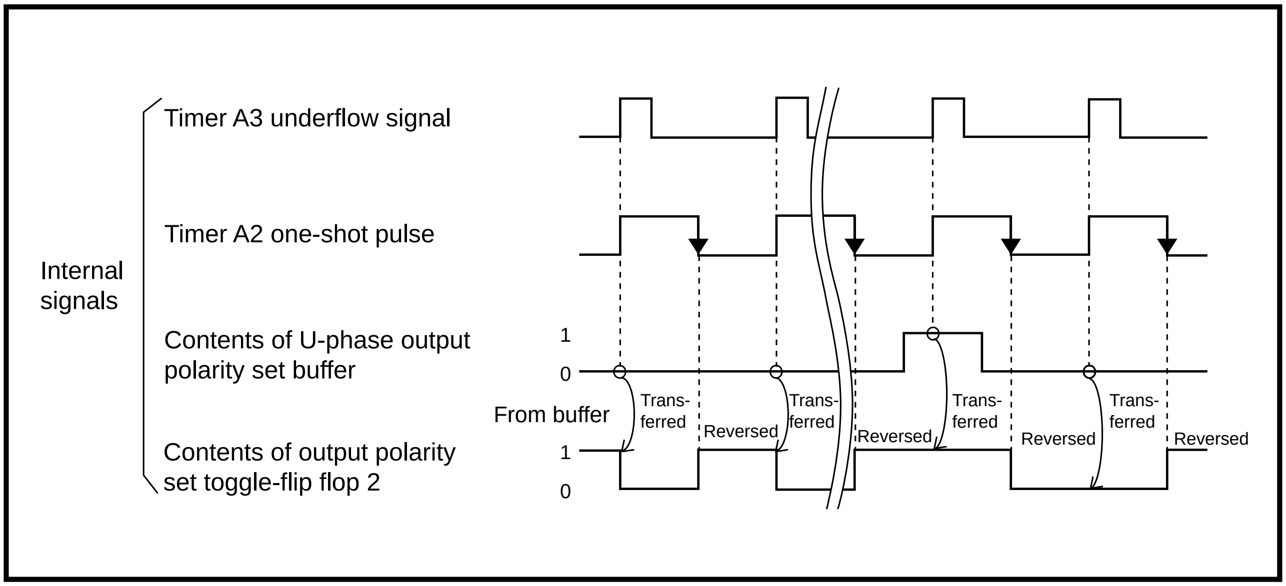


Fig. 10.2.13 Operations of output polarity set buffer and output polarity set toggle flip-flop

10.2.10 Three-phase waveform mode I/O pins

When the three-phase waveform mode is selected, port $P6_0$ through $P6_5$ pins become the three-phase waveform output pins, pin $\overline{P6OUT_{CUT}}$ becomes the three-phase-waveform-output-forcibly-cutoff signal input pin. Figure 10.2.14 shows the pins used in the three-phase waveform mode.

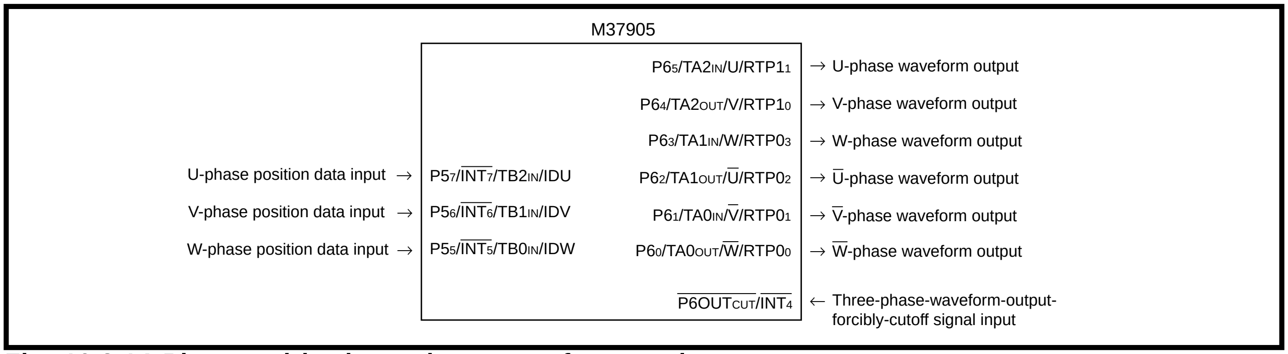


Fig. 10.2.14 Pins used in three-phase waveform mode

10.2.11 Pin $\overline{P6OUT_{CUT}}$ (three-phase-waveform-output-forcibly-cutoff signal input pin)

When a falling edge is input to pin $\overline{P6OUT_{CUT}}$, the waveform output control bit (bit 7 at address $A6_{16}$) becomes "0"; and then the three-phase waveform output pins enter the floating state. (In other words, the three-phase waveform output becomes inactive.)

When restarting the three-phase waveform output after this output becomes inactive, be sure to return the input level at pin $\overline{P6OUT_{CUT}}$ to "H"; and then, be sure to set the waveform output control bit to "1." When the input level at pin $\overline{P6OUT_{CUT}}$ is "L," the waveform output control bit cannot be "1."

Also, at this time, bits 0 through 7 of the port P6 direction register (address 10_{16}) become "000000_2." (Refer to section **"5.2.4 Pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$."**) Therefore, if it is necessary to switch port pins $P6_0$ through $P6_5$ to the port output pins, be sure to do as follows:

① Return the input level at pin $\overline{P6OUT_{CUT}}$ to "H" level.

② Write data to the port P6 register (address E_{16})'s bits, corresponding to the port P6 pins which will output data.

③ Set the port P6 direction register's bits, corresponding to the port P6 pins in ②, to "1" in order to set these port pins to the output mode.

When the input level at pin $\overline{P6OUT_{CUT}}$ is "L," each bit of the port P6 direction register cannot be "1."

Figure 10.2.15 shows the relationship between the $\overline{P6OUT_{CUT}}$ input, waveform output control bit, and three-phase waveform output pin.

Note that, when not inactivating the three-phase waveform output by using pin $\overline{P6OUT_{CUT}}$, be sure to connect pin $\overline{P6OUT_{CUT}}$ to Vcc via a resistor.

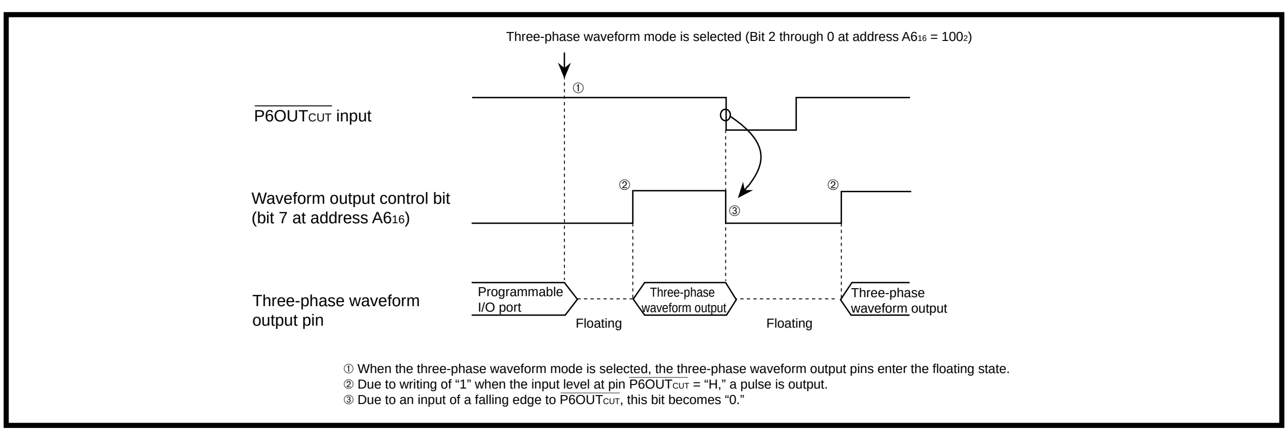


Fig. 10.2.15 Relationship between $\overline{P6OUT_{CUT}}$ input, waveform output control bit, and three-phase waveform output pin

10.3 Three-phase mode 0

10.3.1 Setting for three-phase mode 0

Explanation of the triangular wave modulation output and saw-tooth-wave modulation output in three-phase mode 0 is described below. Table 10.3.1 lists the differences between the triangular wave modulation output and the saw-tooth-wave modulation output (in view of software).

Table 10.3.1 Differences between triangular wave modulation output and saw-tooth-wave modulation output (in view of software)

	Triangular wave modulation output	Saw-tooth-wave modulation output
Trigger of dead-time timer	Falling edge of timers A0 through A2	Falling and Rising edges of timers A0 through A2
Contents of output polarity set buffer	Reversed at each timer A3 interrupt request occurrence.	Not reversed.

Figures 10.3.1 and 10.3.2 show an initial setting example for registers relevant to three-phase mode 0, Figure 10.3.3 shows a data-updating example in three-phase mode 0.
Note that the initial output level at the three-phase waveform output pin is undefined. Be sure to start the three-phase waveform output (in other words, the waveform output is enabled.) after the output level at the three-phase waveform output pin is stabilized.

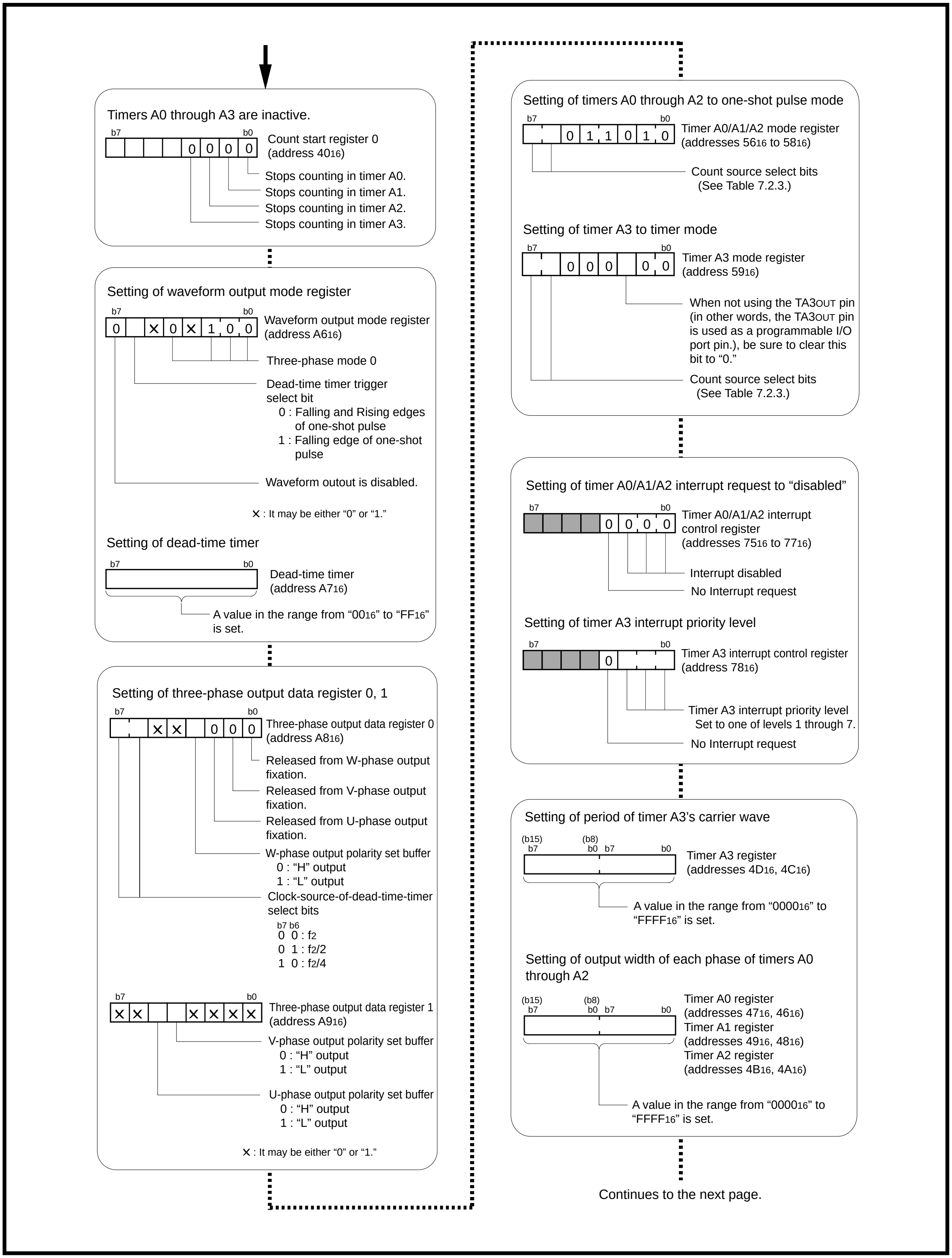


Fig. 10.3.1 Initial setting example for registers relevant to three-phase mode 0 (1)

10.3 Three-phase mode 0

Continued from the preceding page.

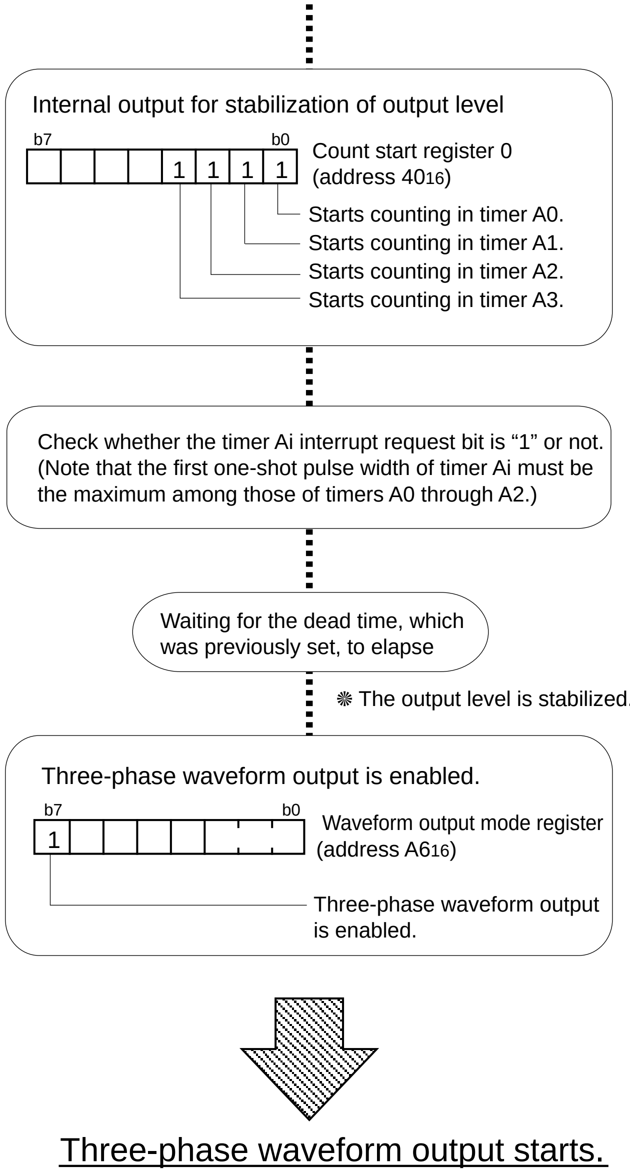


Fig. 10.3.2 Initial setting example for registers relevant to three-phase mode 0 (2)

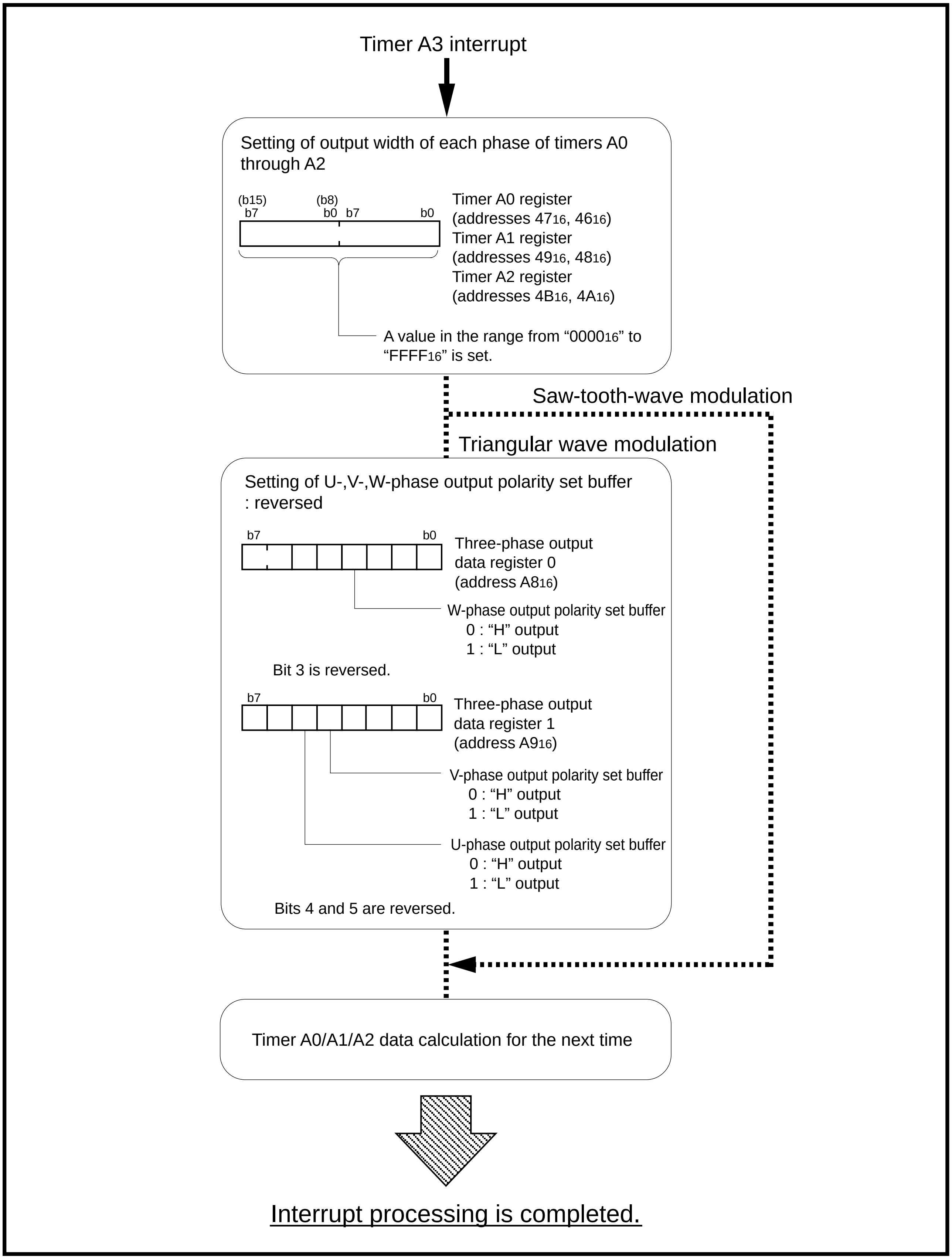


Fig. 10.3.3 Data-updating example in three-phase mode 0

10.3.2 Operation in three-phase wave mode 0

Figure 10.3.4 shows a triangular wave modulation output example (three-phase mode 0), and Figure 10.3.5 shows a saw-tooth-wave modulation output example (three-phase mode 0)

① When an underflow occurs in the timer A3 counter, a timer A3 interrupt request is generated; simultaneously, the one-shot pulse outputs of timer A0 through A2 are started. Also, the contents of the output polarity set buffer of each phase are transferred to the output polarity set toggle flip-flop. In the case of the saw-tooth-wave modulation output, the one-shot pulse of the dead-time timer is output. Also, each of the positive and negative waveform outputs is not allowed to become “L” level from “H” level until the reversed signal of the one-shot pulse output of the dead-time timer rises.

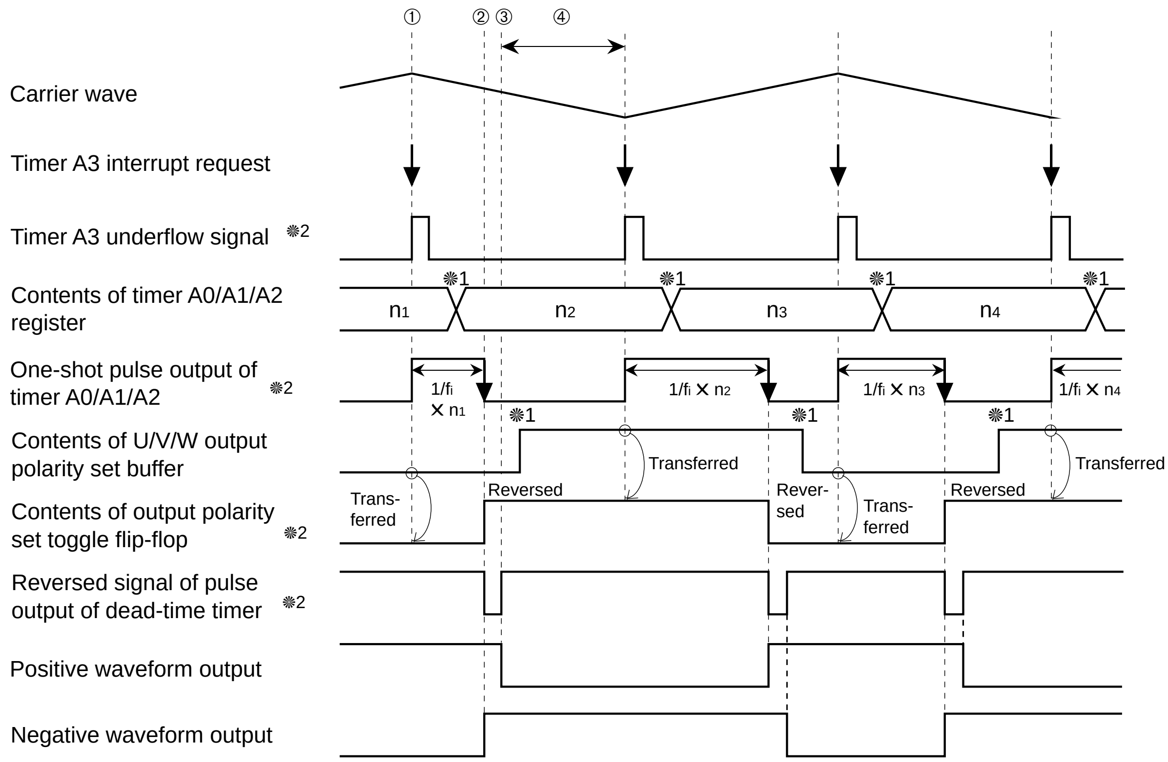
② The contents of the output polarity set toggle flip-flop are reversed at each falling edge of the one shot pulse output of timer A0/A1/A2. Simultaneously, the one-shot pulse of the dead-time timer is output.

③ Each of the positive and negative waveform outputs is not allowed to become “L” level from “H” level until the reversed signal of the one-shot pulse output of the dead-time timer rises.

④ In the case of the triangular wave modulation output, before an underflow occurs in the timer A3 counter again, be sure to write the next data to the output polarity set buffer of each phase.

Repeat procedures from ① through ④ for the three-phase waveform output control.

Figure 10.3.6 shows the triangular wave modulation output model (for one period), and Figure 10.3.7 shows the saw-tooth-wave modulation output model (for one period).



※1 : Written by software
※2 : This is an internal signal, which cannot be read from the external.

f_i : Count source of timer A0/A1/A2

Fig. 10.3.4 Triangular wave modulation output example (three-phase mode 0)

10.3 Three-phase mode 0

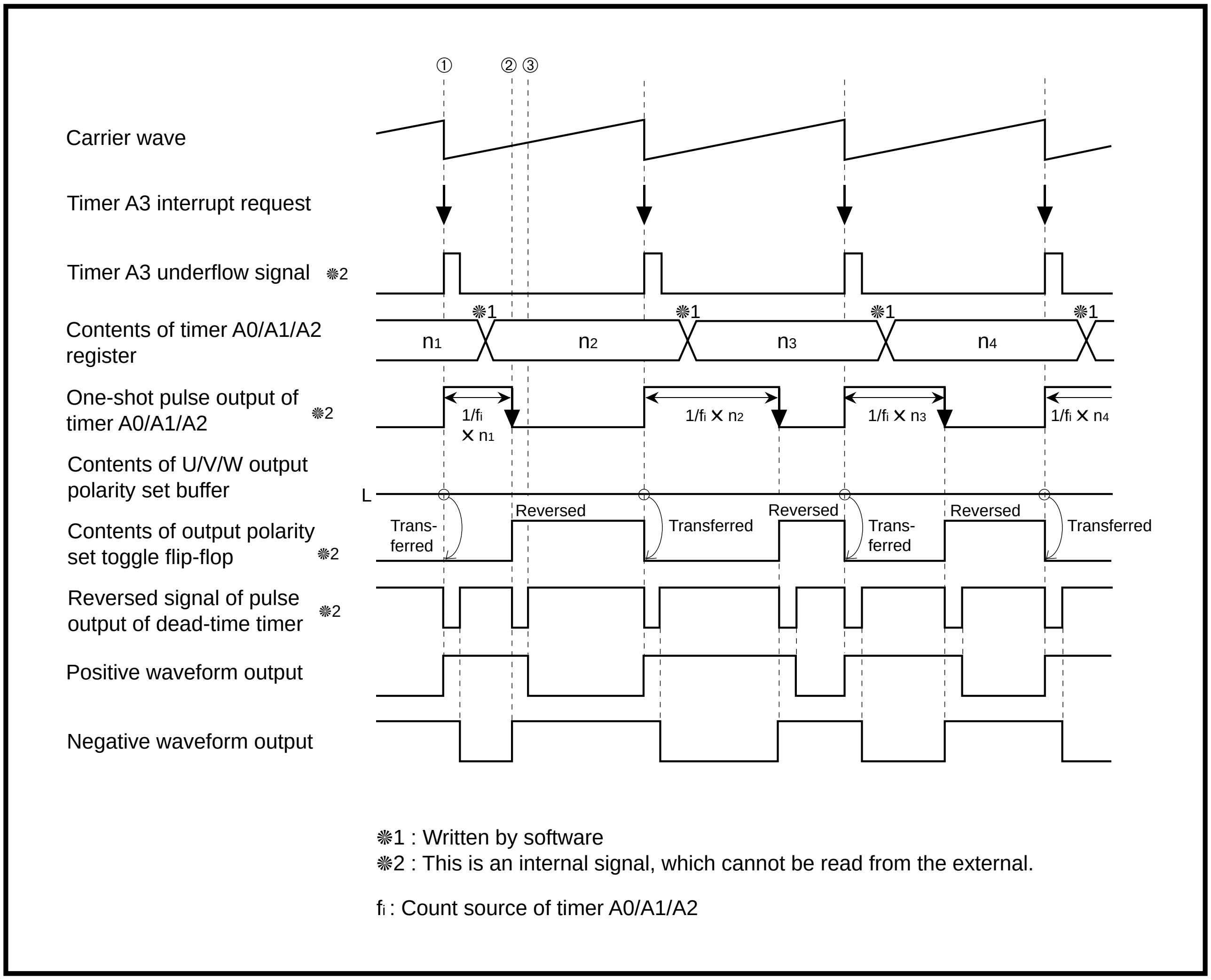
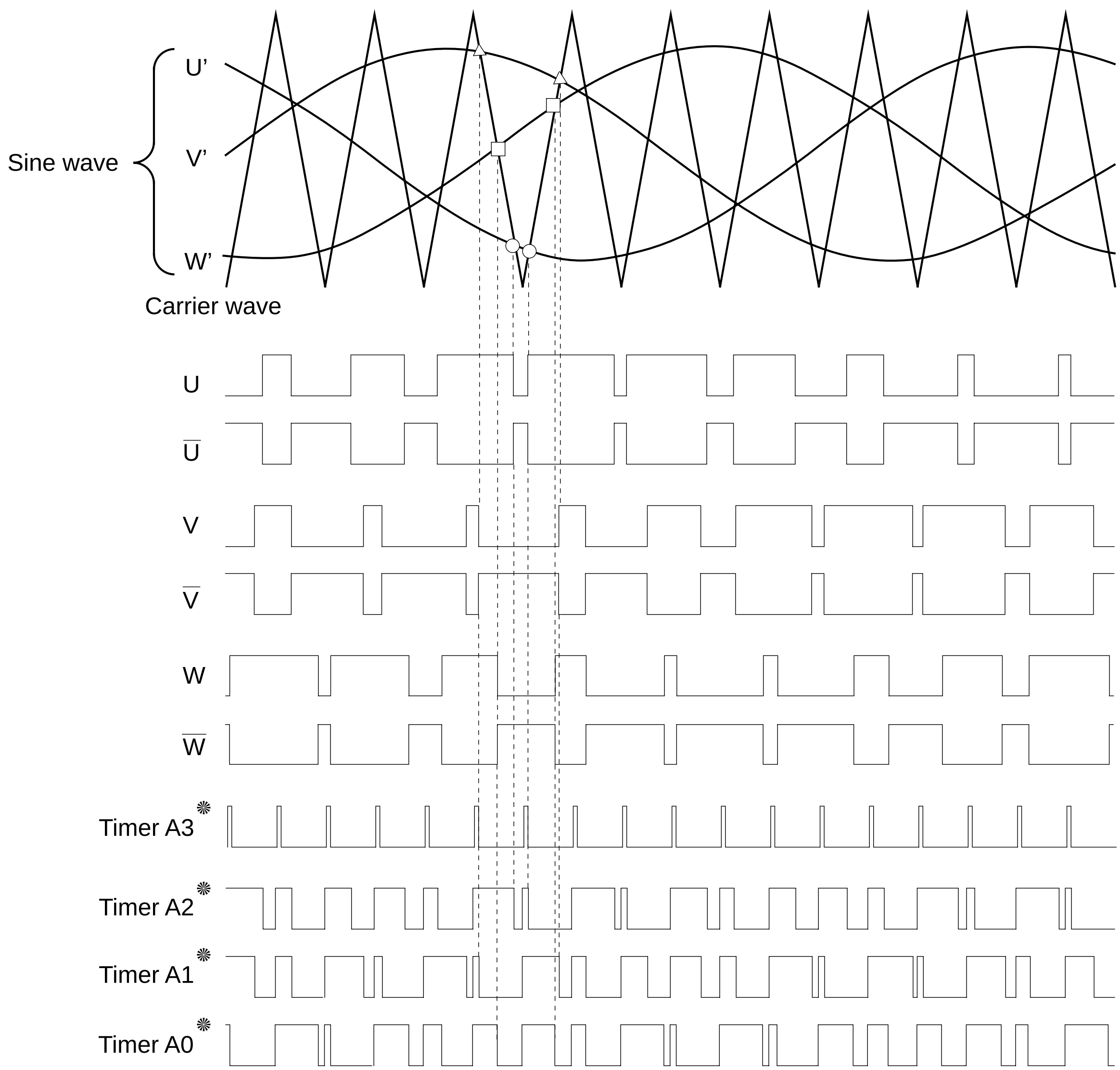


Fig. 10.3.5 Saw-tooth wave modulation output example (three-phase mode 0)



❋ This is an internal signal, which cannot be read from the external.

Note: The dead time is executed.

Fig. 10.3.6 Triangular wave modulation output model (for one period)

10.3 Three-phase mode 0

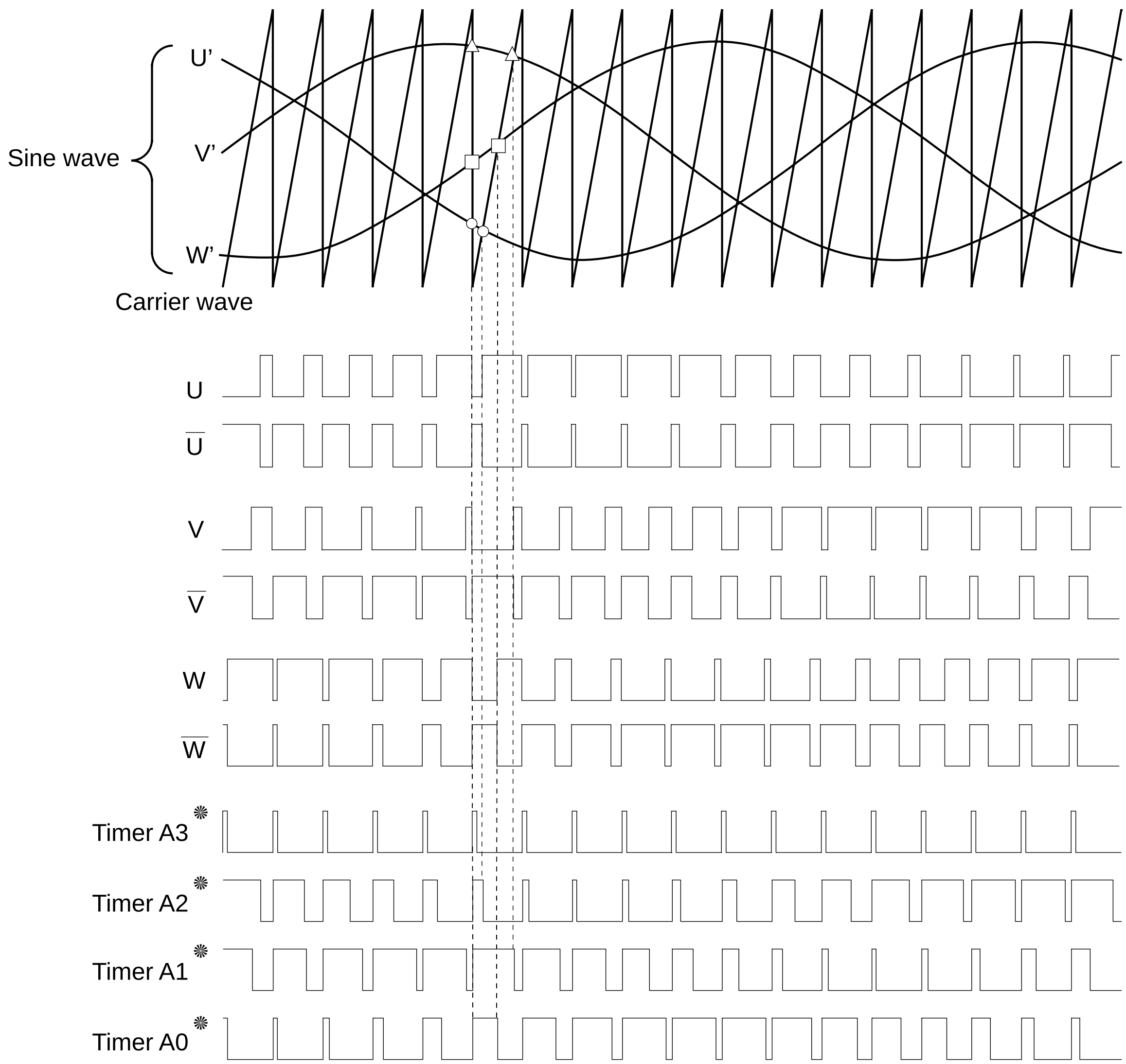


Fig. 10.3.7 Saw-tooth-wave modulation output model (for one period)

10.4 Three-phase mode 1

10.4.1 Setting for three-phase mode 1

In the triangular wave modulation, three-phase mode 1 is more efficiently controllable than three-phase mode 0. Therefore, three-phase mode 1 can mitigates the software’s load.

Figure 10.4.1 and Figure 10.4.2 show an initial setting example of registers relevant to three-phase mode 1, and Figure 10.4.3 shows a data-updating example in three-phase mode 1.

Note that the initial output level at the three-phase waveform output pin is undefined. Be sure to start the three-phase waveform output (in other words, the waveform output is enabled.) after the output level at the three-phase waveform output pin is stabilized.

10.4 Three-phase mode 1

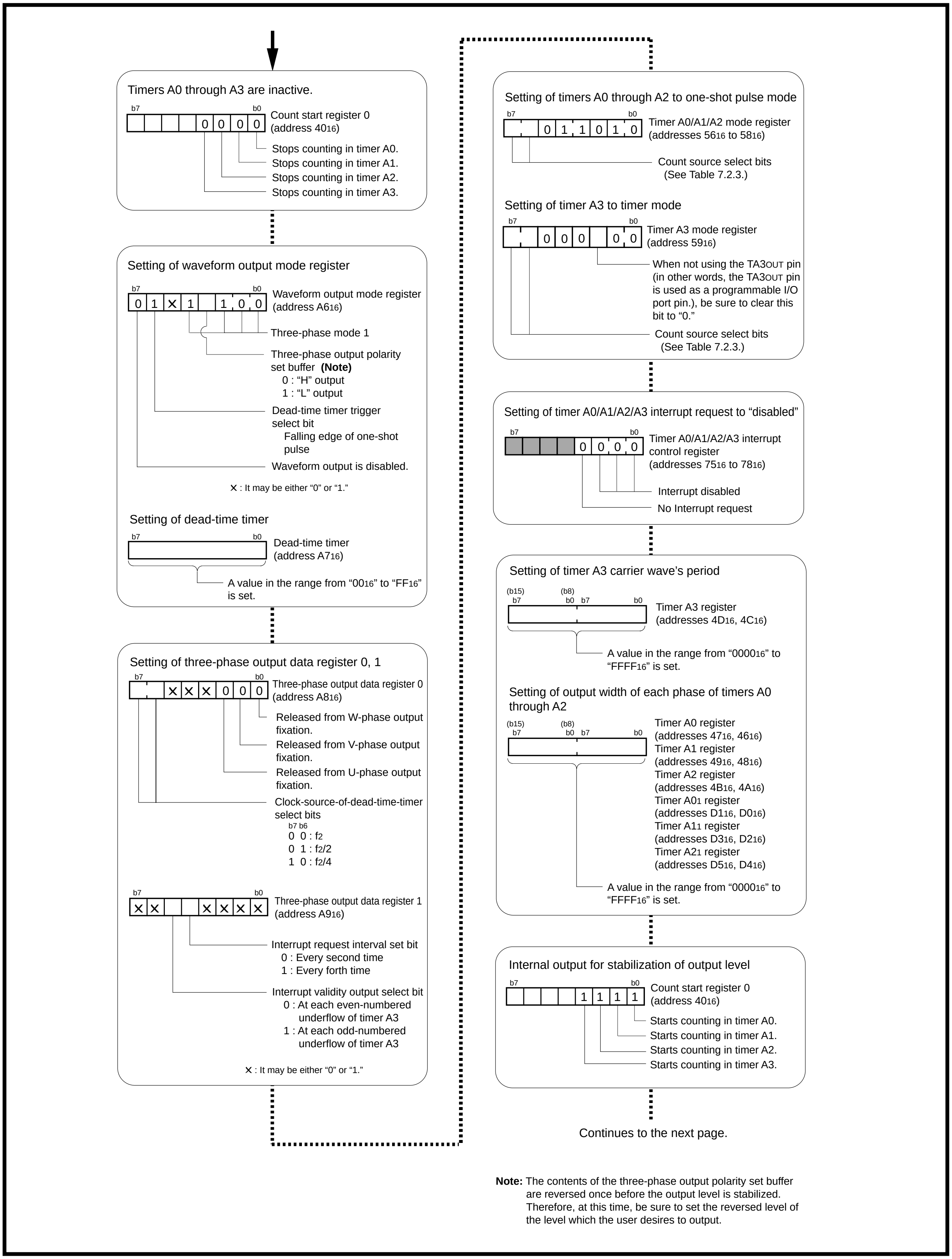


Fig. 10.4.1 Initial setting example for registers relevant to three-phase mode 1 (1)

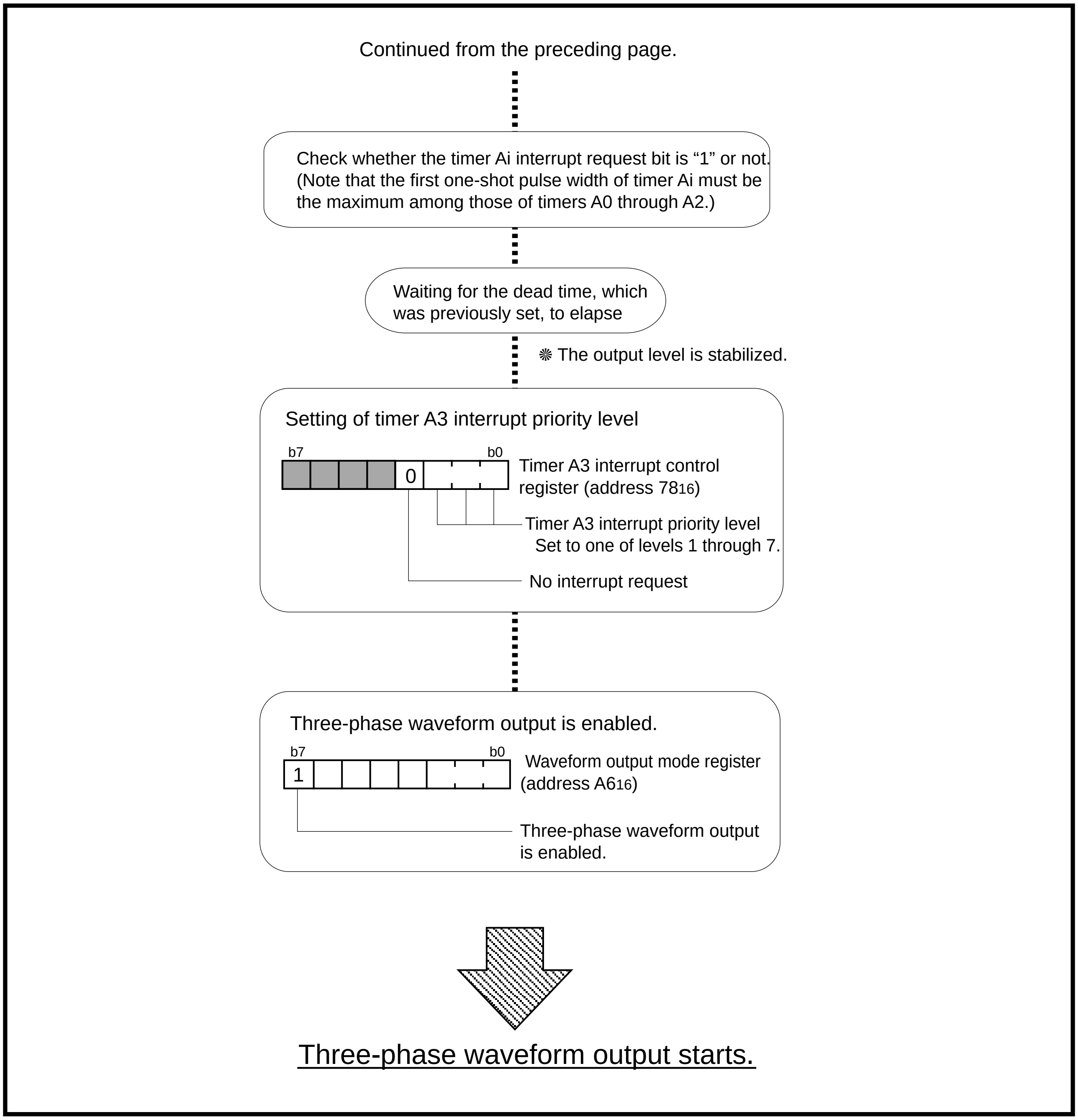


Fig. 10.4.2 Initial setting example for registers relevant to three-phase mode 1 (2)

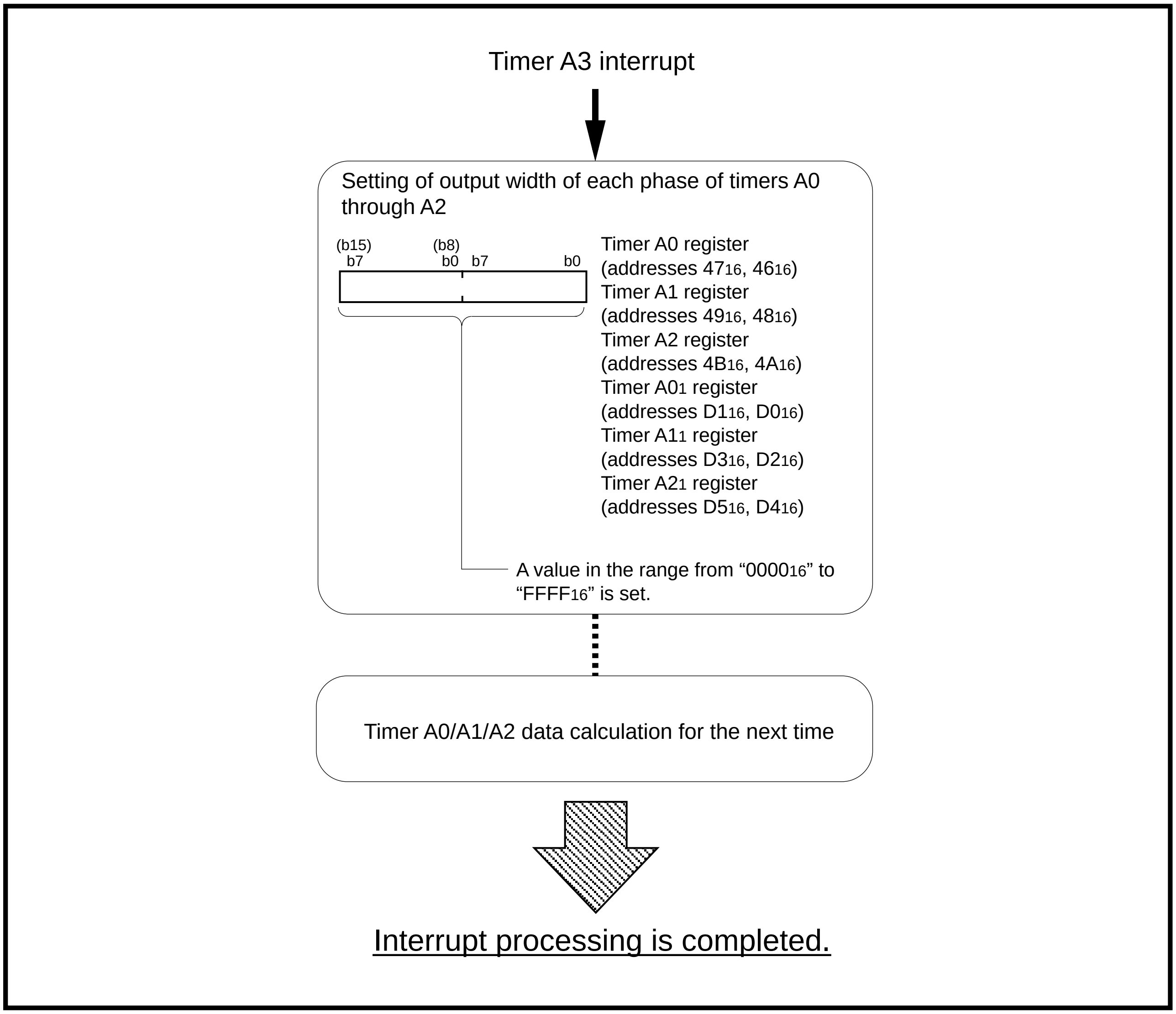


Fig. 10.4.3 Data-updating example in three-phase mode 1

10.4.2 Operation in three-phase mode 1

Figure 10.4.4 shows a triangular wave modulation output example (three-phase mode 1).

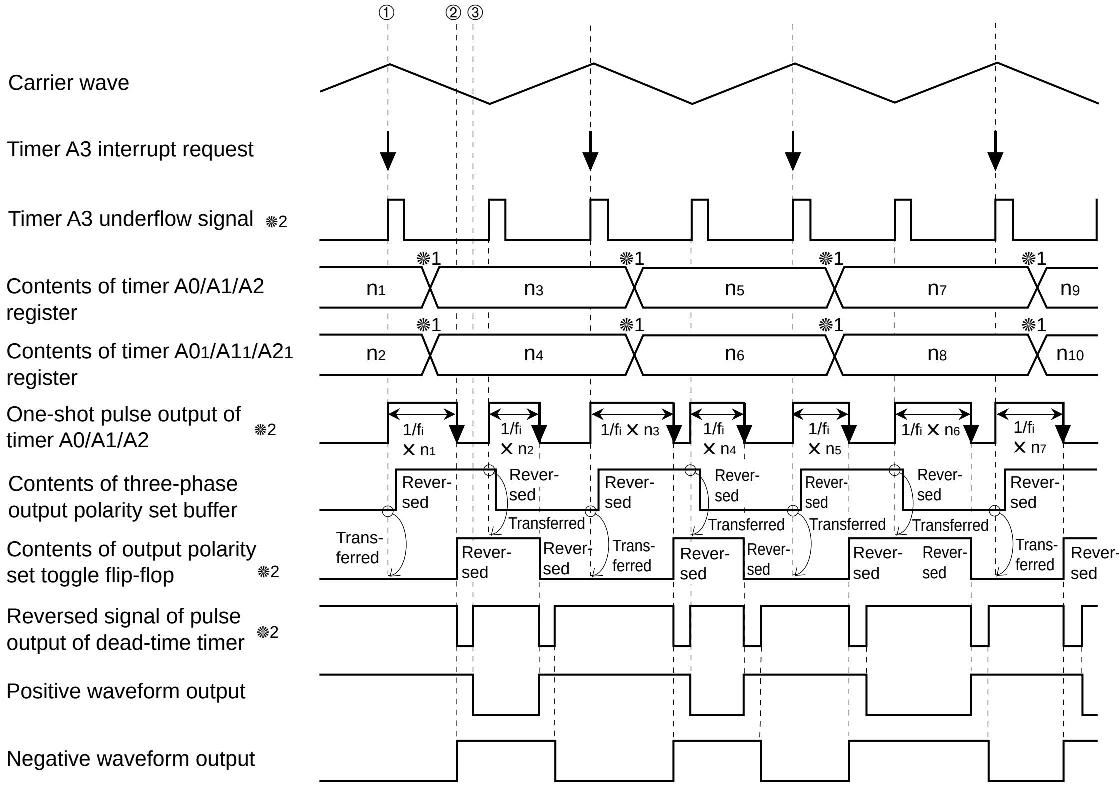
① When an underflow occurs in the timer A3 counter, a timer A3 interrupt request is generated; simultaneously, the one-shot pulse outputs of timers A0 through A2 are started. Also, the contents of the three-phase output polarity set buffer are transferred to the output polarity set toggle flip-flop, and then, the contents of the three-phase output polarity set buffer are reversed.
② The contents of the output polarity set toggle flip-flop are reversed at each falling edge of the one-shot pulse output of timer A0/A1/A2. Simultaneously, the one-shot pulse of the dead-time timer is output.
③ Each of the positive and negative waveform outputs is not allowed to become "L" level from "H" level until the reversed signal of the one-shot pulse output of the dead-time timer rises.

Repeat procedures from ① through ③ for the three-phase waveform output control.

In the case of three-phase mode 1, the value of timer Ai (i = 0 through 2) and the value of timer Ai_1 are counted alternately. Immediately after the count start in timer Ai, however, the value of the timer Ai register is counted twice in succession. (It is a limitation to the case immediately after the count start in timer Ai.) At this time, the timer Ai's one-shot pulse becomes the same length twice in succession, also. Figure 10.4.5 shows an output example at start of three-phase mode 1.

For the triangular wave modulation output model (for one period), see Figure 10.3.6.

10.4 Three-phase mode 1

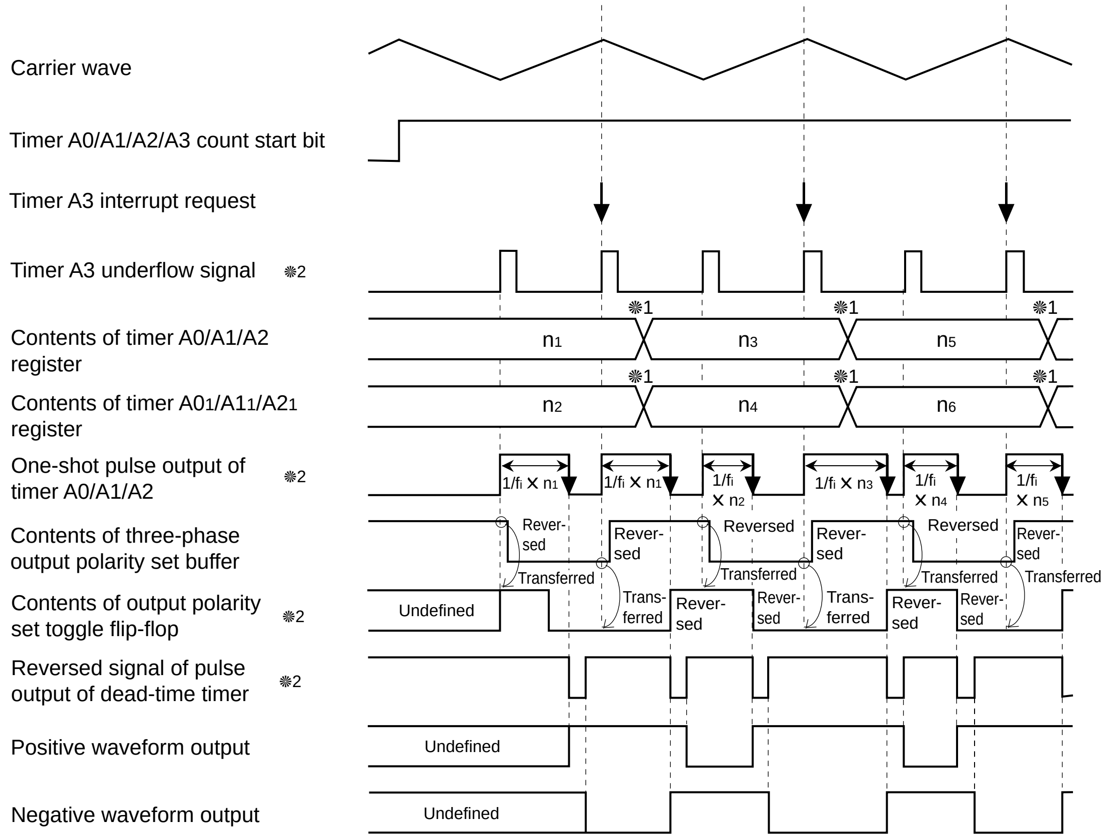


❋1 : Written by software
❋2 : This is an internal signal, which cannot be read from the external.

fi : Count source of timer A0/A1/A2

This applies under the following conditions:
• Timer A3 interrupt request: every second time
• Timer A3 interrupt validity output: even-numbered underflow in time A3

Fig. 10.4.4 Triangular wave modulation output example (three-phase mode 1)



※1 : Written by software
※2 : This is an internal signal, which cannot be read from the external.

fi : Count source of timer A0/A1/A2

This applies under the following conditions:
• Timer A3 interrupt request: every second time
• Timer A3 interrupt validity output: even-numbered underflow in time A3

Fig. 10.4.5 Output example at start of three-phase mode 1

10.5 Three-phase waveform output fixation

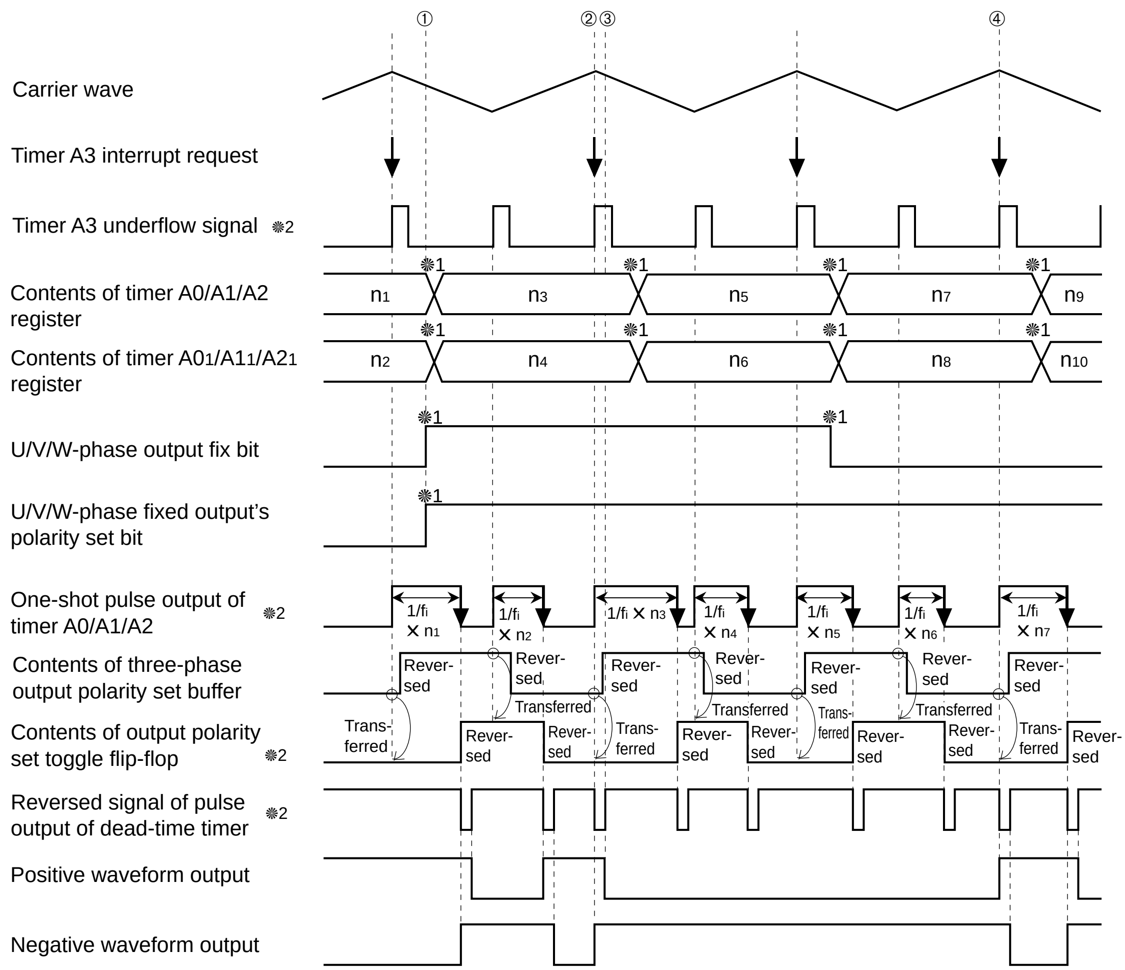
In the three-phase waveform output, by setting of the U/V/W-phase output fix bit (bits 2 through 0 at address $A8_{16}$) to "1," the output level of each phase can be fixed. The output level to be fixed (positive phase) is set by the U/V/W-phase fixed output's polarity set bit (bits 2 through 0 at address $A9_{16}$); in the case of the negative phase, the output level is fixed to the reversed level.
The U/V/W-phase output fix bit serves synchronously with a timer A3 interrupt request.
While the fixed level is output, be sure not to change the value of the U/V/W-phase fixed output's polarity set bit (bits 2 through 0 at address $A9_{16}$).
Figure 10.5.1 shows a triangular wave modulation output example using the U/V/W-phase output fix bit (three-phase mode 1).

① By software, set the following bits:
- the U/V/W-phase output fix bit (bits 2 through 0 at address $A8_{16}$)
- the U/V/W-phase fixed output's polarity set bit (bits 2 through 0 at address $A9_{16}$)

② The contents of the above bits become valid synchronously with the next timer A3 interrupt request, and then, the output level of the positive waveform is fixed to the level which was set by the U/V/W-phase fixed output's polarity set bit. In the case of the negative phase, the output level is fixed to the reversed level.

③ Each of the positive and negative waveform outputs is not allowed to become "L" level from "H" level until the reversed signal of the one-shot pulse output of the dead-time timer rises.

④ The output fixation is also terminated synchronous with a timer A3 interrupt request.



❈1 : Written by software
❈2 : This is an internal signal, which cannot be read from the external.

fi : Count source of timer A0/A1/A2

This applies under the following conditions:
• Timer A3 interrupt request: every second time
• Timer A3 interrupt validity output: every even-numbered underflow in time A3

Fig. 10.5.1 Triangular wave modulation output example using U/V/W-phase output fix bit (three-phase mode 1)

10.6 Position-data-retain function

This function is used to retain the position data synchronously with the three-phase waveform output; and there are three position-data input pins for the U, V, and W phases.

A trigger to retain the position data (hereafter, this trigger is referred to as "retain trigger.") can be selected by the retain-trigger polarity select bit (bit 3 at address AA_{16}); this bits selects the falling edge of each positive phase or rising edge of one.

10.6.1 Operation of position-data-retain function

Figure 10.6.1 shows a usage example of the position-data-retain function (U phase) when a retain trigger is the falling edge of the positive signal.

① At the falling edge of the U-phase waveform output, the state at pin IDU is transferred to the U-phase position data retain bit (bit 2 at address AA_{16}).

② Until the next falling edge of the U-phase waveform output, the above value is retained.

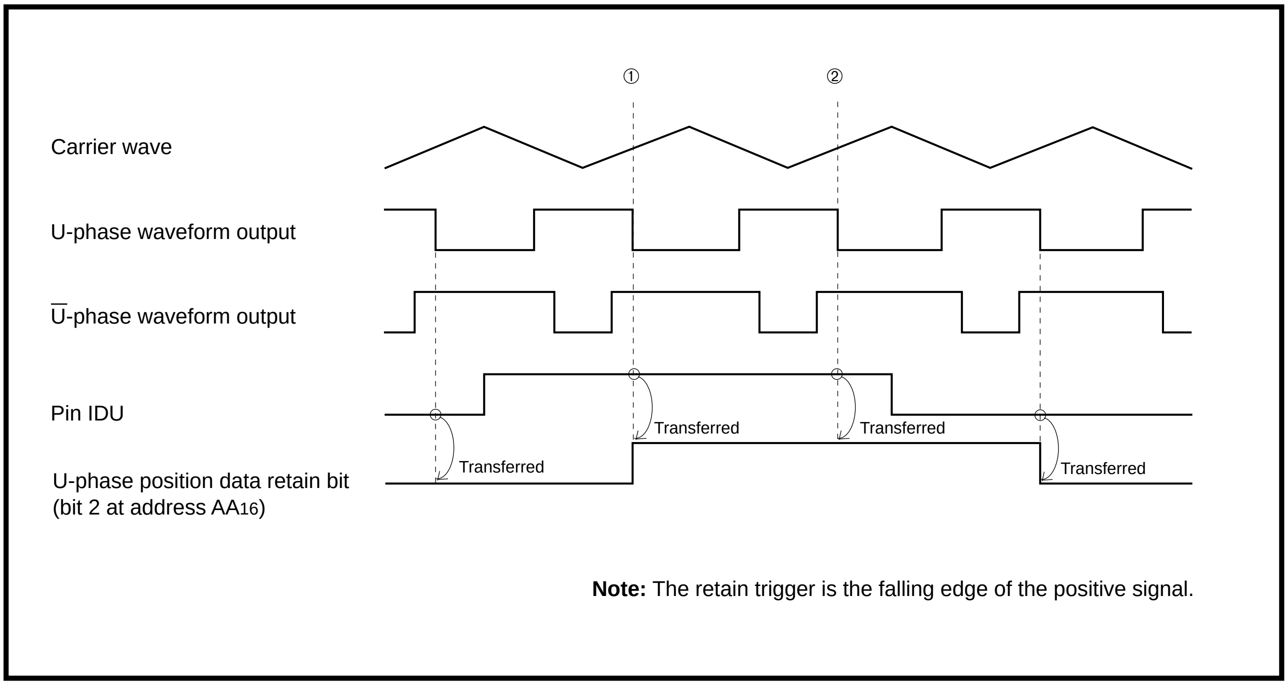


Fig. 10.6.1 Usage example of position-data-retain function (U phase)

[Precautions for three-phase waveform mode]

1. When using the three-phase waveform mode, be sure to fix the waveform output select bits (bits 2 to 0 at address $A6_{16}$) to "100_2," and then, set the relevant registers.
   When not using pulse output port 0 and three-phase waveform mode, be sure to fix the waveform output select bits (bits 2 through 0 at address $A6_{16}$) to "000_2."

2. When not inactivating the three-phase waveform output by using a falling edge input to pin $\overline{P6OUT_{CUT}}$, be sure to connect pin $\overline{P6OUT_{CUT}}$ to Vcc via a resistor.

3. While the fixed level is output, be sure not to change the value of the U/V/W-phase fixed output's polarity set bit (bits 2 through 0 at address $A9_{16}$).

MEMORANDUM

CHAPTER 11
SERIAL I/O

11.1 Overview

Serial I/O consists of 3 channels: UART0, UART1 and UART2. They each have a transfer clock generating timer for the exclusive use of them and can operate independently.
UARTi (i = 0 to 2) has the following 2 operating modes:

(1) Clock synchronous serial I/O mode
Transmitter and receiver use the same clock as the transfer clock. Transfer data has a length of 8 bits.

(2) Clock asynchronous serial I/O (UART) mode
Transfer rate and transfer data format can arbitrarily be set. The user can select one transfer data length from the following: 7 bits, 8 bits, and 9 bits.

Figure 11.1.1 shows the transfer data formats in each operating mode.

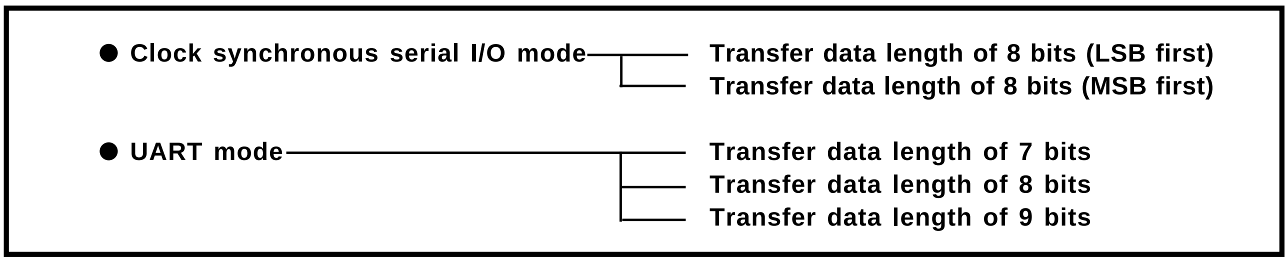


Fig. 11.1.1 Transfer data formats in each operating mode

11.2 Block description

Figure 11.2.1 shows the block diagram of serial I/O. Registers relevant to serial I/O are described below.

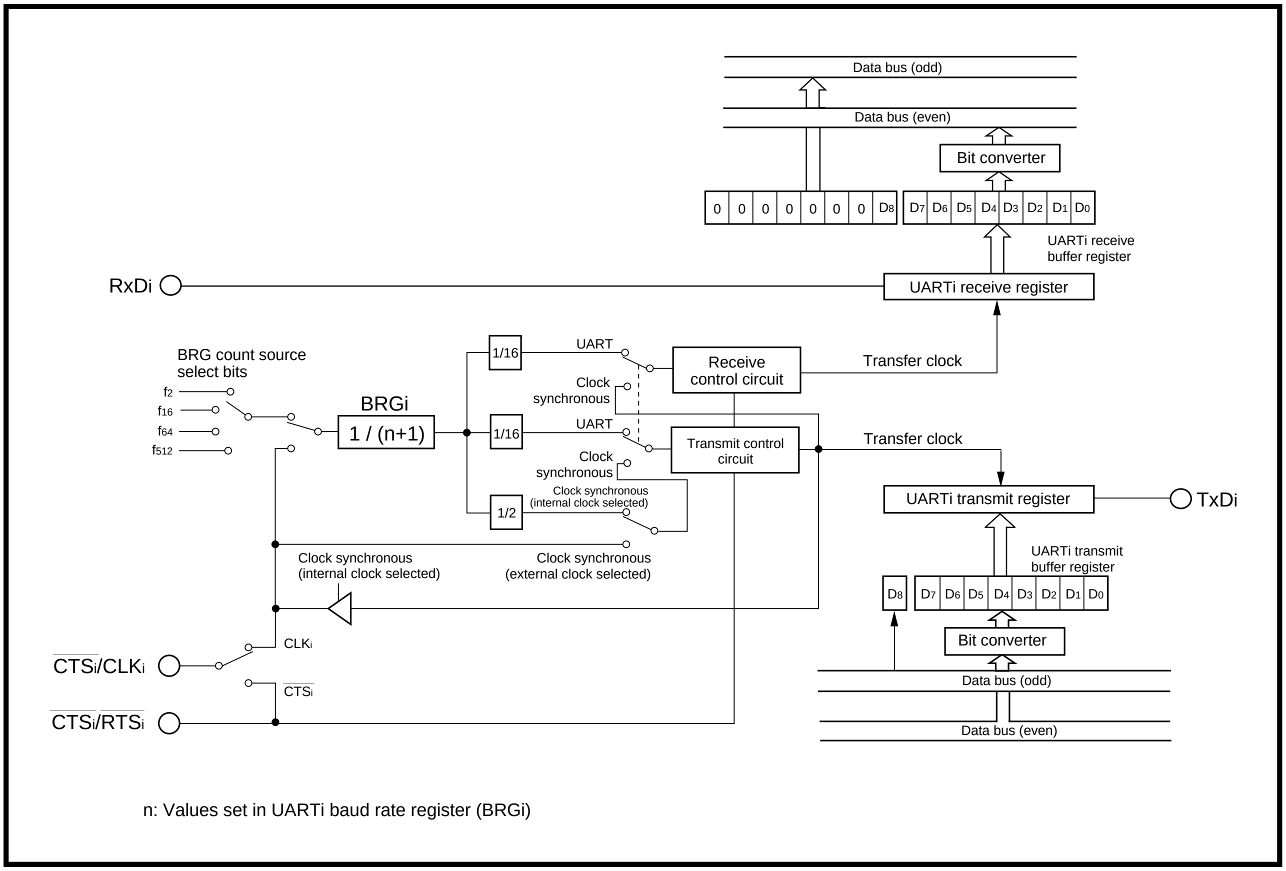


Fig. 11.2.1 Block diagram of serial I/O

11.2 Block description

11.2.1 UARTi transmit/receive mode register

Figure 11.2.2 shows the structure of UARTi transmit/receive mode register.

UART0 transmit/receive mode register (Address 30_{16})
UART1 transmit/receive mode register (Address 38_{16})
UART2 transmit/receive mode register (Address $B0_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	Serial I/O mode select bits	b2 b1 b0 0 0 0 : Serial I/O is invalid. (P1 and P8 function as programmable I/O ports.) 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : } Do not select. 0 1 1 : } 1 0 0 : UART mode (Transfer data length = 7 bits) 1 0 1 : UART mode (Transfer data length = 8 bits) 1 1 0 : UART mode (Transfer data length = 9 bits) 1 1 1 : Do not select.	0	RW
1			0	RW
2			0	RW
3	Internal/External clock select bit	0 : Internal clock 1 : External clock	0	RW
4	Stop bit length select bit (Valid in UART mode) **(Note)**	0 : One stop bit 1 : Two stop bits	0	RW
5	Odd/Even parity select bit (Valid in UART mode when parity enable bit = “1.”) **(Note)**	0 : Odd parity 1 : Even parity	0	RW
6	Parity enable bit (Valid in UART mode) **(Note)**	0 : Parity disabled 1 : Parity enabled	0	RW
7	Sleep select bit (Valid in UART mode) **(Note)**	0 : Sleep mode terminated (Invalid) 1 : Sleep mode selected	0	RW

Note: Bits 4 to 6 are invalid in the clock synchronous serial I/O mode. (They may be either “0” or “1.”) Additionally, fix bit 7 to “0.”

Fig. 11.2.2 Structure of UARTi transmit/receive mode register

(1) Serial I/O mode select bits (bits 0 to 2)
These bits select a UARTi's operating mode.

(2) Internal/External clock select bit (bit 3)

■ **Clock synchronous serial I/O mode**
By clearing this bit to "0" in order to select an internal clock, the clock which is selected with the BRG count source select bits (bits 0 and 1 at addresses 34_{16}, $3C_{16}$ and $B4_{16}$) becomes the count source of the BRGi. (Refer to section **"11.2.6 UARTi baud rate register (BRGi)."**) The BRGi's output divided by 2 becomes the transfer clock. Additionally, the transfer clock is output from the CLK_i pin.
By setting this bit to "1" in order to select an external clock, the clock input to the CLK_i pin becomes the transfer clock.

■ **UART mode**
By clearing this bit to "0" in order to select an internal clock, the clock which is selected with the BRG count source select bits (bits 0 and 1 at addresses 34_{16}, $3C_{16}$ and $B4_{16}$) becomes the count source of the BRGi. (Refer to section **"11.2.6 UARTi baud rate register (BRGi)."**) Then, the CLK_i pin functions as a programmable I/O port pin.
By setting this bit to "1" in order to select an external clock, the clock input to the CLK_i pin becomes the count source of BRGi.
Always in the UART mode, the BRGi's output divided by 16 becomes the transfer clock.

(3) Stop bit length select bit, Odd/Even parity select bit, Parity enable bit (bits 4 to 6)
Refer to section **"11.4.2 Transfer data format."**

(4) Sleep select bit (bit 7)
Refer to section **"11.4.8 Sleep mode."**

11.2.2 UARTi transmit/receive control register 0

Figure 11.2.3 shows the structure of UARTi transmit/receive control register 0.

UART0 transmit/receive control register (Address 34_{16})
UART1 transmit/receive control register (Address $3C_{16}$)
UART2 transmit/receive control register (Address $B4_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	BRG count source select bits	b1 b0 0 0 : Clock f_2 0 1 : Clock f_{16} 1 0 : Clock f_{64} 1 1 : Clock f_{512}	0	RW
1			0	RW
2	$\overline{CTS}$/$\overline{RTS}$ function select bit **(Note 1)**	0 : The $\overline{CTS}$ function is selected. 1 : The $\overline{RTS}$ function is selected.	0	RW
3	Transmit register empty flag	0 : Data is present in the transmit register. (Transmission is in progress.) 1 : No data is present in the transmit register. (Transmission is completed.)	1	RO
4	$\overline{CTS}$/$\overline{RTS}$ enable bit	0 : The $\overline{CTS}$/$\overline{RTS}$ function is enabled. 1 : The $\overline{CTS}$/$\overline{RTS}$ function is disabled.	0	RW
5	UARTi receive interrupt mode select bit	0 : Reception interrupt 1 : Reception error interrupt	0	RW
6	CLK polarity select bit (This bit is used in the clock synchronous serial I/O mode.) **(Note 2)**	0 : At the falling edge of the transfer clock, transmit data is output; at the rising edge of the transfer clock, receive data is input. When not in transferring, pin CLKi's level is "H." 1 : At the falling edge of the transfer clock, transmit data is output; at the falling edge of the transfer clock, receive data is input. When not in transferring, pin CLKi's level is "L."	0	RW
7	Transfer format select bit (This bit is used in the clock synchronous serial I/O mode.) **(Note 2)**	0 : LSB (Least Significant Bit) first 1 : MSB (Most Significant Bit) first	0	RW

Notes 1: Valid when the $\overline{CTS}$/$\overline{RTS}$ enable bit (bit 4) is "0" and $\overline{CTS_i}$/$\overline{RTS_i}$ separate select bit (bit 0, 1, or 4 at address AC_{16}) is "0."
2: Fix these bits to "0" in the UART mode or when serial I/O is disabled.

Fig. 11.2.3 Structure of UARTi transmit/receive control register 0

(1) BRG count source select bits (bits 0 and 1)
Refer to section **"11.2.1 (2) Internal/External clock select bit."**

(2) $\overline{CTS}$/$\overline{RTS}$ function select bit (bit 2)
Refer to section **"11.2.10 $\overline{CTS}$/$\overline{RTS}$ function."**

(3) Transmit register empty flag (bit 3)
This flag is cleared to "0" when the UARTi transmit buffer register's contents have been transferred to the UARTi transmit register. When transmission has been completed and the UARTi transmit register becomes empty, this flag is set to "1."

(4) $\overline{CTS}$/$\overline{RTS}$ enable bit (bit 4)
Refer to section **"11.2.10 $\overline{CTS}$/$\overline{RTS}$ function."**

(5) UARTi receive interrupt mode select bit (bit 5)
Refer to section **"11.2.7 (2) Interrupt request bit."**

(6) CLK polarity select bit (bit 6)
Refer to section **"11.3.1 (3) Polarity of transfer clock."**

(7) Transfer format select bit (bit 7)
Refer to section **"11.3.2 Transfer data format."**

11.2.3 UARTi transmit/receive control register 1

Figure 11.2.4 shows the structure of UARTi transmit/receive control register 1.

UART0 transmit/receive control register 1 (Address 35_{16})
UART1 transmit/receive control register 1 (Address $3D_{16}$)
UART2 transmit/receive control register 1 (Address $B5_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0	RW
1	Transmit buffer empty flag	0 : Data is present in the transmit buffer register 1 : No data is present in the transmit buffer register	1	RO
2	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0	RW
3	Receive complete flag	0 : No data is present in the receive buffer register 1 : Data is present in the receive buffer register	0	RO
4	Overrun error flag	0 : No overrun error 1 : Overrun error detected	0	RO
5	Framing error flag **(Note)** (Valid in UART mode)	0 : No framing error 1 : Framing error detected	0	RO
6	Parity error flag **(Note)** (Valid in UART mode)	0 : No parity error 1 : Parity error detected	0	RO
7	Error sum flag **(Note)** (Valid in UART mode)	0 : No error 1 : Error detected	0	RO

Note: Bits 5 to 7 are invalid in the clock synchronous serial I/O mode.

Fig. 11.2.4 Structure of UARTi transmit/receive control register 1

(1) Transmit enable bit (bit 0)
By setting this bit to "1," UARTi enters the transmission-enabled state. By clearing this bit to "0" during transmission, UARTi enters the transmission-disabled state after the transmission which was in progress at that time is completed.

(2) Transmit buffer empty flag (bit 1)
This flag is set to "1" when data set in the UARTi transmit buffer register has been transferred from the UARTi transmit buffer register to the UARTi transmit register. This flag is cleared to "0" when data has been set in the UARTi transmit buffer register.

(3) Receive enable bit (bit 2)
By setting this bit to "1," UARTi enters the reception-enabled state. By clearing this bit to "0" during reception, UARTi quits the reception immediately and enters the reception-disabled state.

(4) Receive complete flag (bit 3)
This flag is set to "1" when data has been ready in the UARTi receive register and that has been transferred to the UARTi receive buffer register (i.e., when reception is completed). This flag is cleared to "0" in one of the following cases:
- When the low-order byte of the UARTi receive buffer register has been read out
- When the receive enable bit (bit 2) has been cleared to "0"

(5) Overrun error flag (bit 4)
Refer to section **"11.3.7 Processing on detecting overrun error"** and **"11.4.7 Processing on detecting error."**

(6) Framing error flag, Parity error flag, Error sum flag (bits 5 to 7)
Refer to section **"11.4.7 Processing on detecting error."**

11.2.4 UARTi transmit register and UARTi transmit buffer register

Figure 11.2.5 shows the block diagram for the transmitter; Figure 11.2.6 shows the structure of UARTi transmit buffer register.

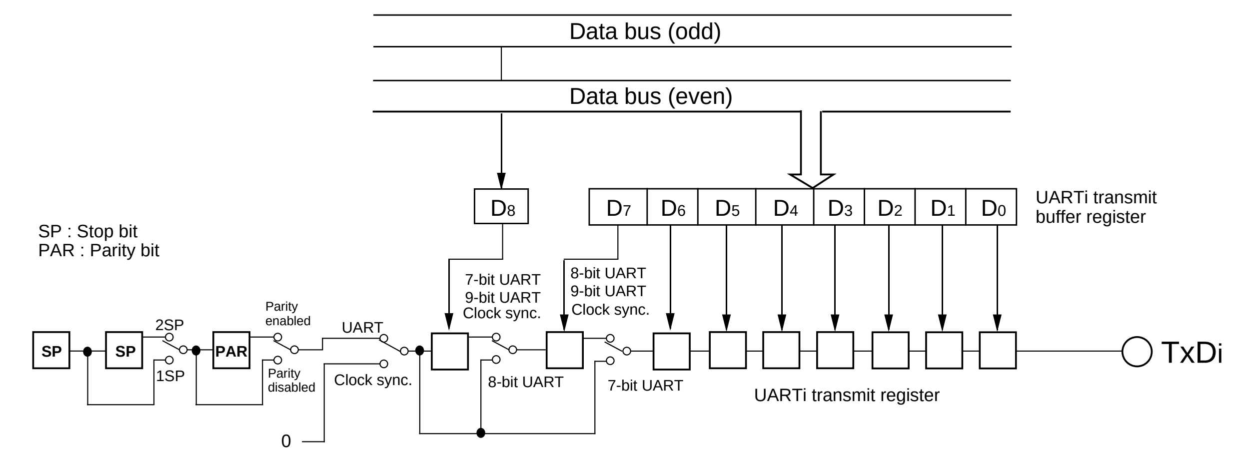


Fig. 11.2.5 Block diagram for transmitter

UART0 transmit buffer register (Addresses 33_{16}, 32_{16})
UART1 transmit buffer register (Addresses $3B_{16}$, $3A_{16}$)
UART2 transmit buffer register (Addresses $B3_{16}$, $B2_{16}$)

(b15) b7 … (b8) b0 | b7 … b0

Bit	Function	At reset	R/W
8 to 0	Transmit data is set.	Undefined	WO
15 to 9	Nothing is assigned.	Undefined	—

Note: Use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction for writing to this register.

Fig. 11.2.6 Structure of UARTi transmit buffer register

Transmit data is set into the UARTi transmit buffer register. Set the transmit data into the low-order byte of this register when the microcomputer operates in the clock synchronous serial I/O mode or when a 7-bit or 8-bit length of transfer data is selected in the UART mode. When a 9-bit length of transfer data is selected in the UART mode, set the transmit data into the UARTi transmit buffer register as follows:
•Bit 8 of the transmit data into bit 0 of high-order byte of this register.
•Bits 7 to 0 of the transmit data into the low-order byte of this register.

The transmit data which has been set in the UARTi transmit buffer register is transferred to the UARTi transmit register when the transmission conditions are satisfied, and then it is output from the TxDi pin synchronously with the transfer clock. The UARTi transmit buffer register becomes empty when the data set in the UARTi transmit buffer register has been transferred to the UARTi transmit register. Accordingly, the user can set the next transmit data.

When the “MSB first” is selected in the clock synchronous serial I/O mode, bit position of set data is reversed, and then the data of which bit position was reversed will be written, as a transmit data, into the UARTi transmit buffer register. (Refer to section **“11.3.2 Transfer data format.”**) Transmit operation itself is the same whichever format is selected, “LSB first” or “MSB first.”
When quitting the transmission which is in progress and setting the UARTi transmit buffer register again, follow the procedure described bellow:

① Clear the serial I/O mode select bits (bits 2 to 0 at addresses 30_{16}, 38_{16} and $B0_{16}$) to “000_2” (serial I/O disabled).
② Set the serial I/O mode select bits again.
③ Set the transmit enable bit (bit 0 at addresses 35_{16}, $3D_{16}$ and $B5_{16}$) to “1” (transmission enabled) and set transmit data in the UARTi transmit buffer register.

11.2.5 UARTi receive register and UARTi receive buffer register

Figure 11.2.7 shows the block diagram of the receiver; Figure 11.2.8 shows the structure of UARTi receive buffer register.

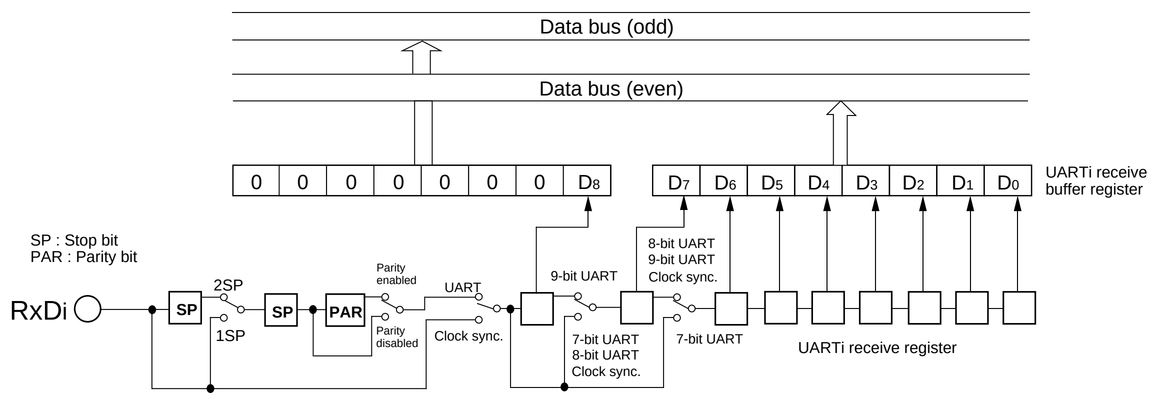
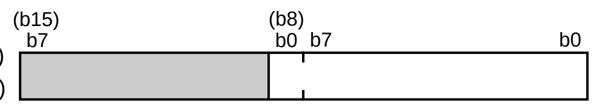


Fig. 11.2.7 Block diagram of receiver

UART0 receive buffer register (Addresses 37_{16}, 36_{16})
UART1 receive buffer register (Addresses $3F_{16}$, $3E_{16}$)
UART2 receive buffer register (Addresses $B7_{16}$, $B6_{16}$)



Bit	Function	At reset	R/W
8 to 0	Receive data is read out from here.	Undefined	RO
15 to 9	The value is “0” at reading.	0	—

Fig. 11.2.8 Structure of UARTi receive buffer register

The UARTi receive register is used to convert serial data, which is input to the RxD_i pin, into parallel data. This register takes in the signal input to the RxD_i pin, bit by bit, synchronously with the transfer clock.
The UARTi receive buffer register is used to read out receive data. When reception has been completed, the receive data taken in the UARTi receive register is automatically transferred to the UARTi receive buffer register. Note that the contents of the UARTi receive buffer register is updated when the next data has been ready in the UARTi receive register before the data transferred to the UARTi receive buffer register is read out. (i.e., an overrun error occurs.)
When “MSB first” is selected in the clock synchronous serial I/O mode, bit position of data in the UARTi receive buffer register is reversed, and then the data of which bit position was reversed will be read out as receive data. (Refer to section **“11.3.2 Transfer data format.”**) Receive operation itself is the same whichever format is selected, “LSB first” or “MSB first.”
The UARTi receive buffer register is initialized by setting the receive enable bit (bit 2 at addresses 35_{16}, $3D_{16}$ and $B5_{16}$) to “1” after clearing it to “0.”
Figure 11.2.9 shows the contents of the UARTi receive buffer register at reception completed.

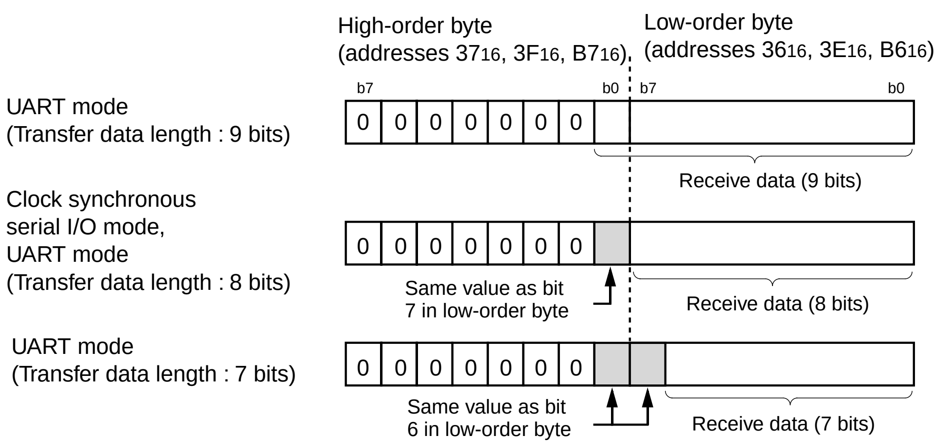


Fig. 11.2.9 Contents of UARTi receive buffer register at reception completed

11.2.6 UARTi baud rate register (BRGi)

The UARTi baud rate register (BRGi) is an 8-bit timer exclusively used for UARTi to generate a transfer clock. It has a reload register. Assuming that the value set in the BRGi is “n” (n = “00_{16}” to “FF_{16}”), the BRGi divides the count source frequency by (n + 1).

In the clock synchronous serial I/O mode, the BRGi is valid when an internal clock is selected, and the BRGi’s output divided by 2 becomes the transfer clock. In the UART mode, the BRGi is always valid, and the BRGi’s output divided by 16 becomes the transfer clock.

The data written to the BRGi is written to both the timer and the reload register whichever transmission/reception is in progress or not. Accordingly, writing to these register must be performed while transmission/reception halts.

Figure 11.2.10 shows the structure of the UARTi baud rate register (BRGi); Figure 11.2.11 shows the block diagram of transfer clock generating section.

UART0 baud rate register (BRG0) (Address 31_{16})
UART1 baud rate register (BRG1) (Address 39_{16})
UART2 baud rate register (BRG2) (Address $B1_{16}$)

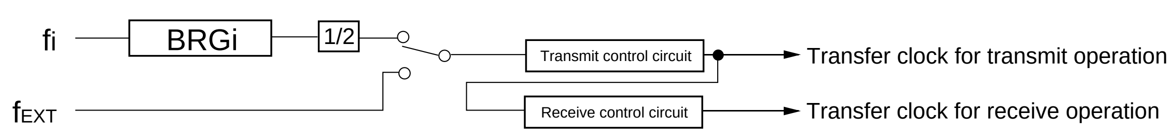
b7 b0

Bit	Function	At reset	R/W
7 to 0	Any value in the range from “00_{16}” to “FF_{16}” can be set. Assuming that the set value = n, BRGi divides the count source frequency by (n + 1).	Undefined	WO

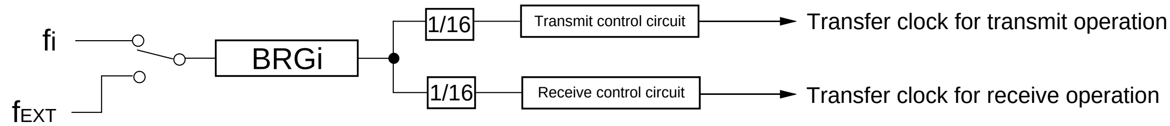
Note: Writing to this register must be performed while the transmission/reception halts.
Use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction for writing to this register.

Fig. 11.2.10 Structure of UARTi baud rate register (BRGi)

<Clock synchronous serial I/O mode>



<UART mode>



f_i : Clock selected by BRG count source select bits (f_2, f_{16}, f_{64}, or f_{512})
f_{EXT} : Clock input to CLK_i pin (external clock)

Fig. 11.2.11 Block diagram of transfer clock generating section

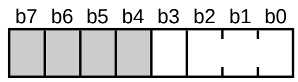
11.2.7 UARTi transmit interrupt control and UARTi receive interrupt control registers

When using UARTi, 2 types of interrupts (UARTi transmit and UARTi receive interrupts) can be used. Each interrupt has its corresponding interrupt control register. Figure 11.2.12 shows the structure of UARTi transmit interrupt control and UARTi receive interrupt control registers.

For details about these interrupts, refer to **"CHAPTER 6. INTERRUPTS."**

For the UARTi receive interrupt, a receive or receive error interrupt can be selected by the UARTi receive interrupt mode selected bit (bit 5 at addresses 34_{16}, $3C_{16}$ and $B4_{16}$).

UART0 transmit interrupt control register (Address 71_{16})
UART0 receive interrupt control register (Address 72_{16})
UART1 transmit interrupt control register (Address 73_{16})
UART1 receive interrupt control register (Address 74_{16})
UART2 transmit interrupt control register (Address $F1_{16}$)
UART2 receive interrupt control register (Address $F2_{16}$)



Bit	Bit name	Function	At reset	R/W
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt requested 1 : Interrupt requested	0	RW **(Note)**
7 to 4	Nothing is assigned.		Undefined	—

Note: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

Fig. 11.2.12 Structure of UARTi transmit interrupt control and UARTi receive interrupt control registers

(1) Interrupt priority level select bits (bits 0 to 2)

These bits select a priority level of the UARTi transmit interrupt or UARTi receive interrupt. When using UARTi transmit/receive interrupts, select one of the priority levels (1 to 7). When a UARTi transmit/receive interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL). The requested interrupt is enabled only when its priority level is higher than the IPL. (However, this applies when the interrupt disable flag (I) = "0.") To disable UARTi transmit/receive interrupts, be sure to set these bits to "000_2" (level 0).

(2) Interrupt request bit (bit 3)

The UARTi transmit interrupt request bit is set to "1" when data has been transferred from the UARTi transmit buffer register to the UARTi transmit register.
The UARTi receive interrupt request bit functions as below:

■ **When receive interrupt is selected (bit 5 = 0 at addresses 34_{16}, $3C_{16}$, $B4_{16}$)**

The UARTi receive interrupt request bit is set to "1" when data has been transferred from the UARTi receive register to the UARTi receive buffer register.
(However, the UARTi receive interrupt request bit does not change when an overrun error has occurred.)

■ **When receive error interrupt is selected (bit 5 = 1 at addresses 34_{16}, $3C_{16}$, $B4_{6}$)**

The UARTi receive interrupt request bit is set to "1" when an error (an overrun error in the clock synchronous serial I/O mode; an overrun error, framing error, or parity error in UART mode) has occurred.

Each interrupt request bit is automatically cleared to "0" when its corresponding interrupt request has been accepted. This bit can be set to "1" or cleared to "0" by software.

11.2.8 Serial I/O pin control register

Figure 11.2.13 shows the structure of the seral I/O pin control register.

Serial I/O pin control register (Address AC_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	$\overline{CTS_0}/\overline{RTS_0}$ separate select bit **(Note)**	0 : $\overline{CTS_0}/\overline{RTS_0}$ are used together. 1 : CTS_0/RTS_0 are separated.	0	RW
1	$\overline{CTS_1}/\overline{RTS_1}$ separate select bit **(Note)**	0 : $\overline{CTS_1}/\overline{RTS_1}$ are used together. 1 : CTS_1/RTS_1 are separated.	0	RW
2	$TxD_0/P1_3$ switch bit	0 : Functions as TxD_0. 1 : Functions as $P1_3$.	0	RW
3	$TxD_1/P1_7$ switch bit	0 : Functions as TxD_1. 1 : Functions as $P1_7$.	0	RW
4	$\overline{CTS_2}/\overline{RTS_2}$ separate select bit **(Note)**	0 : $\overline{CTS_2}/\overline{RTS_2}$ are used together. 1 : CTS_2/RTS_2 are separated.	0	RW
5	$TxD_2/P8_3$ switch bit	0 : Functions as TxD_2. 1 : Functions as $P8_3$.	0	RW
7, 6	The value is “00” at reading.		0	—

Note: Valid when the $\overline{CTS}/\overline{RTS}$ enable bit (bit 4 at addresses 34_{16}, $3C_{16}$, and $B4_{16}$) is “0.”

Fig. 11.2.13 Structure of serial I/O pin control register

(1) $\overline{CTS_0}/\overline{RTS_0}$ separate select bit (bit 0)

Refer to section **“11.2.10 $\overline{CTS}/\overline{RTS}$ function.”**

(2) $\overline{CTS_1}/\overline{RTS_1}$ separate select bit (bit 1)

Refer to section **“11.2.10 $\overline{CTS}/\overline{RTS}$ function.”**

(3) $TxD_0/P1_3$ switch bit (bit 2)

When this bit is set to “1,” the TxD_0 pin functions as a programmable I/O port pin ($P1_3$). When only reception is performed, the TxD_0 pin can be used as the $P1_3$ pin. When performing transmission, be sure to clear this bit to “0.”

(4) $TxD_1/P1_7$ switch bit (bit 3)

When this bit is set to “1,” the TxD_1 pin functions as a programmable I/O port pin ($P1_7$). When only reception is performed, the TxD_1 pin can be used as the $P1_7$ pin. When preforming transmission, be sure to clear this bit to “0.”

(5) $\overline{CTS_2}/\overline{RTS_2}$ separate select bit (bit 4)

Refer to section **“11.2.10 $\overline{CTS}/\overline{RTS}$ function.”**

(6) $TxD_2/P8_3$ switch bit (bit 5)

When this bit is set to “1,” the TxD_2 pin functions as a programmable I/O port pin ($P8_3$). When only reception is performed, the TxD_2 pin can be used as the $P8_3$ pin. When preforming transmission, be sure to clear this bit to “0.”

11.2.9 Port P1 direction register, Port P8 direction register

I/O pins for serial I/O are multiplexed with port P1 and P8 pins. When using pins $P1_1$, $P1_2$, $P1_5$, $P1_6$, $P8_1$, and $P8_2$ as serial I/O's input pins ($\overline{CTS_i}$, RxD_i), clear the corresponding bits of the port P1 and port P8 direction registers to "0" in order to set these pins for the input mode. When using these pins as other serial I/O's pins ($\overline{CTS_i}/\overline{RTS_i}$, CLK_i, TxD_i), these pins are forcibly set as I/O pins for serial I/O regardless of the port P1 and port P8 direction registers' contents. Figure 11.2.14 shows the relationship between the port P1 and port P8 direction registers and serial I/O's I/O pins. For details, refer to the description of each operating mode.

Port P1 direction register (Address 5_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin name	Function	At reset	R/W
0	Pin $\overline{CTS_0}/\overline{RTS_0}$	0 : Input mode 1 : Output mode When using pins $P1_1$, $P1_2$, $P1_5$, and $P1_6$ as serial I/O's input pins ($\overline{CTS_0}$, RxD_0, $\overline{CTS_1}$, RxD_1), clear the corresponding bits to "0."	0	RW
1	Pin $\overline{CTS_0}/CLK_0$		0	RW
2	Pin RxD_0		0	RW
3	Pin TxD_0		0	RW
4	Pin $\overline{CTS_1}/\overline{RTS_1}$		0	RW
5	Pin $\overline{CTS_1}/CLK_1$		0	RW
6	Pin RxD_1		0	RW
7	Pin TxD_1		0	RW

Port P8 direction register (Address 14_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Corresponding pin name	Function	At reset	R/W
0	Pin $\overline{CTS_2}/\overline{RTS_2}$ (Pin AN_8/DA_1) **(Note 1)**	0 : Input mode 1 : Output mode When using pins $P8_1$ and $P8_2$ as serial I/O's input pins ($\overline{CTS_2}$, RxD_2), clear the corresponding bits to "0."	0	RW
1	Pin $\overline{CTS_2}/CLK_2$ (Pin AN_9)		0	RW
2	Pin RxD_2 (Pin AN_{10})		0	RW
3	Pin TxD_2 (Pin AN_{11})		0	RW
7 to 4	Nothing is assigned.		Undefined	–

Notes 1: When using pin $\overline{CTS_2}/\overline{RTS_2}$, be sure that the D-$A_1$ output enable bit (bit 1 at address 96_{16}) = "0" (output disabled).
2: ( ) shows the I/O pins of other internal peripheral devices which are multiplexed.

Fig. 11.2.14 Relationship between port P1 and port P8 direction registers and serial I/O's I/O pins

11.2.10 $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function

When the $\overline{\text{CTS}}$ function is selected, the signal input to the $\overline{\text{CTS}_i}$ pin must be at "L" level. (This is one of the transmit conditions.)
When the $\overline{\text{RTS}}$ function is selected, the $\overline{\text{RTS}_i}$ pin outputs the following signals:

(1) Clock synchronous serial I/O mode

When the receive enable bit (bit 2 at addresses 35_{16}, $3D_{16}$, $B5_{16}$) = "0" (reception disabled), the $\overline{\text{RTS}_i}$ pin outputs "H" level.
When the receive enable bit = "0" (reception disabled), the $\overline{\text{RTS}_i}$ pin outputs "L" level by setting the receive enable bit to "1," or by reading the low-order byte of the UARTi receive buffer register.
When the receive enable bit = "1" (continuously reception), the $\overline{\text{RTS}_i}$ pin outputs "L" level by reading the low-order byte of the UARTi receive buffer register.
When reception has started, the $\overline{\text{RTS}_i}$ pin outputs "H" level.
When an internal clock is selected (bit 3 at addresses 30_{16}, 38_{16}, $B0_{16}$ = "0"), do not select the $\overline{\text{RTS}}$ function because the $\overline{\text{RTS}}$ output is undefined.

(2) UART mode

When the receive enable bit (bit 2 at addresses 35_{16}, $3D_{16}$, $B5_{16}$) = "0" (reception disabled), the $\overline{\text{RTS}_i}$ pin outputs "H" level.
When the receive enable bit = "0" (reception disabled), the $\overline{\text{RTS}_i}$ pin outputs "L" level by setting the receive enable bit to "1," or by reading the low-order byte of the UARTi receive buffer register.
When the receive enable bit = "1" (continuously reception), the $\overline{\text{RTS}_i}$ pin outputs "L" level by reading the low-order byte of the UARTi receive buffer register.
When reception has started, the $\overline{\text{RTS}_i}$ pin outputs "H" level.

Selection of the $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function depends on the following bits.
•$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at addresses 34_{16}, $3C_{16}$, $B4_{16}$: see Figure 11.2.3.)
•$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ enable bit (bit 4 at addresses 34_{16}, $3C_{16}$, $B4_{16}$: see Figure 11.2.3.)
•$\overline{\text{CTS}_0}$/$\overline{\text{RTS}_0}$ separate select bit (bit 0 at address AC_{16}: see Figure 11.2.13.)
•$\overline{\text{CTS}_1}$/$\overline{\text{RTS}_1}$ separate select bit (bit 1 at address AC_{16}: see Figure 11.2.13.)
•$\overline{\text{CTS}_2}$/$\overline{\text{RTS}_2}$ separate select bit (bit 4 at address AC_{16}: see Figure 11.2.13.)

Table 11.2.1 lists the selection of the $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function.

Table 11.2.1 Selection of $\overline{\text{CTS}}/\overline{\text{RTS}}$ function

$\overline{\text{CTS}}/\overline{\text{RTS}}$ enable bit		0			1
$\overline{\text{CTS}_i}/\overline{\text{RTS}_i}$ separate select bit		0		1	×
$\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit		0	1	×	×
Functions	$P1_0/\overline{\text{CTS}_0}/\overline{\text{RTS}_0}$ pin	$\overline{\text{CTS}_0}$	$\overline{\text{RTS}_0}$	$\overline{\text{RTS}_0}$	$P1_0$
	$P1_1/\overline{\text{CTS}_0}/CLK_0$ pin	$P1_1$ or CLK_0	$P1_1$ or CLK_0	$\overline{\text{CTS}_0}$ **(Notes 2, 3)**	$P1_1$ or CLK_0
	$P1_4/\overline{\text{CTS}_1}/\overline{\text{RTS}_1}$ pin	$\overline{\text{CTS}_1}$	$\overline{\text{RTS}_1}$	$\overline{\text{RTS}_1}$	$P1_4$
	$P1_5/\overline{\text{CTS}_1}/CLK_1$ pin	$P1_5$ or CLK_1	$P1_5$ or CLK_1	$\overline{\text{CTS}_1}$ **(Notes 2, 3)**	$P1_5$ or CLK_1
	$P8_0/AN_8/\overline{\text{CTS}_2}/\overline{\text{RTS}_2}/DA_1$ pin **(Note1)**	$\overline{\text{CTS}_2}$	$\overline{\text{RTS}_2}$	$\overline{\text{RTS}_2}$	$P8_0$, AN_8, or DA_1
	$P8_1/AN_9/\overline{\text{CTS}_2}/CLK_2$ pin	$P8_1$, AN_9 or CLK_2	$P8_1$, AN_9 or CLK_2	$\overline{\text{CTS}_2}$ **(Notes 2, 3)**	$P8_1$, AN_9, or CLK_2

×: It may be either "0" or "1."

Notes 1: When using the $\overline{\text{CTS}_2}/\overline{\text{RTS}_2}$ pin, be sure that the D-A_1 output enable bit (bit 1 at address 96_{16}) = "0" (output disabled).

2: When using the $P1_1$, $P1_5$, or $P8_1$ pin as the $\overline{\text{CTS}_i}$ pin, be sure to clear the corresponding bit of the port P1 or port P8 direction register to "0."

3: When $\overline{\text{CTS}_i}/\overline{\text{RTS}_i}$ separation is selected, the CLK_i pin cannot be used. Accordingly, $\overline{\text{CTS}_i}/\overline{\text{RTS}_i}$ cannot be separated in the clock synchronous serial I/O mode. When separating $\overline{\text{CTS}_i}/\overline{\text{RTS}_i}$ in UART mode, be sure to select an internal clock.

11.3 Clock synchronous serial I/O mode

Table 11.3.1 lists the performance overview in the clock synchronous serial I/O mode, and Table 11.3.2 lists the functions of I/O pins in this mode.

Table 11.3.1 Performance overview in clock synchronous serial I/O mode

Item		Functions
Transfer data format		Transfer data has a length of 8 bits. LSB first or MSB first can be selected by software.
Transfer rate	When selecting internal clock	BRGi's output divided by 2
	When selecting external clock	Maximum 5 Mbps
Transmit/Receive control		$\overline{CTS}$ function or $\overline{RTS}$ function can be selected by software.

Table 11.3.2 Functions of I/O pins in clock synchronous serial I/O mode

Pin name	Functions	Method of selection
TxD_i ($P1_3$, $P1_7$, $P8_3$)	Serial data output pin	$TxD_0/P1_3$, $TxD_1/P1_7$, or $TxD_2/P8_3$ switch bit = "0" (Dummy data is output when performing only reception.) **(Note)**
	Programmable I/O port pin	$TxD_0/P1_3$, $TxD_1/P1_7$, or $TxD_2/P8_3$ switch bit = "1"
RxD_i ($P1_2$, $P1_6$, $P8_2$)	Serial data input pin	Port P1 or P8 direction register's corresponding bit = "0"
	Programmable I/O port pin	– (Can be used as an I/O port pin when performing only transmission.)
CLK_i ($P1_1$, $P1_5$, $P8_1$)	Transfer clock output pin	Internal/External clock select bit = "0"
	Transfer clock input pin	Internal/External clock select bit = "1"
$\overline{CTS_i}$, $\overline{RTS_i}$ ($P1_0$, $P1_1$, $P1_4$, $P1_5$, $P8_0$, $P8_1$)	$\overline{CTS}$ input pin	See Table 11.2.1.
	$\overline{RTS}$ output pin	
	Programmable I/O port	

Port P1 direction register: address 05_{16}
Port P8 direction register: address 14_{16}
Internal/External clock select bit: bit 3 at addresses 30_{16}, 38_{16}, $B0_{16}$
$TxD_0/P1_3$ switch bit: bit 2 at address AC_{16}
$TxD_1/P1_7$ switch bit: bit 3 at address AC_{16}
$TxD_2/P8_3$ switch bit: bit 5 at address AC_{16}
Note: The TxD_i pin outputs "H" level until transmission starts after UARTi's operating mode is selected.

11.3.1 Transfer clock (Synchronizing clock)

Data transfer is performed synchronously with a transfer clock. For the transfer clock, the following selection is possible:

- Whether to generate a transfer clock internally or to input it from the external.
- Polarity of transfer clock.

The transfer clock is generated by operation of the transmit control circuit. Accordingly, even when performing only reception, set the transmit enable bit to "1," and set dummy data in the UARTi transmit buffer register in order to make the transmit control circuit active.

(1) Internal generation of transfer clock

The count source selected with the BRG count source select bits is divided by the BRGi, and the BRGi output is further divided by 2. This divided output is the transfer clock. The transfer clock is output from the CLK_i pin.

$$\text{Transfer clock's frequency} = \frac{f_i}{2\,(n+1)}$$

f_i: Frequency of BRGi's count source (f_2, f_{16}, f_{64}, or f_{512})
n: Setting value of BRGi

(2) Input of transfer clock from the external

A clock input from the CLK_i pin becomes the transfer clock.

(3) Porarity of transfer clock

As shown in Figure 11.3.1, the polarity of the transfer clock can be selected by the CLK polarity select bit (bit 6 at addresses 34_{16}, $3C_{166}$, $B4_{16}$).

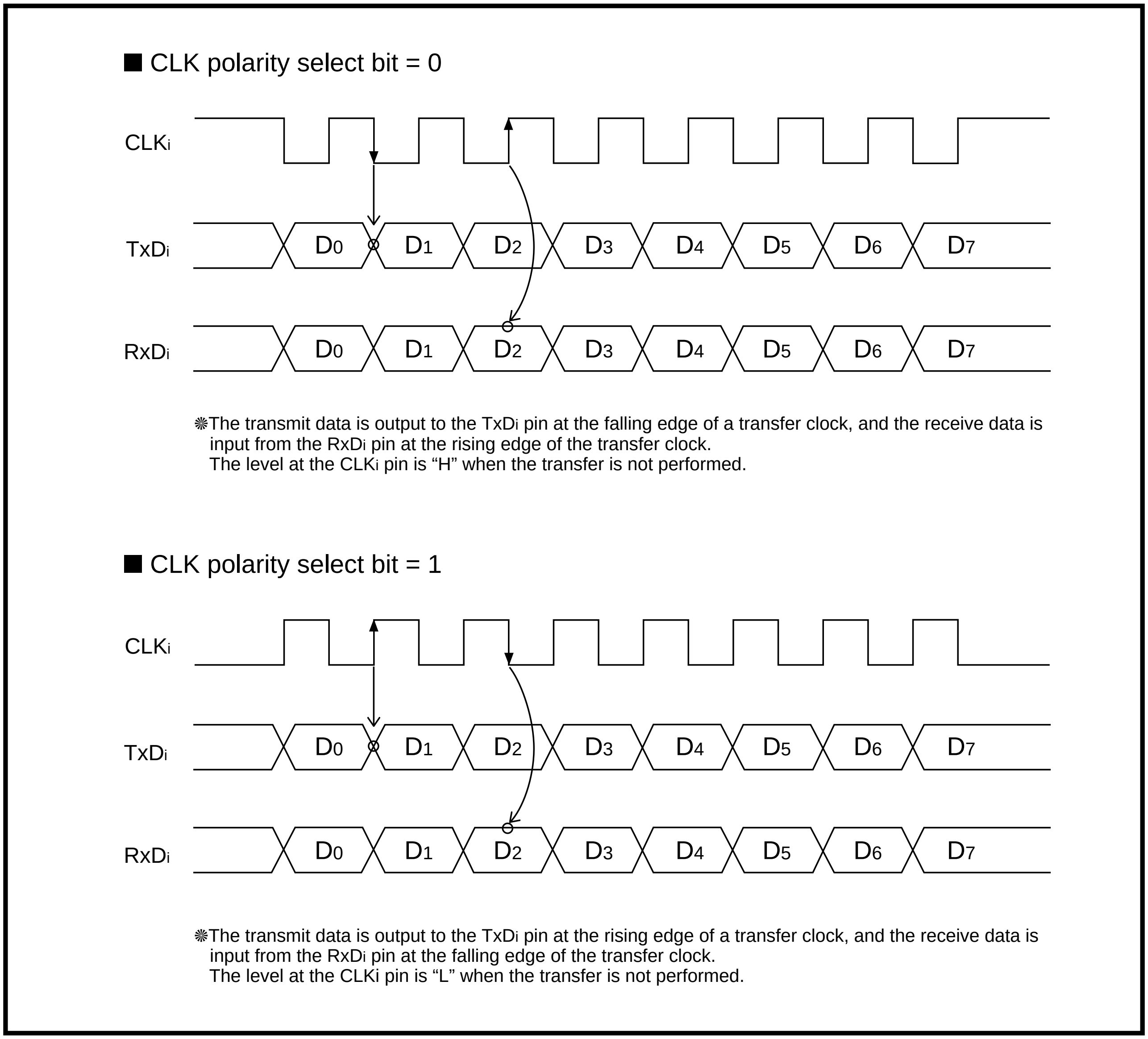


Fig. 11.3.1 Polarity of transfer clock

11.3.2 Transfer data format

LSB first or MSB first can be selected as the transfer data format. Table 11.3.3 lists the relationship between the transfer data format and writing/reading to and from the UARTi transmit/receive buffer register. The transfer format select bit (bit 7 at addresses 34_{16}, $3C_{16}$, $B4_{16}$) selects the transfer data format. When this bit is cleared to “0,” the set data is written to the UARTi transmit buffer register as the transmit data, as it is. Similarly, the data in the UARTi receive buffer register is read out as the receive data, as it is. (See the upper row in Table 11.3.3.) When this bit is set to “1,” each bit’s position of set data is reversed, and the resultant data will be written to the UARTi transmit buffer register as the transmit data. Similarly, each bit’s position of data in the UARTi receive buffer register is reversed, and the resultant data will be read out as the receive data. (See the lower row in Table 11.3.3.)

Note that only the method of writing/reading to and from the UARTi transmit/receive buffer register is affected by selection of the transfer data format, and that the transmit/receive operation is unaffected by it.

Table 11.3.3 Relationship between transfer data format and writing/reading to and from UARTi transmit/ receive buffer register

Transfer format select bit	Transfer data format	Writing to UARTi transmit buffer register	Reading from UARTi receive buffer register
0	LSB (Least Significant Bit) first	Data bus → UARTi transmit buffer register DB_7 → D_7 DB_6 → D_6 DB_5 → D_5 DB_4 → D_4 DB_3 → D_3 DB_2 → D_2 DB_1 → D_1 DB_0 → D_0	Data bus ← UARTi receive buffer register DB_7 ← D_7 DB_6 ← D_6 DB_5 ← D_5 DB_4 ← D_4 DB_3 ← D_3 DB_2 ← D_2 DB_1 ← D_1 DB_0 ← D_0
1	MSB (Most Significant Bit) first	Data bus → UARTi transmit buffer register DB_7 → D_0 DB_6 → D_1 DB_5 → D_2 DB_4 → D_3 DB_3 → D_4 DB_2 → D_5 DB_1 → D_6 DB_0 → D_7	Data bus ← UARTi receive buffer register DB_7 ← D_0 DB_6 ← D_1 DB_5 ← D_2 DB_4 ← D_3 DB_3 ← D_4 DB_2 ← D_5 DB_1 ← D_6 DB_0 ← D_7

11.3.3 Method of transmission

Figure 11.3.2 shows an initial setting example for relevant registers when transmitting. Transmission is started when all of the following conditions (① to ③) has been satisfied. When an external clock is selected, satisfy conditions ① to ③ with the following preconditions satisfied.

<Preconditions>
The CLK_i pin's input is at "H" level (External clock selected, when the CLK polarity select bit = "0")
The CLK_i pin's input is at "L" level (External clock selected, when the CLK polarity select bit = "1")
Note: When an internal clock is selected, the above preconditions are ignored.

① Transmit data is present in the UARTi transmit buffer register (transmit buffer empty flag = "0")
② Transmission is enabled (transmit enable bit = "1").
③ The $\overline{CTS_i}$ pin's input is at "L" level (when the $\overline{CTS}$ function selected).
Note: When the $\overline{CTS}$ function is not selected, condition ③ is ignored.

By connecting the $\overline{RTS_i}$ pin (receiver side) and $\overline{CTS_i}$ pin (transmitter side), the timing of transmission and that of reception can be matched. For details, refer to section **"11.3.6 Receive operation."**
When using interrupts, it is necessary to set the relevant registers to enable interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."**
Figure 11.3.3 shows the write operation of data after transmission start, and Figure 11.3.4 shows the detect operation of transmit completion.

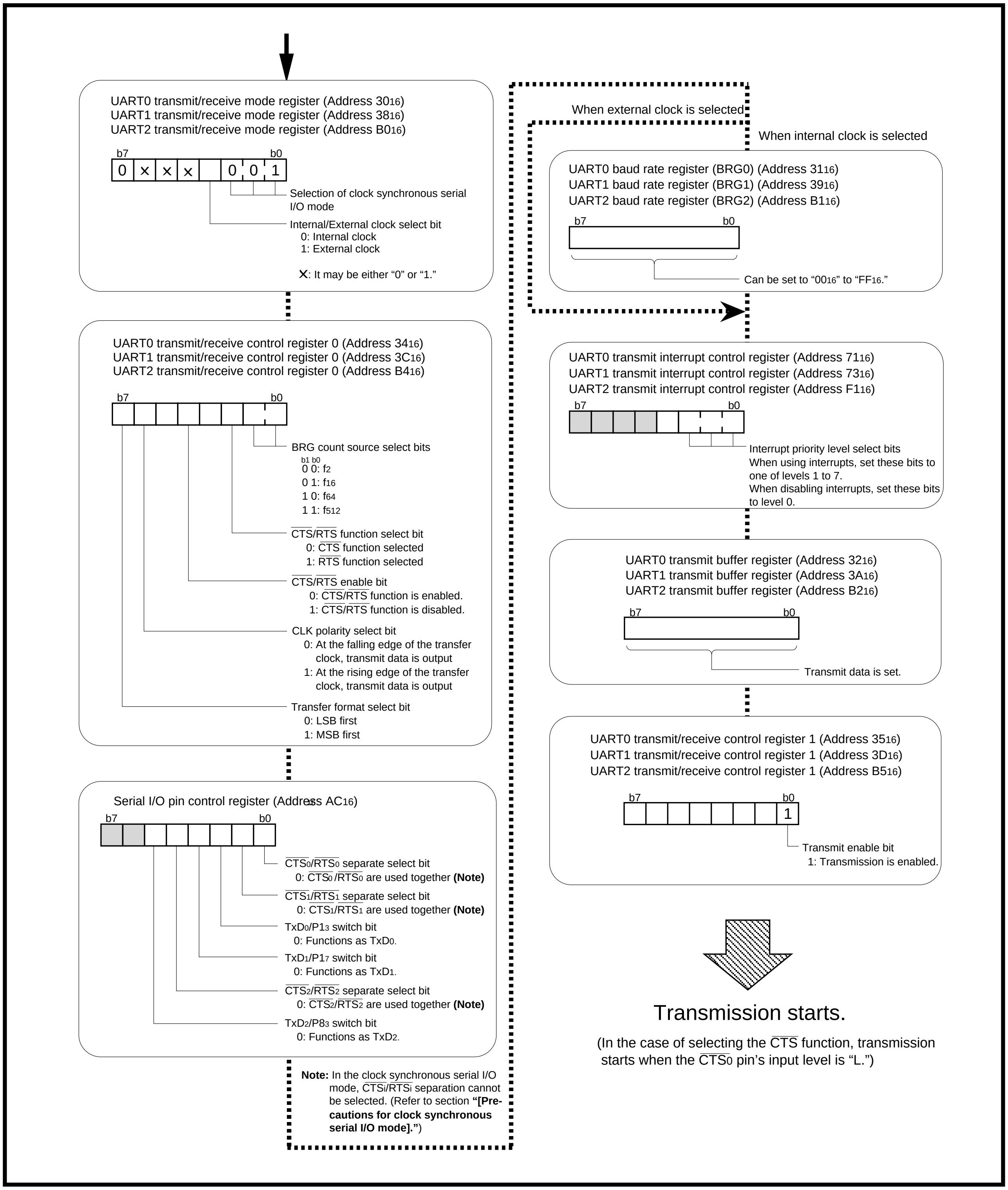


Fig. 11.3.2 Initial setting example for relevant registers when transmitting

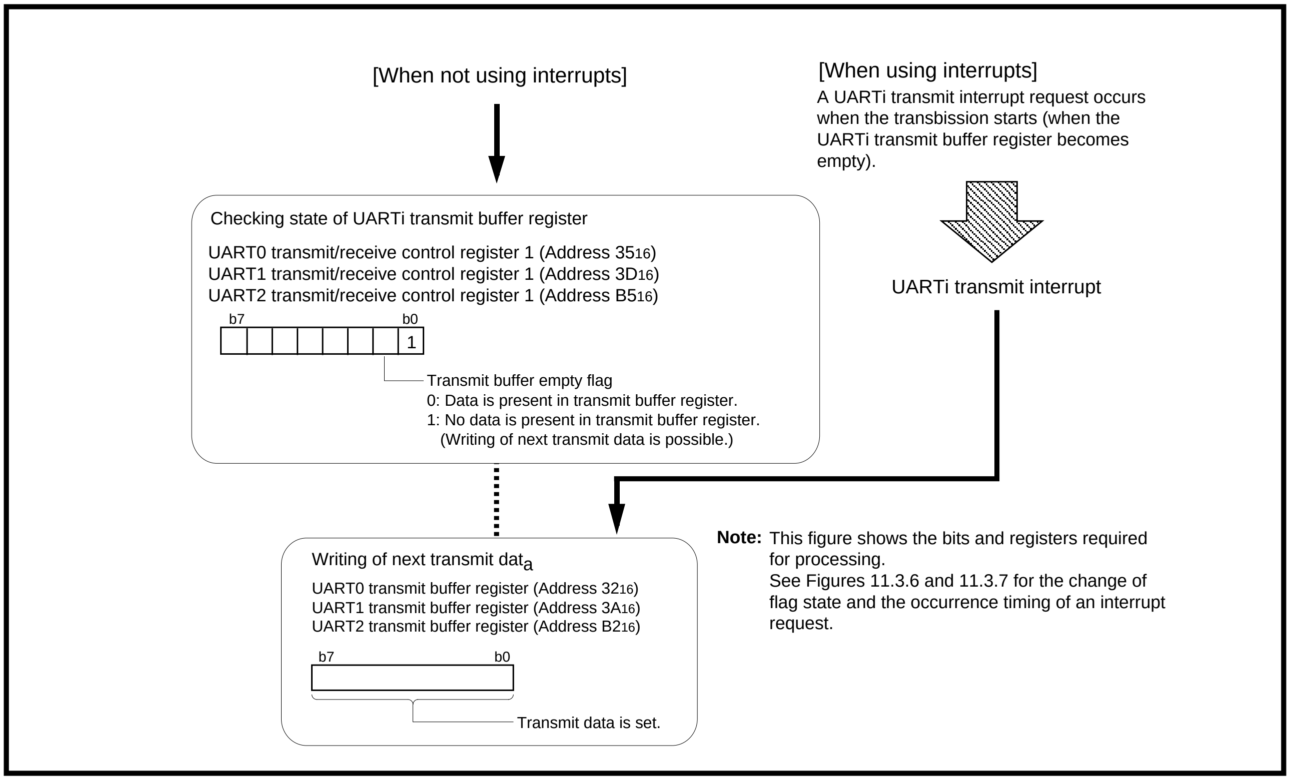


Fig. 11.3.3 Write operation of data after transmission start

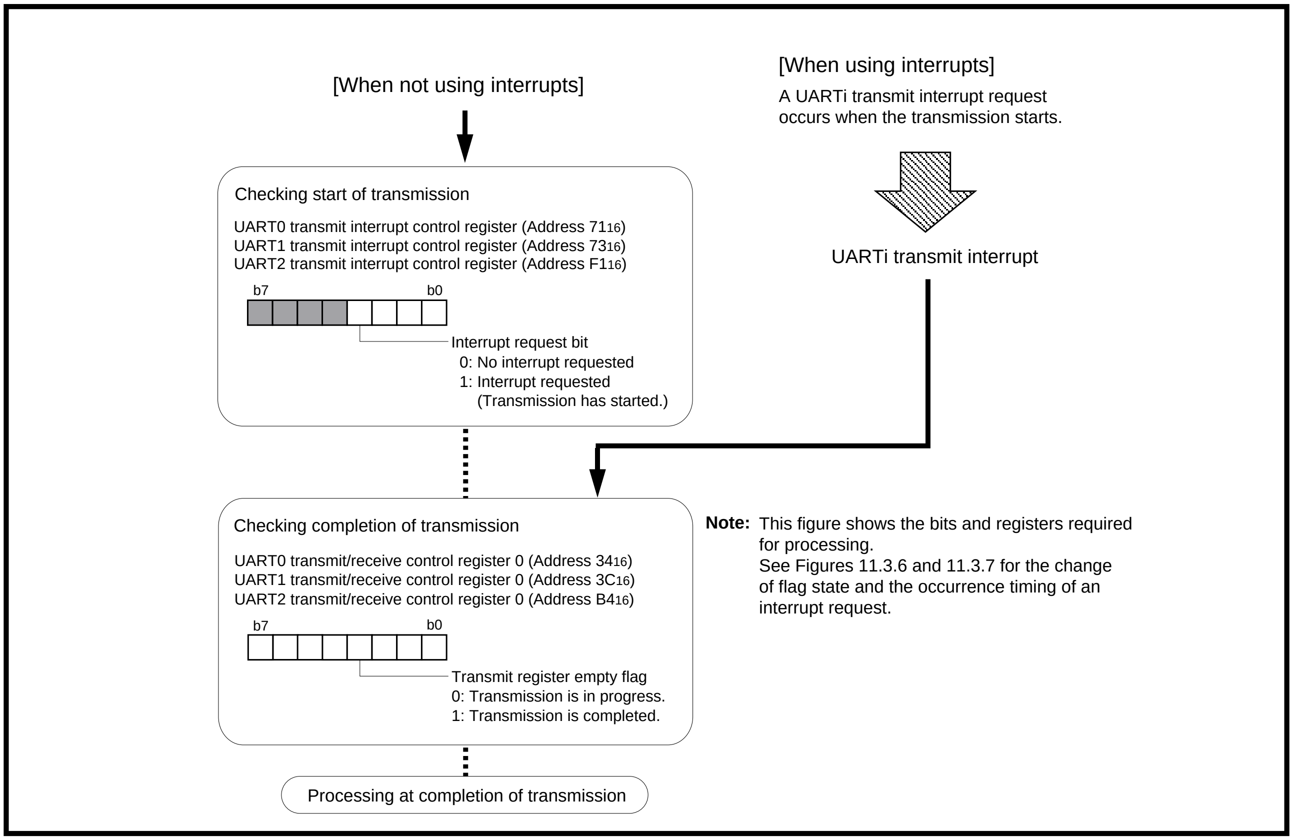


Fig. 11.3.4 Detect operation of transmit completion

11.3.4 Transmit operation

When the transmit conditions described in section **"11.3.3 Method of transmission"** have been satisfied in the case of an internal clock selected, a transfer clock is generated and the following operations are automatically performed after 1 cycle of the transfer clock or less has passed. In the case of an external clock selected, when the transmit conditions have been satisfied and then an external clock is input to the CLK_i pin, the following operations are automatically performed:

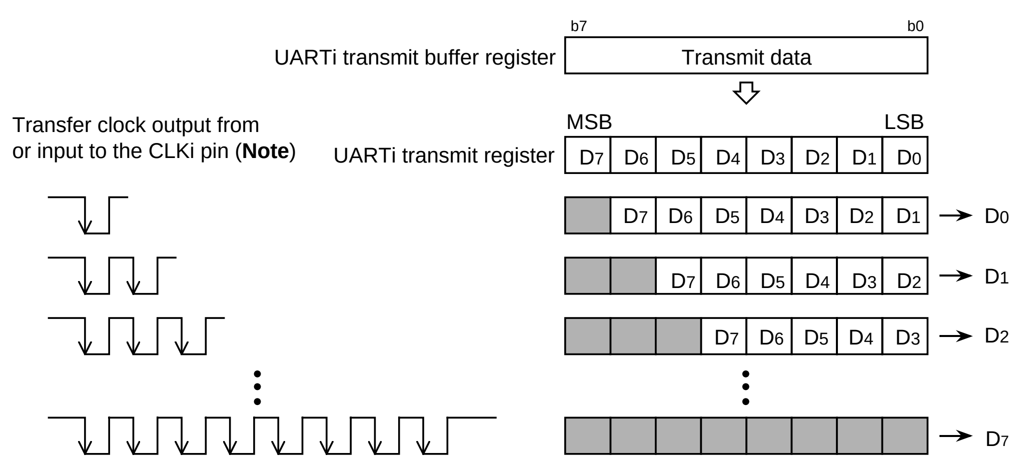
•The UARTi transmit buffer register's contents are transferred to the UARTi transmit register.
•The transmit buffer empty flag is set to "1."
•The transmit register empty flag is cleared to "0."
•8 transfer clocks are generated (in the case of an internal clock selected).
•A UARTi transmit interrupt request occurs, and the interrupt request bit is set to "1."

The transmit operations are described below:

① Data in the UARTi transmit register is transmitted from the TxD_i pin synchronously with the valid edge* of the clock output from or input to the CLK_i pin.
② This data is transmitted, bit by bit, sequentially beginning with the least significant bit.
③ When 1-byte data has been transmitted, the transmit register empty flag is set to "1." This indicates the completion of transmission.

Valid edge* : A falling edge is selected when the CLK polarity select bit = "0."
A rising edge is selected when the CLK polarity select bit = "1."

Figure 11.3.5 shows the transmit operation.
When an internal clock is selected, if the transmit conditions for the next data are satisfied at completion of the transmission, the transfer clock is generated continuously. Accordingly, when performing transmission continuously, set the next transmit data to the UARTi transmit buffer register during transmission (when the transmit register empty flag = "0"). When the transmit conditions for the next data are not satisfied, the transfer clock stops at "H" level (when the CLK polarity select bit = "0"), or "L" level (when the CLK polarity select bit = "1").
Figures 11.3.6 and 11.3.7 show examples of transmit timing.



Note: This applies when the CLK polarity select bit = "0."
When the CLK polarity select bit = "1," data is shifted at the rising edge of the transfer clock.

Fig. 11.3.5 Transmit operation

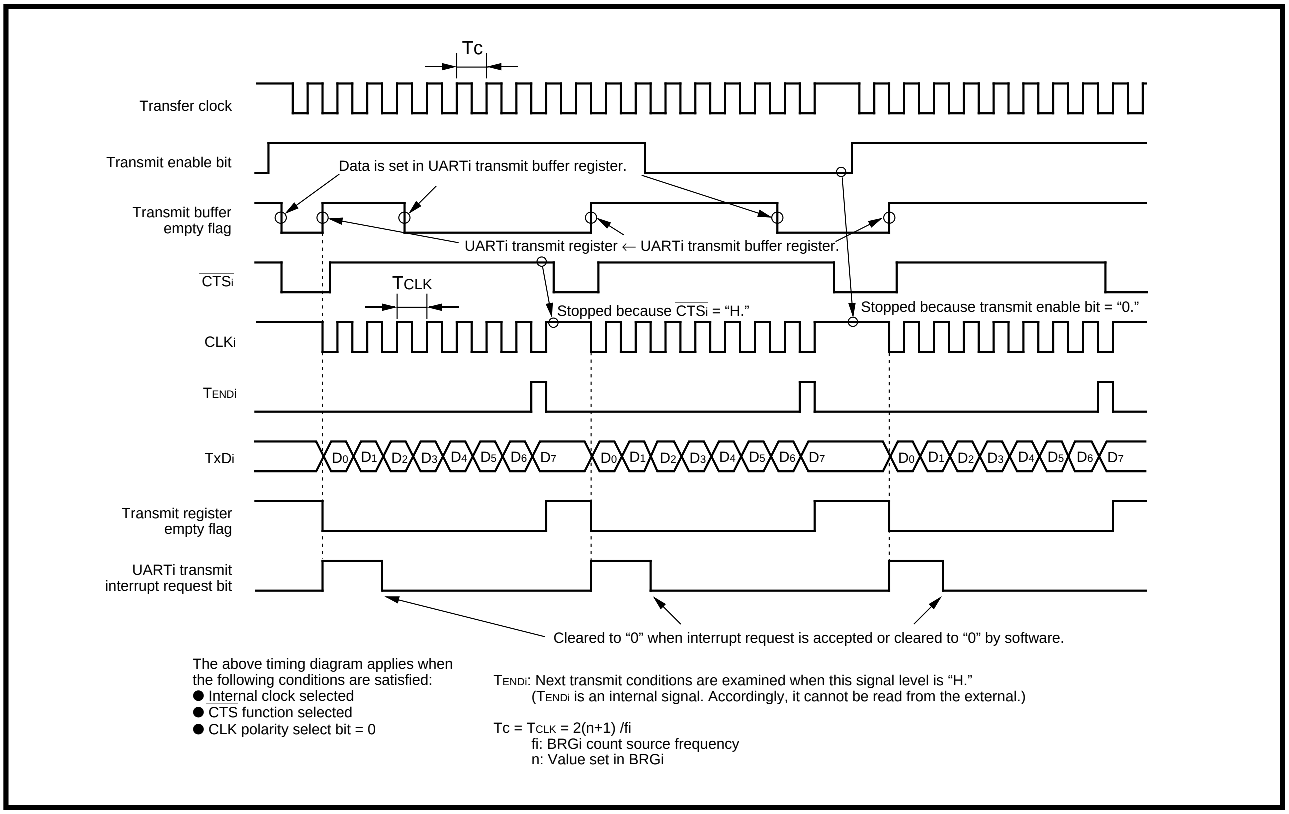


Fig. 11.3.6 Example of transmit timing (when internal clock and $\overline{CTS}$ function selected)

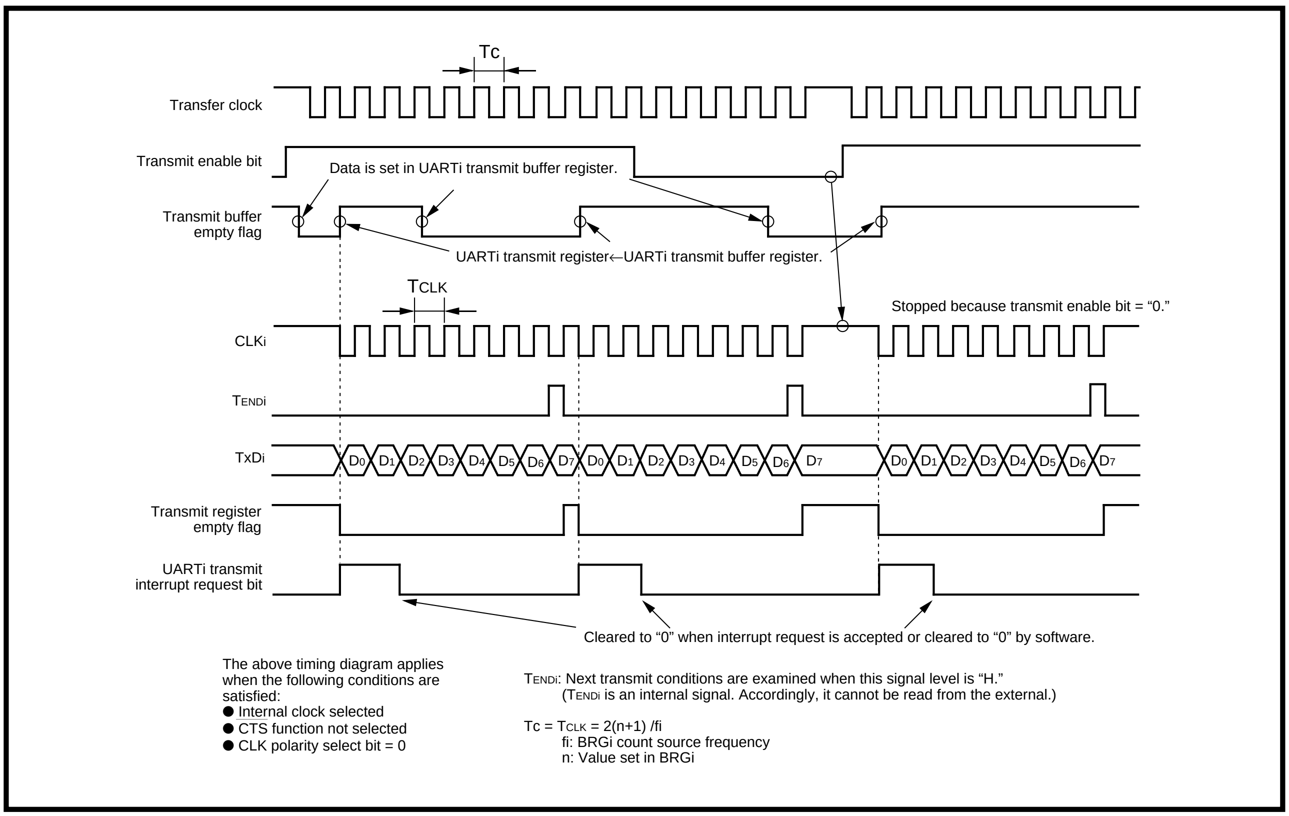


Fig. 11.3.7 Example of transmit timing (when internal clock selected and $\overline{CTS}$ function not selected)

11.3.5 Method of reception

Figure 11.3.8 shows an initial setting example for relevant registers when receiving. Reception is started when all of the following conditions (① to ③) have been satisfied. When an external clock is selected, satisfy conditions ① to ③ with the following preconditions satisfied.

<Preconditions>
The CLK_i pin's input is at "H" level (External clock selected, when the CLK polarity select bit = "0").
The CLK_i pin's input is at "L" level (External clock selected, when the CLK polarity select bit = "1").
Note: When an internal clock is selected, the above preconditions are ignored.

① Dummy data is present in the UARTi transmit buffer register (transmit buffer empty flag = "0")
② Reception is enabled (receive enable bit = "1").
③ Transmission is enabled (transmit enable bit = "1").

By connecting the $\overline{RTS_i}$ pin (receiver side) and $\overline{CTS_i}$ pin (transmitter side), the timing of transmission and that of reception can be matched. For details, refer to section **"11.3.6 Receive operation."**
When using interrupts, it is necessary to set the relevant registers to enable interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."**
Figure 11.3.9 shows processing after reception is completed.

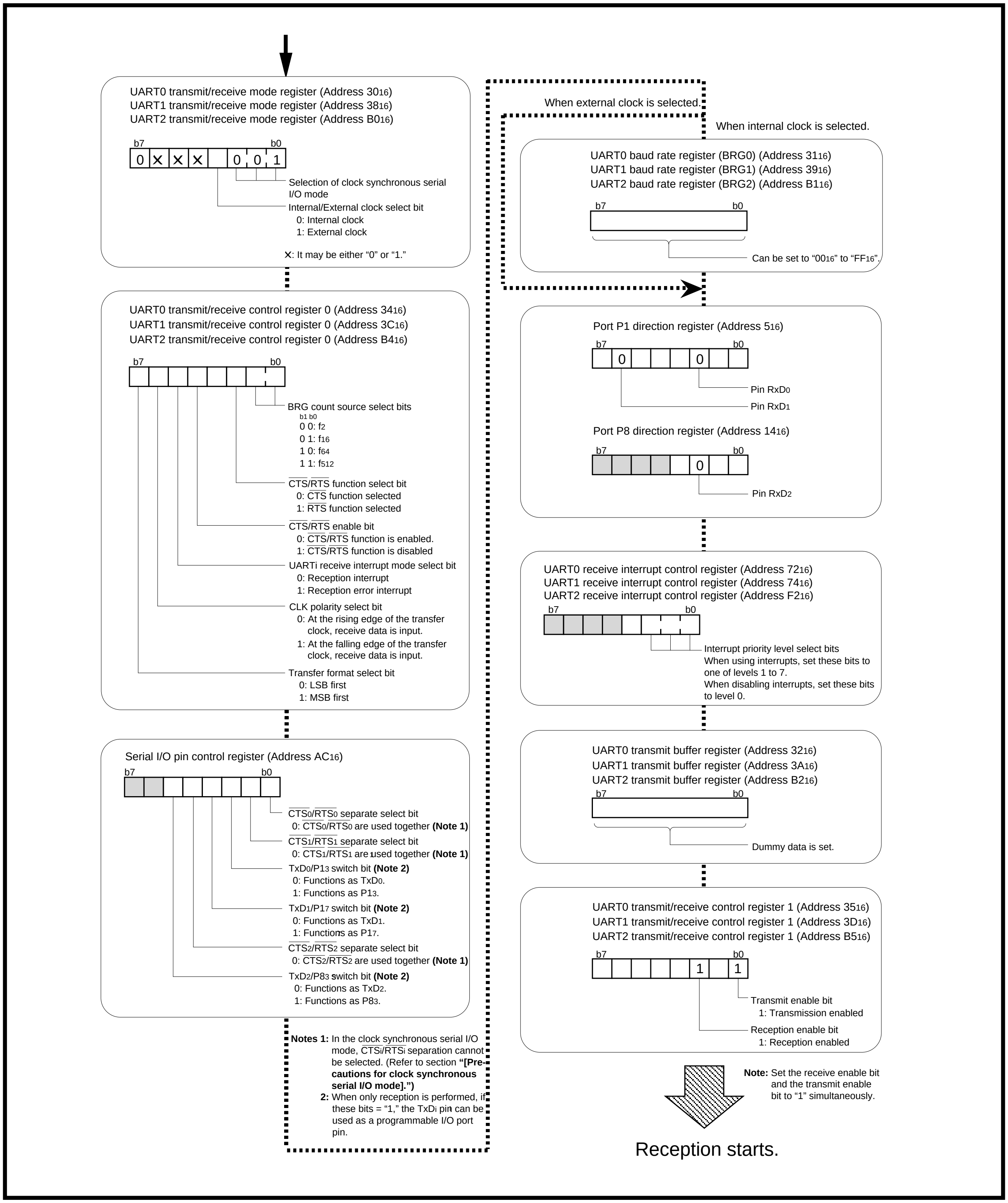
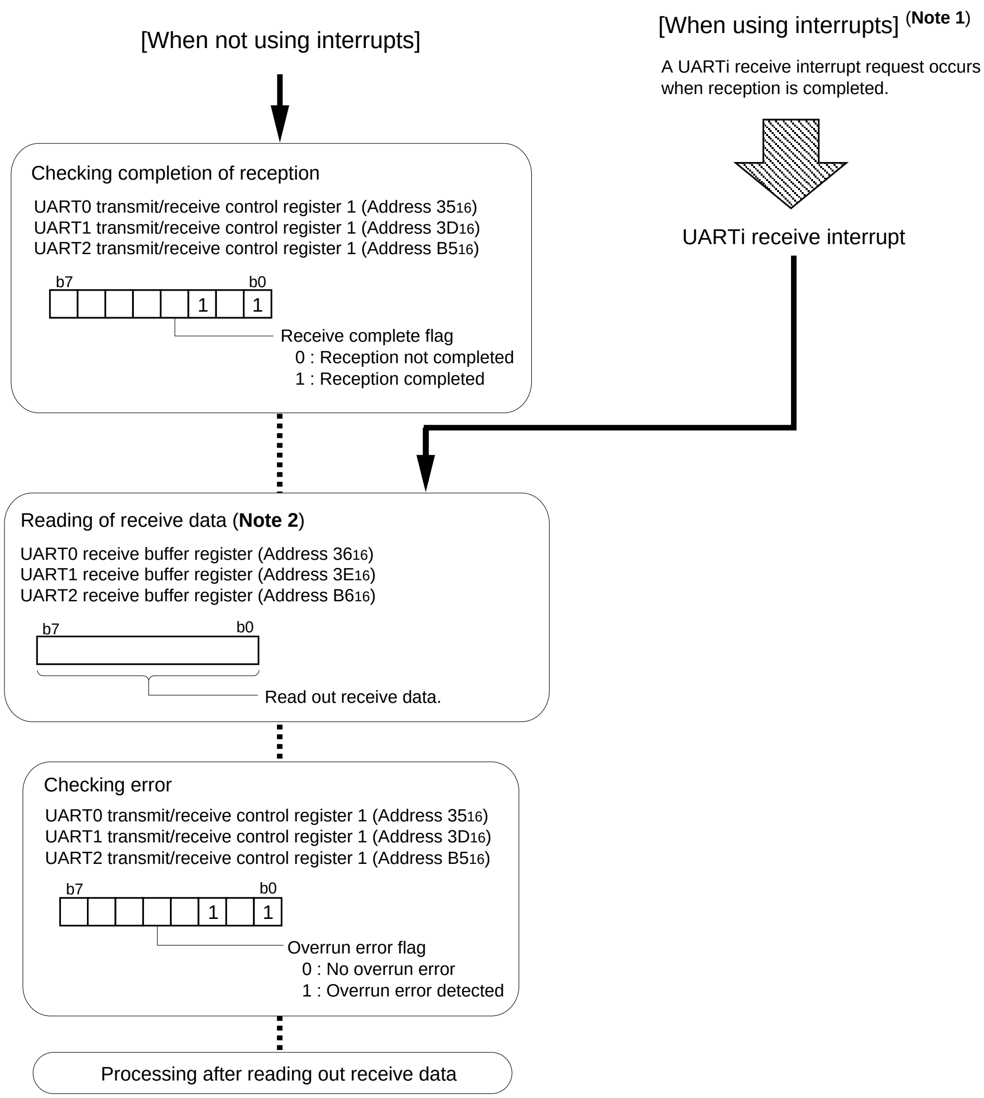


Fig. 11.3.8 Initial setting example for relevant registers when receiving



Notes 1: When performing the processing after reception is completed, using an interrupt, be sure to select a receive interrupt (UARTi receive interrupt mode select bit = “0.”)

2: In the case of an external clock and the $\overline{RTS}$ function selected, the $\overline{RTS_i}$ output level becomes “L” when the UARTi receive buffer register is read out. Accordingly, when performing reception continuously, be sure to write the dummy data to the UARTi transmit buffer register before reading out the UARTi receive buffer register.

3: This figure shows the bits and registers required for the processing. See Figure 11.3.12 for the change of flag state and the occurrence timing of an interrupt request.

Fig. 11.3.9 Processing after reception is completed

11.3.6 Receive operation

In the case of an internal clock selected, when the receive conditions described in section **"11.3.5 Method of reception"** have been satisfied, a transfer clock is generated and the reception is started after 1 cycle of the transfer clock or less has passed. In the case of an external clock selected, when the receive conditions have been satisfied, the UARTi enters the receive-enabled state, and then reception will be started when an external clock is input to the CLKi pin.

In the case of an external clock selected, when connecting the $\overline{RTS_i}$ pin to the $\overline{CTS_i}$ pin of the transmitter side, the timing of transmission and that of reception can be matched. In the case of an internal clock selected, do not use the $\overline{RTS}$ function. It is because the $\overline{RTS}$ output is undefined in the case of an internal clock selected.

In the case of an external clock and the $\overline{RTS}$ function selected, the $\overline{RTS_i}$ pin's output level becomes as described below.

When the receive enable bit = "0," if one of the following is performed, the $\overline{RTS_i}$ pin's output level becomes "L" and informs of the transmitter side that reception has become enabled:
- The receive enable bit is set to "1."
- The low-order byte of the UARTi receive buffer register is read out.

When the receive enable bit = "1," if the low-order byte of the UARTi receive buffer register is read out, the $\overline{RTS_i}$ pin's output level becomes "L."

Accordingly, when performing reception continuously, an overrun occurrence can be avoided because the $\overline{RTS}$ output level does not become "L" until the receive data is read out.

When reception has started, the $\overline{RTS_i}$ pin's output level becomes "H."

Figure 11.3.10 shows a connection example.

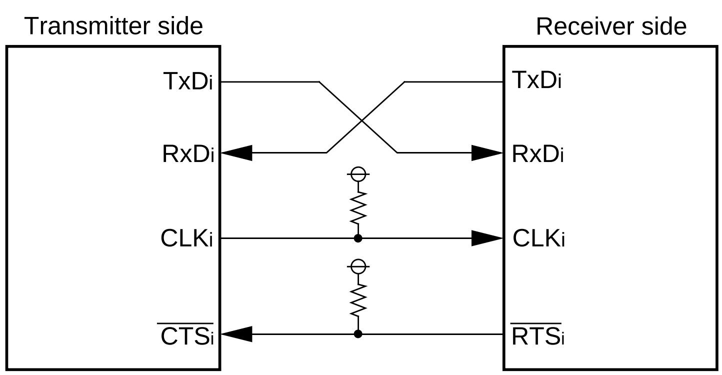


Fig. 11.3.10 Connection example

The receive operations are described below:

① The signal input to the RxD_i pin is taken into the most significant bit of the UARTi receive register synchronously with the valid edge* of the clock output from the CLK_i pin or input to the CLK_i pin.
② The contents of the UARTi receive register are shifted, bit by bit, to the right.
③ Steps ① and ② are repeated at each valid edge of the clock output from the CLK_i pin or input to the CLK_i pin.
④ When 1-byte data has been prepared in the UARTi receive register, the contents of this register are transferred to the UARTi receive buffer register.
⑤ Simultaneously with step ④, the receive complete flag is set to "1." Additionally, when the receive interrupt is selected (UARTi receive interrupt mode select bit = "0"), a UARTi receive interrupt request occurs and its interrupt request bit is set to "1."

Valid edge* : A rising edge is selected when the CLK polarity select bit = "0."
A falling edge is selected when the CLK polarity select bit = "1."

The receive complete flag is cleared to "0" when the low-order byte of the UARTi receive buffer register is read out. Figure 11.3.11 shows the receive operation, and Figure 11.3.12 shows an example of receive timing (when an external clock is selected).
When the transfer format select bit is "1" (MSB first), each bit's position of this register's contents is reversed, and then the resultant data is read out.

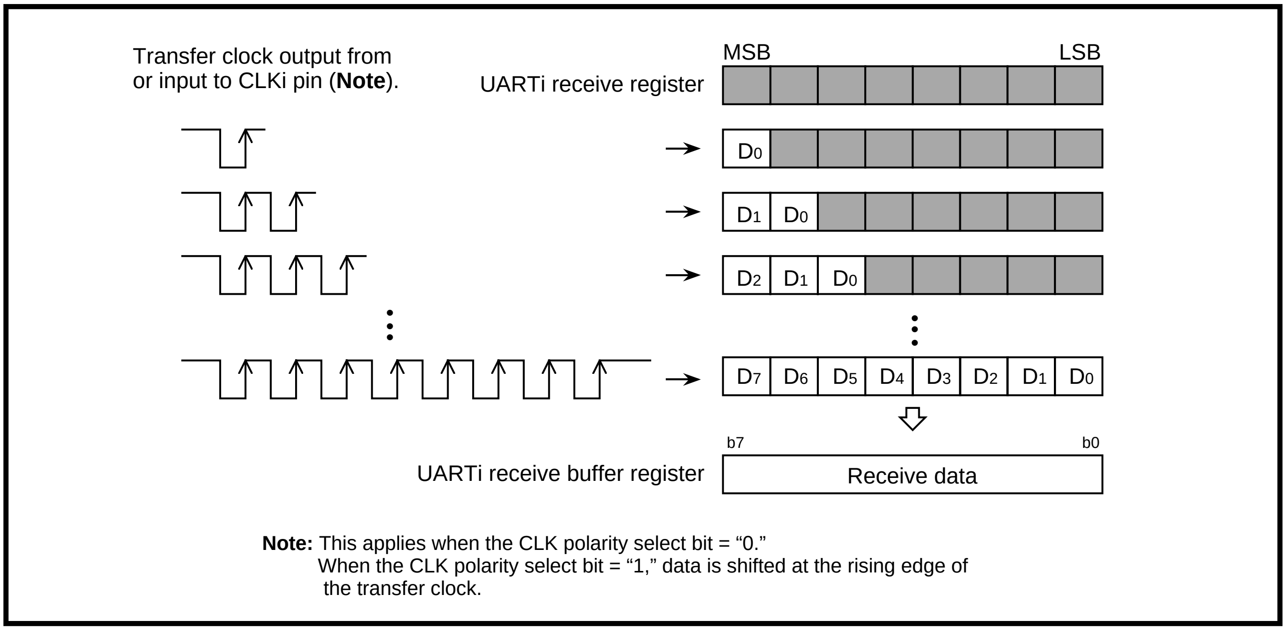


Fig. 11.3.11 Receive operation

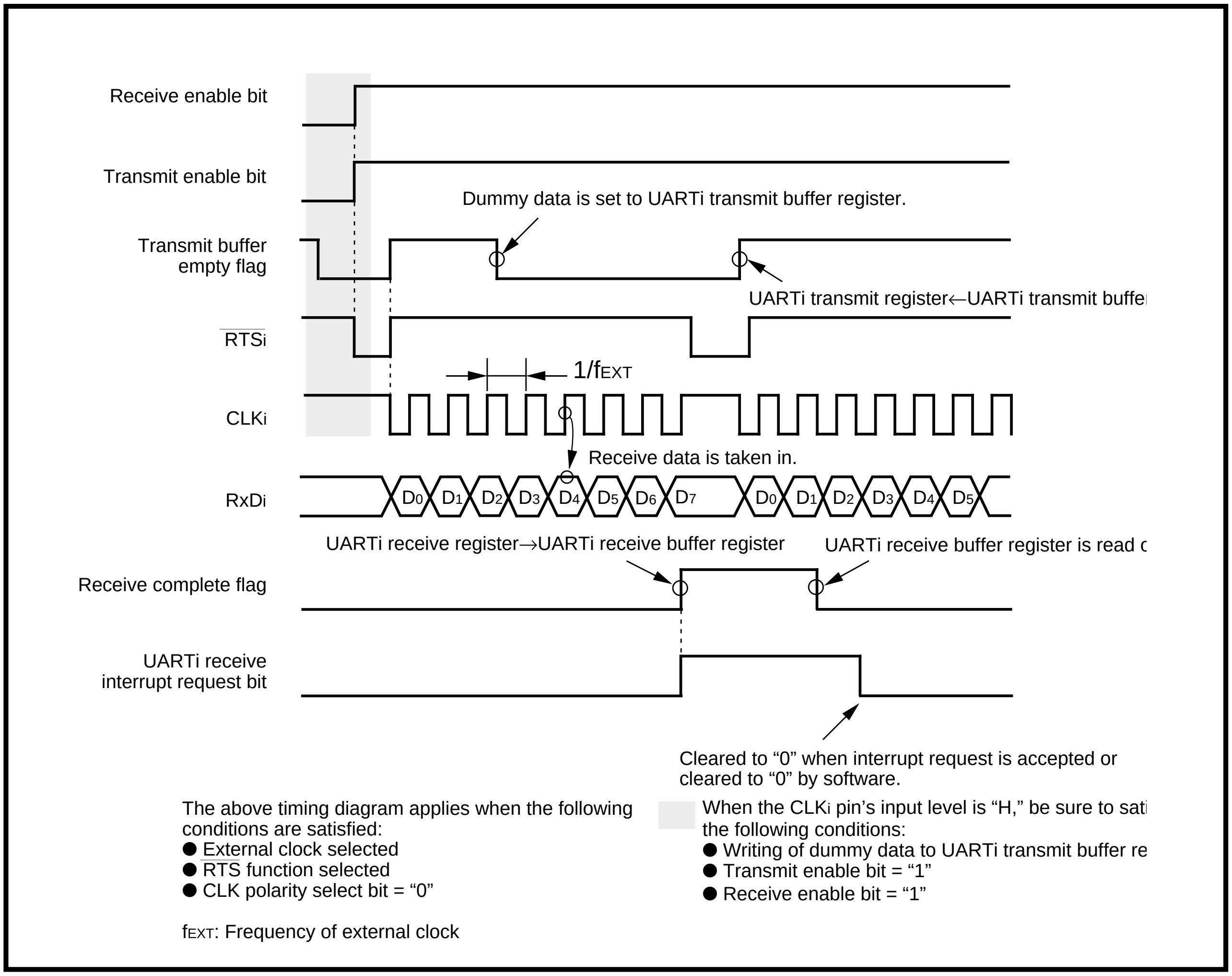


Fig. 11.3.12 Example of receive timing (when external clock selected)

11.3.7 Processing on detecting overrun error

In the clock synchronous serial I/O mode, an overrun error can be detected.
An overrun error occurs when the next data has been prepared in the UARTi receive register with the receive complete flag = “1” (i.e. data is present in the UARTi receive buffer register) and next data is transferred to the UARTi receive buffer register. In other words, an overrun error occurs when the next data has been prepared before reading out the contents of the UARTi receive buffer register. When an overrun error has occurred, the next receive data is written into the UARTi receive buffer register. Additionally, when the receive error interrupt is selected (UARTi receive interrupt mode select bit = “1”), a UARTi receive interrupt request occurs and its interrupt request bit is set to “1.” When the receive interrupt is selected (UARTi receive interrupt mode select bit = “0”), the UARTi receive interrupt request bit does not change.
An overrun error is detected when data is transferred from the UARTi receive register to the UARTi receive buffer register, and the overrun error flag is set to “1.” The overrun error flag is cleared to “0” by clearing the receive enable bit to “0.”

When an overrun error occurs during reception, be sure to initialize the overrun error flag and UARTi receive buffer register, and then perform reception again. When it is necessary to perform retransmission owing to a receiver-side overrun error which has occurred during transmission, be sure to set the UARTi transmit buffer register again, and start transmission again.
The methods of initializing the UARTi receive buffer register and that of setting the UARTi transmit buffer register again are described below.

(1) Method of initializing UARTi receive buffer register

① Clear the receive enable bit to “0” (reception disabled).
② Set the receive enable bit to “1” again (reception enabled).

(2) Method of setting UARTi transmit buffer register again

① Clear the serial I/O mode select bits to “000_2” (serial I/O invalidated).
② Set the serial I/O mode select bits to “001_2” again.
③ Set the transmit enable bit to “1” (transmission enabled), and set the transmit data to the UARTi transmit buffer register.

[Precautions for clock synchronous serial I/O mode]

1. A transfer clock is generated by operation of the transmit control circuit. Accordingly, even when performing only reception, the transmit operation (in other words, setting for transmission) must be performed. In this case, be sure to set as follows. Additionally, in this case, dummy data is output from the TxD_i pin to the external:
   - When performing reception, be sure to enable the reception after dummy data is set to the low-order byte of the UARTi transmit buffer register. Also, be sure to set dummy data at each 1-byte data reception.
   - At reception, be sure to set the receive enable bit and transmit enable bit to “1” simultaneously.

   When performing only reception, if any of the $TxD_0/P1_3$, $TxD_1/P1_7$ and $TxD_2/P8_3$ switch bits (bits 2, 3 and 5 at address AC_{16}) is set to “1,” the corresponding TxD_i pin can be used as a programmable I/O port pin.

2. When an external clock is selected, with the input level at the CLK_i pin = “H” (the CLK polarity select bit = “0”) or “L” (the CLK polarity select bit = “1”), be sure to satisfy all of the following three conditions:

   <At transmission>
   ① Transmit data is written to the UARTi transmit buffer register.
   ② The transmit enable bit is set to “1.”
   ③ “L” level is input to the $\overline{CTS_i}$ pin (when the $\overline{CTS}$ function selected).

   <At reception>
   ① Dummy data is written to the UARTi transmit buffer register.
   ② The receive enable bit is set to “1.”
   ③ The transmit enable bit is set to “1.”

3. When using the $\overline{CTS_2}/\overline{RTS_2}$ pin, be sure that the D-A_1 output enable bit (bit 1 at address 96_{16}) = “0” (output disabled).

4. While the $\overline{CTS_i}/\overline{RTS_i}$ separation is selected, the CLK_i pin cannot be used. Accordingly, in the clock synchronous serial I/O mode, the $\overline{CTS_i}/\overline{RTS_i}$ separation cannot be selected.

5. Writing to the UARTi baud rate register (BRGi) must be performed while transmission/reception halts.

6. When an internal clock is selected, do not use the $\overline{RTS}$ function because the $\overline{RTS}$ output is undefined.

7. When performing transmission, be sure to clear any of the $TxD_0/P1_3$, $TxD_1/P1_7$, and $TxD_2/P8_3$ switch bits to “0” (bits 2, 3, and 5 at address AC_{16}).

11.4 Clock asynchronous serial I/O (UART) mode

Table 11.4.1 lists the performance overview in the UART mode, and Table 11.4.2 lists the functions of I/O pins in this mode.

Table 11.4.1 Performance overview in UART mode

Item		Functions
Transfer data format	Start bit	1 bit
	Character bit (Transfer data)	7 bits, 8 bits, or 9 bits
	Parity bit	0 bit or 1 bit (Odd or Even can be selected.)
	Stop bit	1 bit or 2 bits
Transfer rate	When selecting internal clock	BRGi's output divided by 16
	When selecting external clock	Maximum 312.5 kbps
Error detection		4 types (overrun, framing, parity, and summing): presence of an error can be detected only by check of the error sum flag.

Table 11.4.2 Functions of I/O pins in UART mode

Pin name	Functions	Method of selection
TxD_i ($P1_3$, $P1_7$, $P8_3$)	Serial data output pin	$TxD_0/P1_3$, $TxD_1/P1_7$, or $TxD_2/P8_3$ switch bit = "0." **(Note)**
	Programmable I/O port pin	$TxD_0/P1_3$, $TxD_1/P1_7$, or $TxD_2/P8_3$ switch bit = "1."
RxD_i ($P1_2$, $P1_6$, $P8_2$)	Serial data input pin	Port P1 or P8 direction register's corresponding bit = "0"
	Programmable I/O port pin	– (Can be used as a programmable I/O port pin when performing only transmission.)
CLK_i ($P1_1$, $P1_5$, $P8_1$)	BRGi's count source input pin	Internal/External clock select bit = "1"
	Programmable I/O port pin	Internal/External clock select bit = "0"
$\overline{CTS_i}/\overline{RTS_i}$ ($P1_0$, $P1_1$, $P1_4$, $P1_5$, $P8_0$, $P8_1$)	$\overline{CTS}$ input pin	See Table 11.2.1.
	$\overline{RTS}$ output pin	
	Programmable I/O port pin	

Port P1 direction register: address 05_{16}
Port P8 direction register: address 14_{16}
Internal/External clock select bit: bit 3 at addresses 30_{16}, 38_{16}, and $B0_{16}$
$TxD_0/P1_3$ switch bit: bit 2 at address AC_{16}
$TxD_1/P1_7$ switch bit: bit 3 at address AC_{16}
$TxD_2/P8_3$ switch bit: bit 5 at address AC_{16}

Note: The TxD_i pin outputs "H" level while transmission is not performed after the UARTi's operating mode is selected.

11.4 Clock asynchronous serial I/O (UART) mode

11.4.1 Transfer rate (Frequency of transfer clock)

The transfer rate is determined by the BRGi (addresses 31_{16}, 39_{16}, $B1_{16}$).

When "n" is set into BRGi, BRGi divides the count source frequency by (n + 1). The BRGi's output is further divided by 16, and the resultant clock becomes the transfer clock. Accordingly, "n" is expressed by the following formula.

$$n = \frac{F}{16 \times B} - 1$$

n: Value set in BRGi (00_{16} to FF_{16})
F: BRGi's count source frequency (Hz)
B: Transfer rate (bps)

An internal clock or an external clock can be selected as the BRGi's count source with the internal/external clock select bit (bit 3 at addresses 30_{16}, 38_{16}, $B0_{16}$). When an internal clock is selected, the clock selected with the BRG count source select bits (bits 0 and 1 at addresses 34_{16}, $3C_{16}$, $B4_{16}$) becomes the BRGi's count source. When an external clock is selected, the clock input to the CLK_i pin becomes the BRGi's count source.

Be sure to set the same transfer rate for both transmitter and receiver sides. Tables 11.4.3 and 11.4.4 list the setting examples of transfer rate.

Each of the values, listed in these tables, realizes the actual transfer rate of which error toward an ideal transfer rate is within 1 %.

Table 11.4.3 Setting examples of transfer rate (1)

Transfer rate (bps)	f_{sys} = 19.6608 MHz			f_{sys} = 20 MHz		
	BRGi's count source	BRGi's set value: n **(Note)**	Actual time (bps)	BRGi's count source	BRGi's set value: n **(Note)**	Actual time (bps)
300	f_{64}	63 ($3F_{16}$)	300.00	f_{64}	64 (40_{16})	300.48
600	f_{16}	127 ($7F_{16}$)	600.00	f_{16}	129 (81_{16})	600.96
1200	f_{16}	63 ($3F_{16}$)	1200.00	f_{16}	64 (40_{16})	1201.92
2400	f_{16}	31 ($1F_{16}$)	2400.00			
4800	f_2	127 ($7F_{16}$)	4800.00	f_2	129 (81_{16})	4807.69
9600	f_2	63 ($3F_{16}$)	9600.00	f_2	64 (40_{16})	9615.38
14400	f_2	42 ($2A_{16}$)	14288.37	f_2	42 ($2A_{16}$)	14534.88
19200	f_2	31 ($1F_{16}$)	19200.00			
31250				f_2	19 (13_{16})	31250.00
38400	f_2	15 ($0F_{16}$)	38400.00			

Note: This applies when the peripheral device's clock select bits 1, 0 (bits 7, 6 at address BC_{16}) = "00_2."

Table 11.4.4 Setting examples of transfer rate (2)

Transfer rate (bps)	f_{sys} = 15.9744 MHz			f_{sys} = 16 MHz		
	BRGi's count source	BRGi's set value: n **(Note)**	Actual time (bps)	BRGi's count source	BRGi's set value: n **(Note)**	Actual time (bps)
300	f_{64}	51 (33_{16})	300.00	f_{64}	51 (33_{16})	300.48
600	f_{16}	103 (67_{16})	600.00	f_{16}	103 (67_{16})	600.96
1200	f_{16}	51 (33_{16})	1200.00	f_{16}	51 (33_{16})	1201.92
2400	f_2	207 (CF_{16})	2400.00	f_2	207 (CF_{16})	2403.85
4800	f_2	103 (67_{16})	4800.00	f_2	103 (67_{16})	4807.69
9600	f_2	51 (33_{16})	9600.00	f_2	51 (33_{16})	9615.38
14400	f_2	34 (22_{16})	14262.86			
19200	f_2	25 (19_{16})	19200.00	f_2	25 (19_{16})	19230.77
31250	f_2	15 ($0F_{16}$)	31200.00	f_2	15 ($0F_{16}$)	31250.00
38400	f_2	12 ($0C_{16}$)	38400.00	f_2	12 ($0C_{16}$)	38461.51

Note: This applies when the peripheral device's clock select bits 1, 0 (bits 7, 6 at address BC_{16}) = "00_2."

■ **Error-permitted range of transfer baud**

During reception, the receive data input to the RxD_i pin is taken at the rising edge of the transfer clock. (Refer to section **"11.4.6 Receive operation."**) Accordingly, in order to receive data correctly, the stop bit must be input when the transfer clock of one-set receive data rises last. Figure 11.4.1 shows the relationship between the transfer clock and receive data.

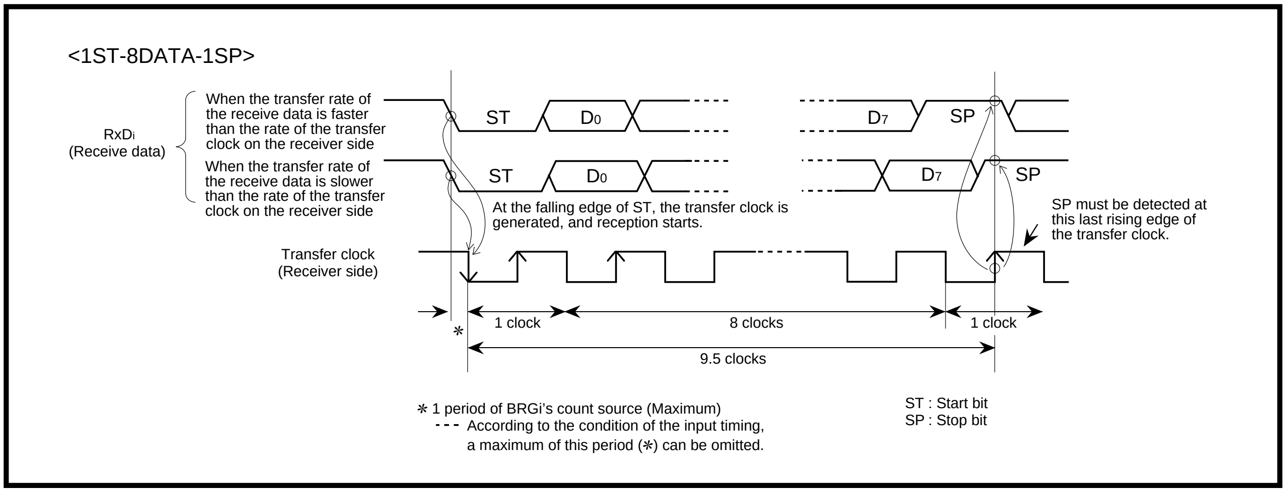


Fig. 11.4.1 Relationship between transfer clock and receive data

Accordingly, the transfer rate of the receiver and transmitter sides must satisfy the following formula in order to receive data correctly.

$$\left(\frac{1}{Bt} \times (b-1) + \frac{1}{F}\right) < \left(\frac{1}{Br} \times (b-0.5) + \frac{1}{F}\right) < \left(\frac{1}{Bt} \times b\right)$$

Br: Transfer rate on receiver side (bps)
Bt: Transfer rate on transmitter side (bps)
F : BRGi's count source frequency on receiver side (Hz)
b : Entire bit number of one-set data
(ex: 12 bits in the case of 1ST-8DATA-1PAR-2SP; See Figure 11.4.2.)

Be sure to satisfy the above formula, and set the timing with enough margin. Also, the user shall make sufficient evaluation before actually using it.

11.4.2 Transfer data format

The transfer data format can be selected from formats shown in Figure 11.4.2. Bits 4 to 6 at addresses 30_{16}, 38_{16} and $B0_{16}$ select the transfer data format. (See Figure 11.2.2.) Set the same transfer data format for both transmitter and receiver sides.
Figure 11.4.3 shows an example of transfer data format. Table 11.4.5 lists each bit in transmit data.

Transfer data length of 7 bits
- 1ST—7DATA—1SP
- 1ST—7DATA—2SP
- 1ST—7DATA—1PAR—1SP
- 1ST—7DATA—1PAR—2SP

Transfer data length of 8 bits
- 1ST—8DATA—1SP
- 1ST—8DATA—2SP
- 1ST—8DATA—1PAR—1SP
- 1ST—8DATA—1PAR—2SP

Transfer data length of 9 bits
- 1ST—9DATA—1SP
- 1ST—9DATA—2SP
- 1ST—9DATA—1PAR—1SP
- 1ST—9DATA—1PAR—2SP

ST : Start bit
DATA : Character bit (Transfer data)
PAR : Parity bit
SP : Stop bit

Fig. 11.4.2 Transfer data format

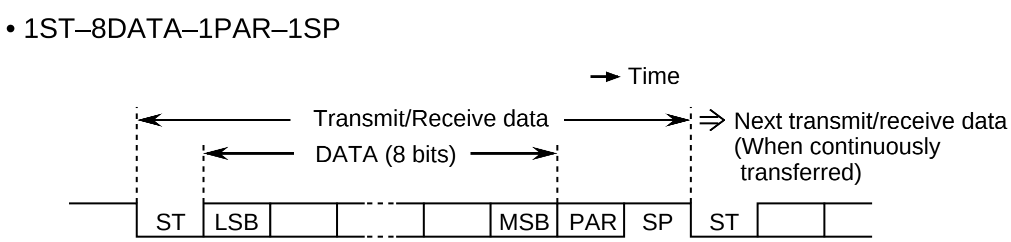


Fig. 11.4.3 Example of transfer data format

Table 11.4.5 Each bit in transmit data

Name	Functions
ST Start bit	“L” signal equivalent to 1 character bit. This is added immediately before the character bits. It indicates start of data transmission.
DATA Character bit	Transmit data which is set in the UARTi transmit buffer register.
PAR Parity bit	A signal that is added immediately after the character bits in order to improve data reliability. The level of this signal changes according to selection of odd/even parity in such a way that the sum of “1”s in the sum of this bit and character bits is always an odd or even number.
SP Stop bit	“H” level signal equivalent to 1 or 2 character bits. This is added immediately after the character bits (or parity bit when parity is enabled). It indicates completion of data transmission.

11.4.3 Method of transmission

Figure 11.4.4 shows an initial setting example for relevant registers when transmitting.

The difference depending on the transfer data length (7 bits, 8 bits, or 9 bits) is the transmit data's length only. When selecting a 7- or 8-bit data length, be sure to set the transmit data into the low-order byte of the UARTi transmit buffer register. When selecting a 9-bit data length, be sure to set the transmit data into the low-order byte and bit 0 of the high-order byte.

Transmission is started when all of the following conditions (① to ③) are satisfied:

① Transmit data is present in the UARTi transmit buffer register (transmit buffer empty flag = "0").
② Transmit is enabled (transmit enable bit = "1").
③ The $\overline{CTS_i}$ pin's input level is "L" (when the $\overline{CTS}$ function selected).
Note: When the $\overline{CTS}$ function is not selected, condition ③ is ignored.

By connecting the $\overline{RTS_i}$ pin (receiver side) and $\overline{CTS_i}$ pin (transmitter side), the timing of transmission and that of reception can be matched. For details, refer to section **"11.4.6 Receive operation."**

When using interrupts, it is necessary to set the relevant registers to enable interrupts. For details, refer to **"CHAPTER 6. INTERRUPTS."**

Figure 11.4.5 shows writing data after transmission is started, and Figure 11.4.6 shows detection of transmit completion.

11.4 Clock asynchronous serial I/O (UART) mode

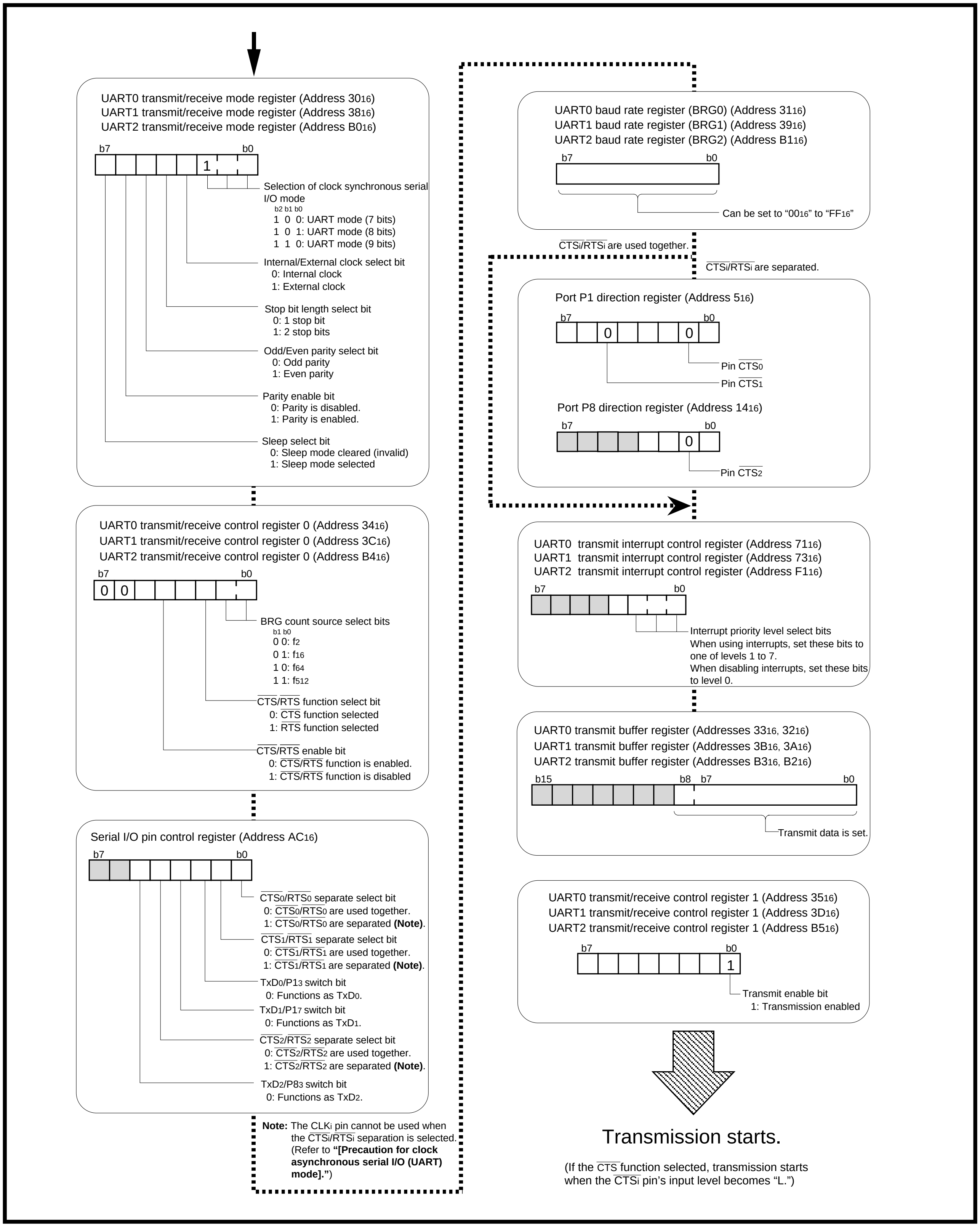


Fig. 11.4.4 Initial setting example for relevant registers when transmitting

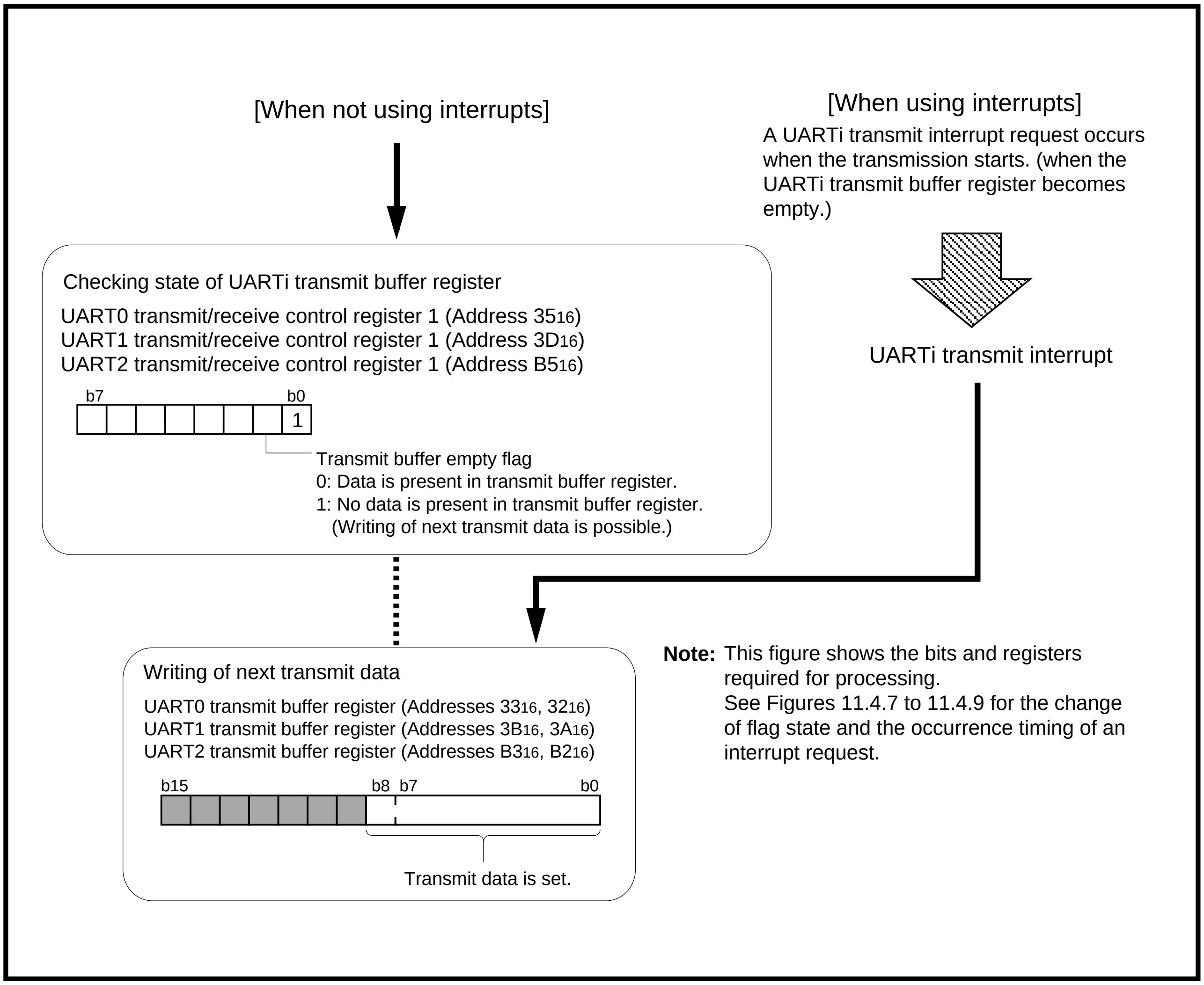


Fig. 11.4.5 Write operation of data after transmission start

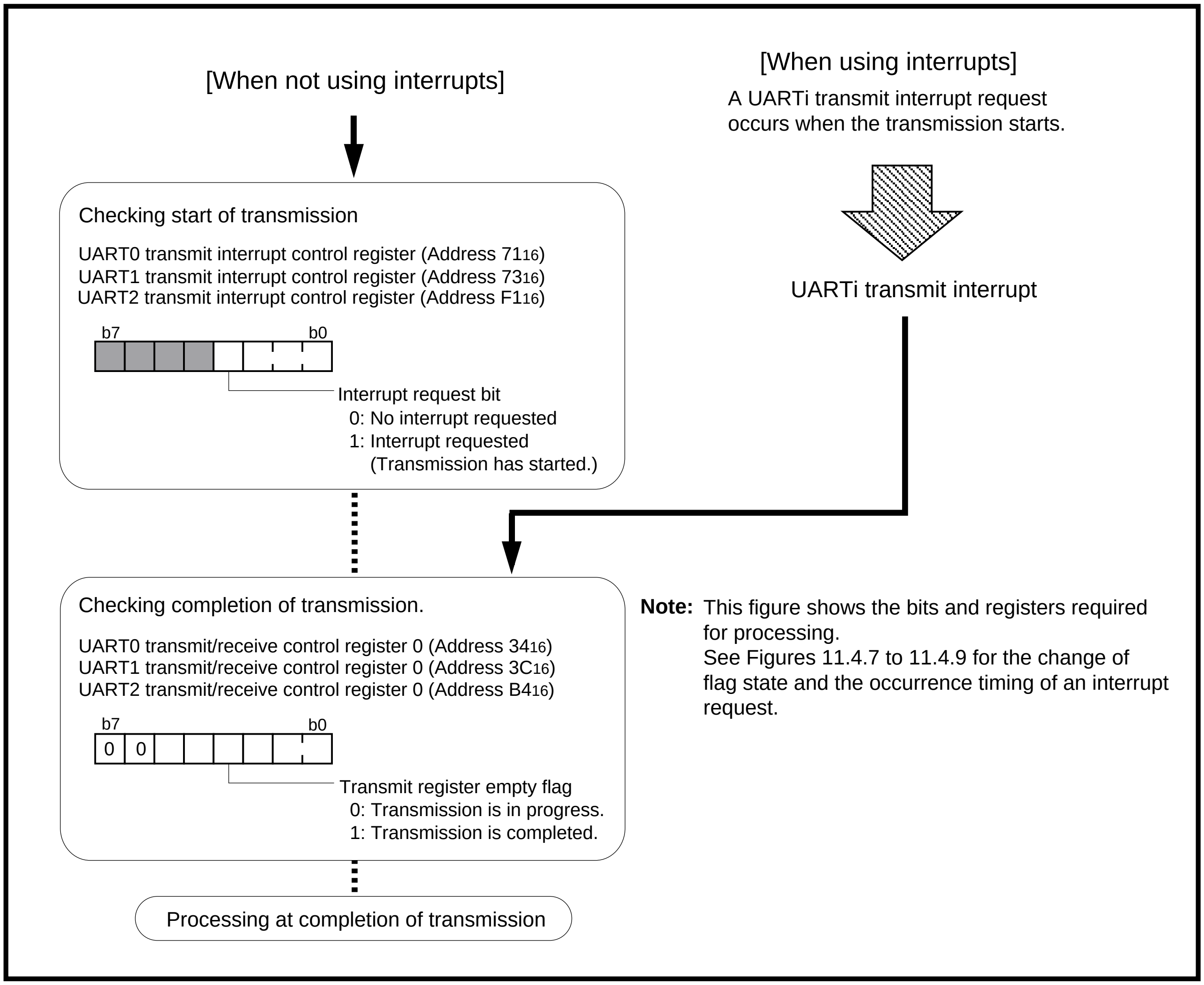


Fig. 11.4.6 Detect operation of transmit completion

11.4.4 Transmit operation

When the receive conditions described in section **"11.4.3 Method of transmission"** have been satisfied, a transfer clock is generated, and the following operations are automatically performed after 1 cycle of the transfer clock or less has passed.

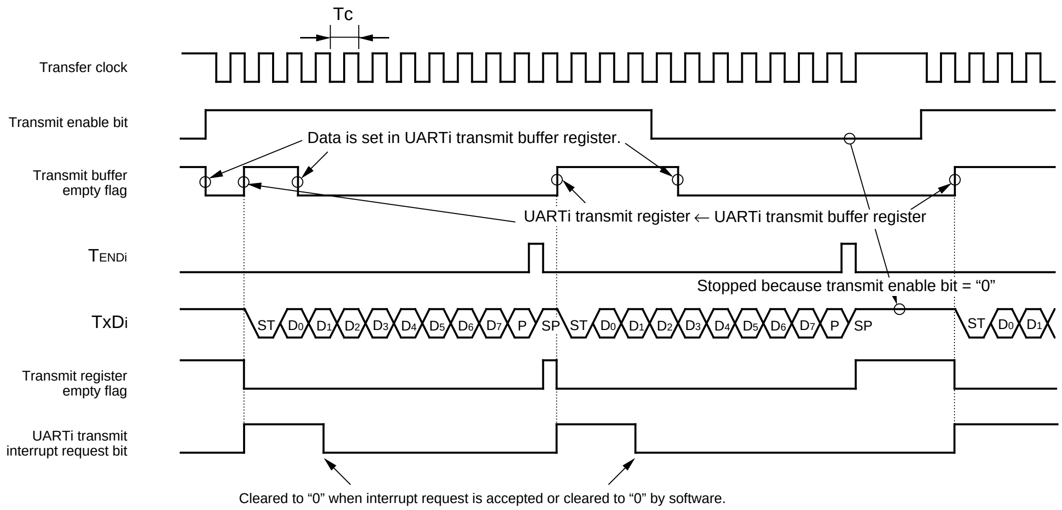
•The UARTi transmit buffer register's contents are transferred to the UARTi transmit register.
•The transmit buffer empty flag is set to "1."
•The transmit register empty flag is cleared to "0."
•A UARTi transmit interrupt request occurs, and the interrupt request bit is set to "1."

The transmit operations are described below:

① Data in the UARTi transmit register is transmitted from the TxD_i pin.
② This data is transmitted bit by bit sequentially in order of ST→DATA (LSB)→•••→DATA (MSB)→PAR →SP according to the transfer data format.
③ The transmit register empty flag is set to "1" at the center of the stop bit (or the second stop bit if 2 stop bits selected). This indicates completion of transmission. Additionally, whether the transmit conditions for the next data are satisfied or not is examined.

When the transmit conditions for the next data are satisfied in step ③, the start bit is generated following the stop bit, and the next data is transmitted. When performing transmission continuously, be sure to set the next transmit data in the UARTi transmit buffer register during transmission (i.e. when the transmit register empty flag = "0"). When the transmit conditions for the next data are not satisfied, the TxD_i pin outputs "H" level and the transfer clock stops.
Figures 11.4.7 and 11.4.8 show examples of transmit timing when the transfer data length = 8 bits, and Figure 11.4.9 shows an example of transmit timing when the transfer data length = 9 bits.

11.4 Clock asynchronous serial I/O (UART) mode



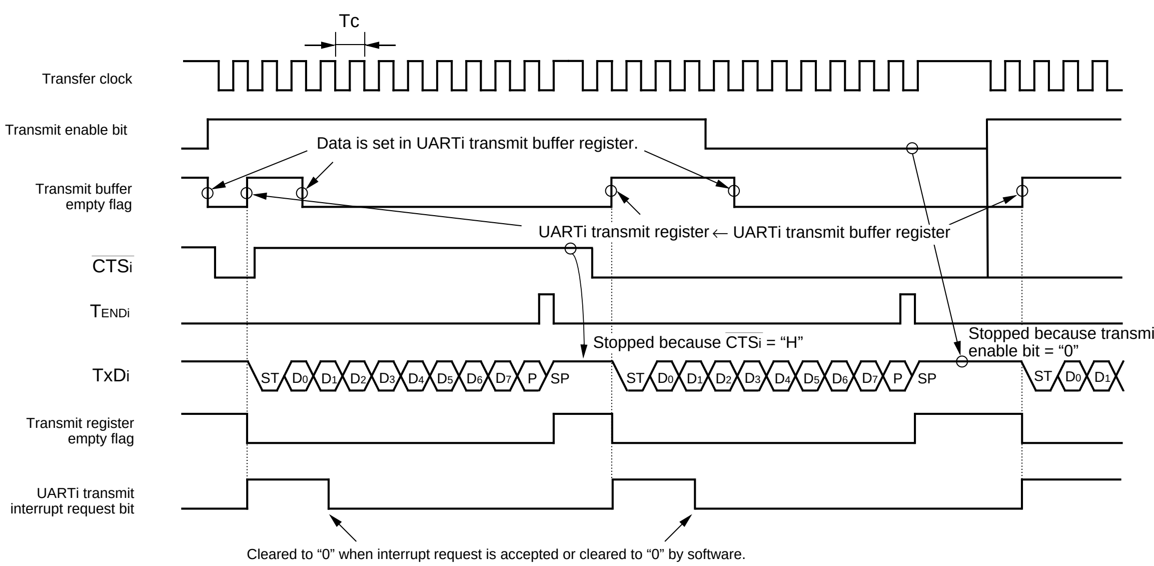
The above timing diagram applies when the following conditions are satisfied:
- Parity enabled
- 1 stop bit
- CTS function not selected

TENDi: Next transmit conditions are examined when this signal level becomes "H." (TENDi is an internal signal. Accordingly, it cannot be read from the external.)

Tc: 16 (n + 1)/fi or 16 (n + 1)/fEXT
fi: BRGi's count source frequency (internal clock)
fEXT: BRGi's count source frequency (external clock)
n: Value set in BRGi

ST: Start bit
D0 to D7: Transfer data
P: Parity bit
ST: Stop bit

Fig. 11.4.7 Example of transmit timing when transfer data length = 8 bits (when parity enabled, 1 stop bit selected, $\overline{\text{CTS}}$ function not selected)



The above timing diagram applies when the following conditions are satisfied:
- Parity enabled
- 1 stop bit
- $\overline{\text{CTS}}$ function selected

TENDi: Next transmit conditions are examined when this signal level becomes "H." (TENDi is an internal signal. Accordingly, it cannot be read from the external.)

Tc = 16 (n + 1)/fi or 16 (n + 1)/fEXT
fi: BRGi's count source frequency (internal clock)
fEXT: BRGi's count source frequency (external clock)
n: Value set in BRGi

ST: Start bit
D0 to D7: Transfer data
P: Parity bit
ST: Stop bit

Fig. 11.4.8 Example of transmit timing when transfer data length = 8 bits (when parity enabled, 1 stop bit and selecting $\overline{\text{CTS}}$ function selected)

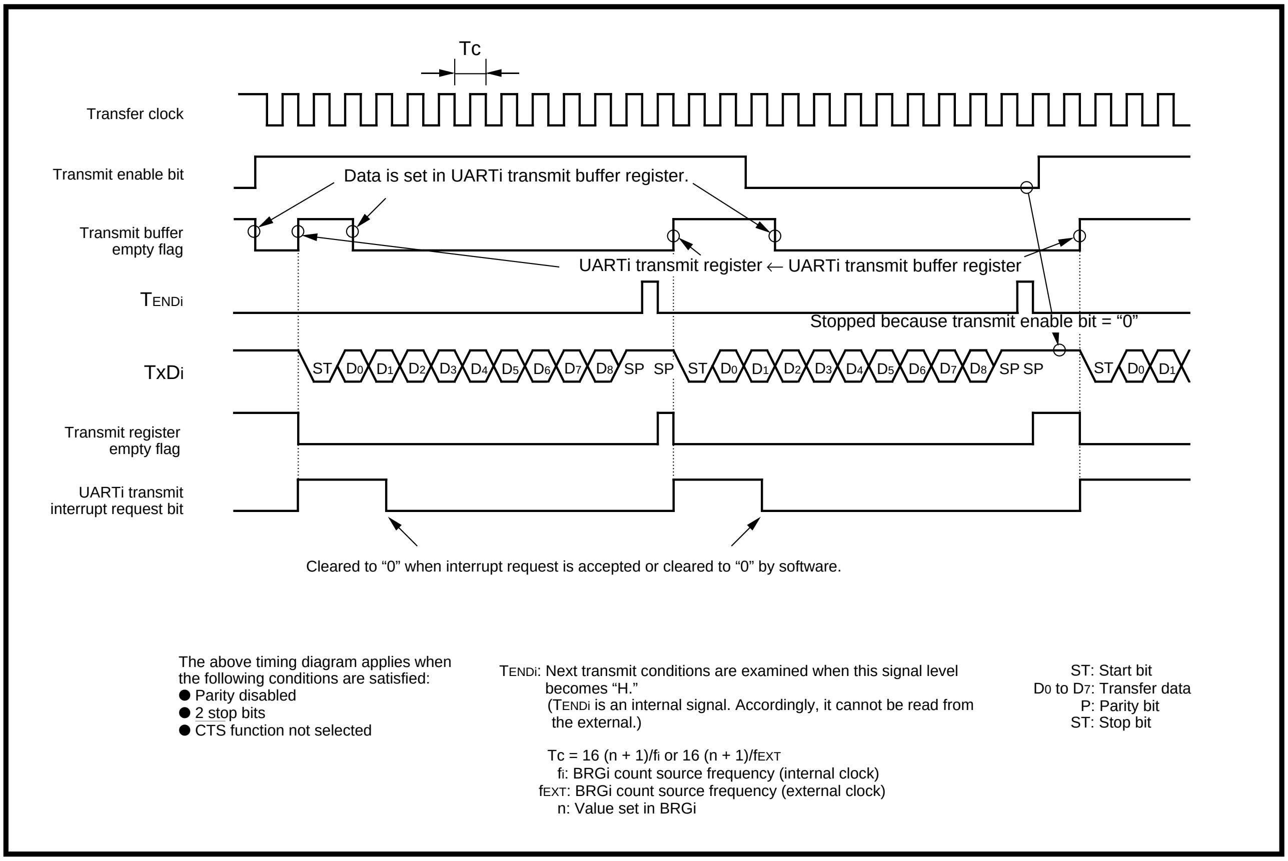


Fig. 11.4.9 Example of transmit timing when transfer data length = 9 bits (when parity disabled, 2 stop bits selected, $\overline{CTS}$ function not selected)

11.4.5 Method of reception

Figure 11.4.10 shows an initial setting example for relevant registers when receiving. Reception is started when all of the following conditions (① and ②) have been satisfied:

① Reception is enabled (receive enable bit = “1”).
② The start bit (its falling edge) is detected.

By connecting the $\overline{RTS_i}$ pin (receiver side) and $\overline{CTS_i}$ pin (transmitter side), the timing of transmission and that of reception can be matched. For details, refer to section **“11.4.6 Receive operation.”**
When using interrupts, it is necessary to set the relevant registers to enable interrupts. For details, refer to **“CHAPTER 6. INTERRUPTS.”**
Figure 11.4.11 shows processing after reception is completed.

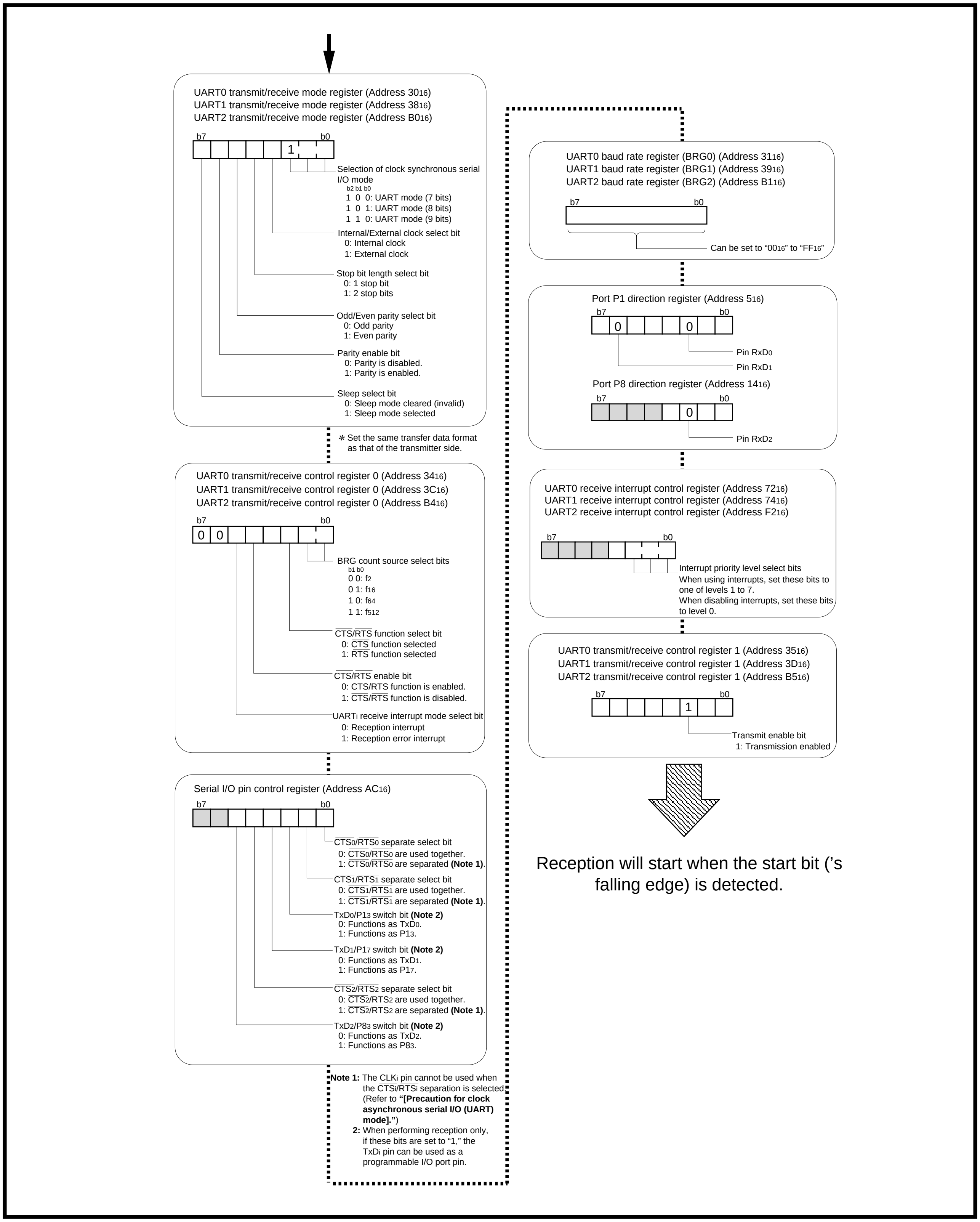
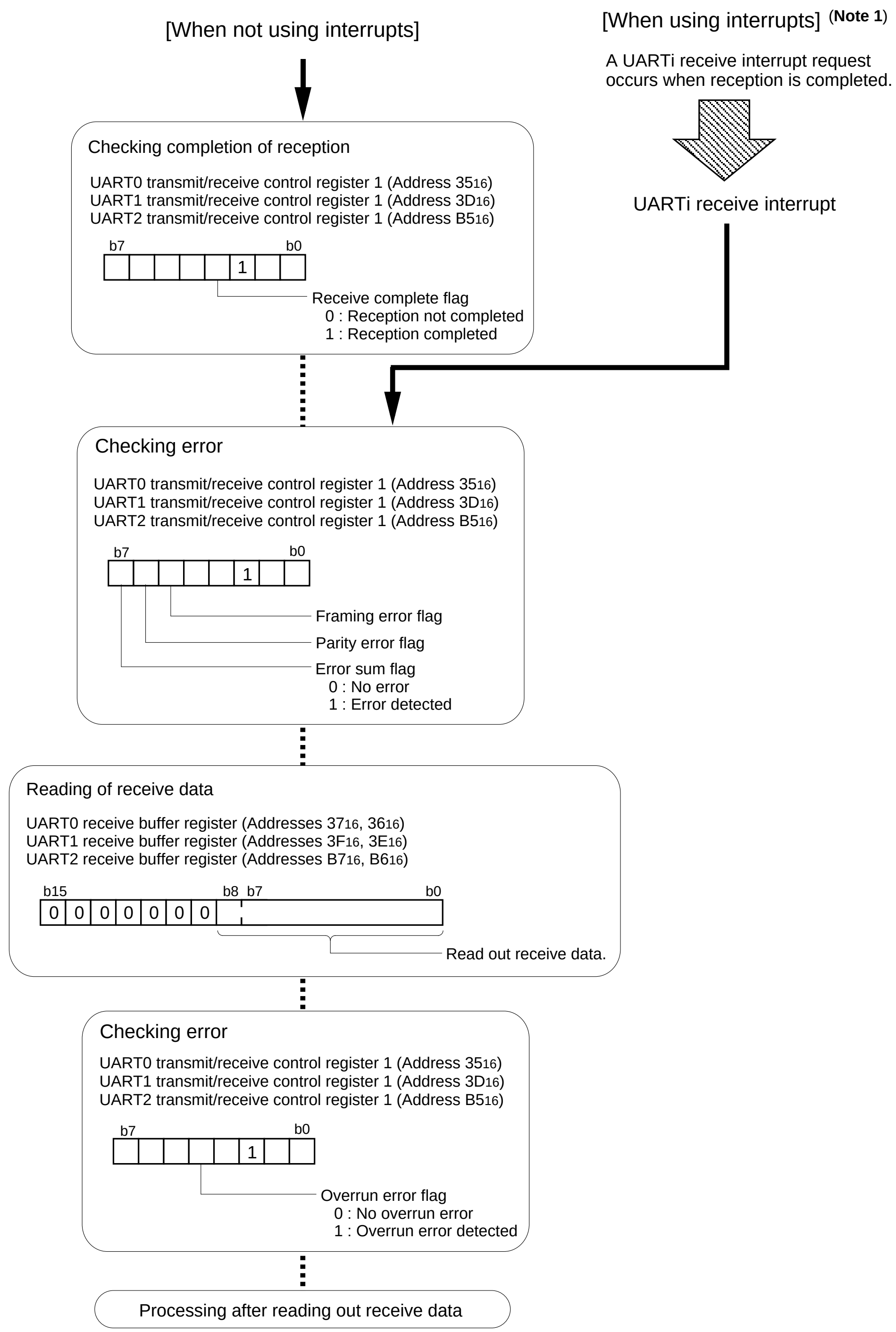


Fig. 11.4.10 Initial setting example for relevant registers when receiving



Notes 1: When performing the processing after the reception is completed, using an interrupt, be sure to select the receive interrupt (UARTi receive interrupt mode select bit = "0").
2: This figure shows the bits and registers required for the processing. See Figure 11.4.13 for the change of flag state and the occurrence timing of an interrupt request.

Fig. 11.4.11 Processing after reception is completed

11.4.6 Receive operation

When the receive enable bit is set to "1," the UARTi enters the receive-enabled state. Then, reception will start when ST ('s falling edge) is detected and a transfer clock is generated.
If the $\overline{RTS}$ function selected, when connecting the $\overline{RTS_i}$ pin to the $\overline{CTS_i}$ pin of the transmitter side, the timing of transmission and that of reception can be matched. If the $\overline{RTS}$ function selected, the $\overline{RTS_i}$ pin's output level becomes as described below.

When the receive enable bit = "0," if one of the following is performed, the $\overline{RTS_i}$ pin's output level becomes "L" and informs of the transmitter side that reception has become enabled:
• The receive enable bit is set to "1."
• The low-order byte of the UARTi receive buffer register is read out.
When the receive enable bit = "1," if the low-order byte of the UARTi receive buffer register is read out, the $\overline{RTS_i}$ pin's output level becomes "L."
Accordingly, when performing reception continuously, an overrun occurrence can be avoided because the $\overline{RTS}$ output level does not become "L" until the receive data is read out.
When reception has started, the $\overline{RTS_i}$ pin's output level becomes "H."
Figure 11.4.12 shows a connection example.

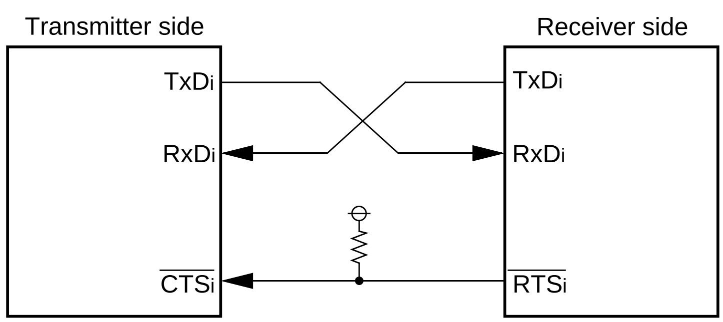
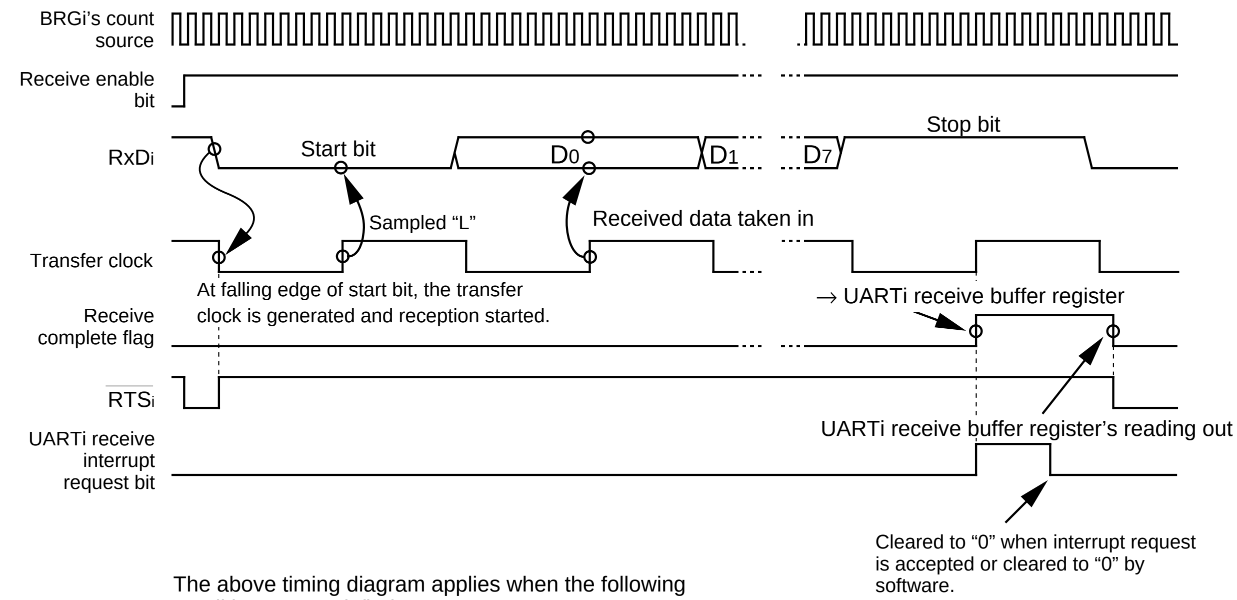


Fig. 11.4.12 Connection example

The receive operation is described below.

① The signal input to the RxD_i pin is taken into the most significant bit of the UARTi receive register, synchronously with the transfer clock's rising edge.
② The contents of the UARTi receive register are shifted, bit by bit, to the right.
③ Steps ① and ② are repeated at each rising edge of the transfer clock.
④ When one set of data has been prepared, in other words, when the shift operation has been performed several times according to the selected data format, the UARTi receive register's contents are transferred to the UARTi receive buffer register.
⑤ Simultaneously with step ④, the receive complete flag is set to "1." Additionally, when the receive interrupt is selected (UARTi receive interrupt mode select bit = "0"), a UARTi receive interrupt request occurs and its interrupt request bit is set to "1."

The receive complete flag is cleared to "0" when the low-order byte of the UARTi receive buffer register has been read out. Figure 11.4.13 shows an example of receive timing when the transfer data length = 8 bits.



The above timing diagram applies when the following conditions are satisfied:
- Parity disabled
- 1 stop bit
- $\overline{RTS}$ function selected

Fig. 11.4.13 Example of receive timing when transfer data length = 8 bits (when parity disabled, 1 stop bit and $\overline{RTS}$ function selected)

11.4.7 Processing on detecting error

In the UART mode, 3 types of errors can be detected. Each error can be detected when the data in the UARTi receive register is transferred to the UARTi receive buffer register, and the corresponding error flag is set to “1.” When any error occurs, the error sum flag is set to “1.” Accordingly, presence of errors can be judged by using the error sum flag.
Table 11.4.6 lists the conditions for setting each error flag to “1” and method to clear it to “0.”
Additionally, when the receive error interrupt is selected (UARTi receive interrupt mode select bit = “1”), the UARTi receive interrupt request bit is set to “1” only when each error has occurred. When the receive interrupt is selected (UARTi receive interrupt mode select bit = “0”), the UARTi receive interrupt request bit is set to “1” when reception has been completed or when a framing or parity error has occurred. (Even when an overrun error has occurred, this bit does not change).

Table 11.4.6 Conditions for setting each error flag to “1” and method to clear it to “0”

Error flag	Conditions for setting	Method to clear
Overrun error flag	When the next data is prepared in the UARTi receive register with the receive complete flag = “1” (i.e. data is present in the UARTi receive buffer register). In other words, when the next data is prepared before the contents of the UARTi receive buffer register are read out **(Note).**	• Clear the receive enable bit to “0.”
Framing error flag	When the number of detected stop bits does not match the set number of stop bits.	• Clear the receive enable bit to “0.” • Read out the low-order byte of the UARTi receive buffer register.
Parity error flag	When the sum of “1”s in the sum of the parity bit and character bits does not match the set number of “1”s.	• Clear the receive enable bit to “0.” • Read out the low-order byte of the UARTi receive buffer register.
Error sum flag	When any error listed above has occurred.	• Clear the all error flags, which are overrun, framing and parity error flags.

Note: The next data is written into the UARTi receive buffer register.

When an error occurs during reception, be sure to initialize the error flag and the UARTi receive buffer register, and then perform reception again. When it is necessary to perform retransmission owing to an error which has occurred on the receiver side during transmission, be sure to set the UARTi transmit buffer register again, and then perform the retransmission.
The method to initialize the UARTi receive buffer register and that to set the UARTi transmit buffer register again are described below.

(1) Method to initialize UARTi receive buffer register

① Clear the receive enable bit to “0” (reception disabled).
② Set the receive enable bit to “1” again (reception enabled).

(2) Method to set UARTi transmit buffer register again

① Clear the serial I/O mode select bits to “000_2” (serial I/O invalid).
② Set the serial I/O mode select bits again.
③ Set the transmit enable bit to “1” (transmission enabled), and set the transmit data to the UARTi transmit buffer register.

11.4.8 Sleep mode

This mode is used to transfer data between the specified microcomputers, which are connected by using UARTi. The sleep mode is selected by setting the sleep select bit (bit 7 at addresses 30_{16}, 38_{16} and $B0_{16}$) to “1” when receiving.

In the sleep mode, receive operation is performed when the MSB (D_8 when the transfer data = 9-bit length, D_7 when it is 8-bit length, D_6 when it is 7-bit length) of the receive data is “1.” Receive operation is not performed when the MSB is “0.” (The UARTi receive register’s contents are not transferred to the UARTi receive buffer register. Additionally, the receive complete flag and each error flag do not change, and no UARTi receive interrupt request occurs.)

The following shows an usage example of the sleep mode when the transfer data = 8-bit length.

① Be sure to set the same transfer data format for the master and slave microcomputers. Additionally, be sure to select the sleep mode for the slave microcomputers.

② Then, transmit the data, of which structure is as follows, from the master microcomputer:
- Bit 7 = “1”
- Bits 6 to 0 indicate the address of the slave microcomputer to be communicated

③ Each slave microcomputer receives the data described in step ②. (At this time, a UARTi receive interrupt request occurs.)

④ Be sure to check for each slave microcomputer, in the interrupt routine, whether bits 6 to 0 of the receive data match its own address.

⑤ For the slave microcomputer of which address matches bits 6 to 0 of the receive data, terminate the sleep mode. (Do not terminate the sleep mode for the other slave microcomputers.)
By performing steps ② to ⑤, “the microcomputer which performs transfer” is specified.

⑥ Transmit the data of which bit 7 = “0” from the master microcomputer. (Only one slave microcomputer specified in steps ② to ⑤ can receive this data. The other microcomputers do not receive this data.)

⑦ By repeating step ⑥, continuous transfer can be performed between two specific microcomputers. When communicating with another slave microcomputer, perform steps ② to ⑤ in order to specify the new slave microcomputer.

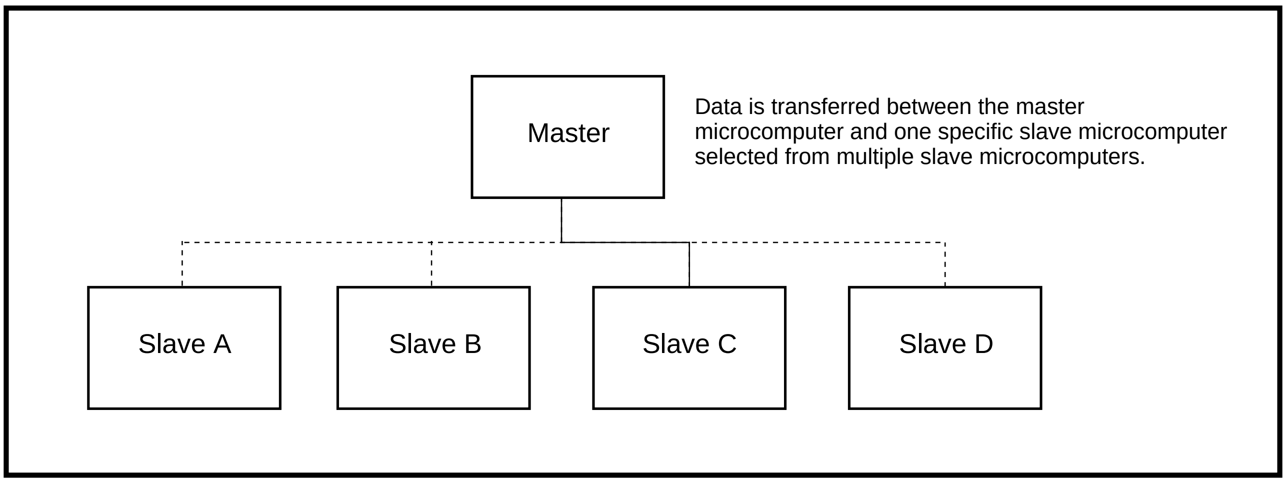


Fig. 11.4.14 Sleep mode

[Precautions for clock asynchronous serial I/O (UART) mode]

1. When using pin $\overline{CTS_2}/\overline{RTS_2}$, be sure that the D-A$_1$ output enable bit (bit 1 at address 96_{16}) = "0" (output disabled).

2. When separating $\overline{CTS_i}/\overline{RTS_i}$, the CLK_i pin cannot be used. Accordingly, when separating $\overline{CTS_i}/\overline{RTS_i}$ in UART mode, be sure to select an internal clock.

3. Writing to the UARTi baud rate register (BRGi) must be performed while transmission/reception halts.

4. When transmitting, be sure to clear the $TxD_0/P1_3$, $TxD_1/P1_7$, or $TxD_2/P8_3$ switch bit (bit 2, 3, or 5 at address AC_{16}) to "0."

MEMORANDUM

CHAPTER 12

A-D CONVERTER

12.1 Overview

The A-D conversion is performed in the 8-bit resolution mode or the 10-bit resolution mode. Also, the input voltage can be compared with the set value by using the A-D converter (in other words, the comparator function). Whether to perform the A-D conversion or comparison can be selected for each pin.

❋ In chapter 12, the operations common to the A-D converter's functions (8-bit resolution, 10-bit resolution, comparator) are simply referred to as "operation."

Table 12.1.1 lists the performance specifications of the A-D converter.

Table 12.1.1 Performance specifications of A-D converter

Item		Performance specifications	
A-D conversion method		Successive approximation conversion method	
Resolution		Either of 8-bit or 10-bit resolution can be selected by software.	
Absolute accuracy		8-bit resolution mode	: ±2 LSB
		10-bit resolution mode	: ±3 LSB
Analog input pin **(Note)**		12 pins (AN_0 to AN_{11})	
Conversion rate per analog input pin		8-bit resolution mode	: 49 ϕ_{AD} cycles
		10-bit resolution mode	: 59 ϕ_{AD} cycles
Comparator function	Comparison operation	Comparison between the set value and analog input voltage	
	Comparison rate per analog input pin	14 ϕ_{AD} cycles	

ϕ_{AD} : A-D converter's operation clock

Note: For each of analog input pin AN_i (i = 0 to 11), whether to use pin AN_i as an input pin of the A-D converter or as that of the comparator can be selected by using the comparator function select register 0 (address DC_{16}) or the comparator function select register 1 (address DD_{16}).

(1) 8-bit resolution mode

The input voltage from pin AN_i (i = 0 to 11) is A-D converted, and the 8-bit A-D conversion result is stored in A-D register i. (Refer to sections **"12.3 A-D conversion method"** and **"12.5 Comparison voltage in 8-bit resolution mode."**)

(2) 10-bit resolution mode

The input voltage from pin AN_i is A-D converted, and the 10-bit A-D conversion result is stored in A-D register i. (Refer to section **"12.3 A-D conversion method."**)

(3) Comparator function

The 8-bit value which has been set in A-D register i is compared with the voltage input from pin AN_i; and then, the result of comparison is stored into the AN_i pin comparator result bit. (Refer to section **"12.6 Comparator function."**)

(4) Operation modes

The A-D converter is equipped with the following 4 modes. The A-D conversion and comparison (in other words, the comparator function) are performed in the same operation modes.
The operation mode depends on the analog input pin.

■ **One-shot mode**

This mode is used to perform the operation once for a voltage input from one selected analog input pin. This mode can be used with analog input pin AN_i (i = 0 to 11).

■ **Repeat mode**

This mode is used to perform the operation repeatedly for a voltage input from one selected analog input pin. This mode can be used with analog input pin AN_i (i = 0 to 11).

■ **Single sweep mode**

This mode is used to perform the operation for voltages input from multiple pins selected from analog input pins AN_j (j = 0 to 7), one at a time. This mode can be used with analog input pins AN_j (j = 0 to 7).

■ **Repeat sweep mode 0**

This mode is used to perform the operation repeatedly for voltages input from multiple pins selected from analog input pins AN_j (j = 0 to 7) .

■ **Repeat sweep mode 1**

This mode is used to perform the operation repeatedly for voltages input from analog input pins AN_j (j = 0 to 7). In this mode, analog input pins are divided into two groups: frequently-used pins and non-frequently-used pins.
This mode can be used with analog input pins AN_j (j = 0 to 7).

12.2 Block description

Figure 12.2.1 shows the block diagram of the A-D converter. Registers relevant to the A-D converter are described below.

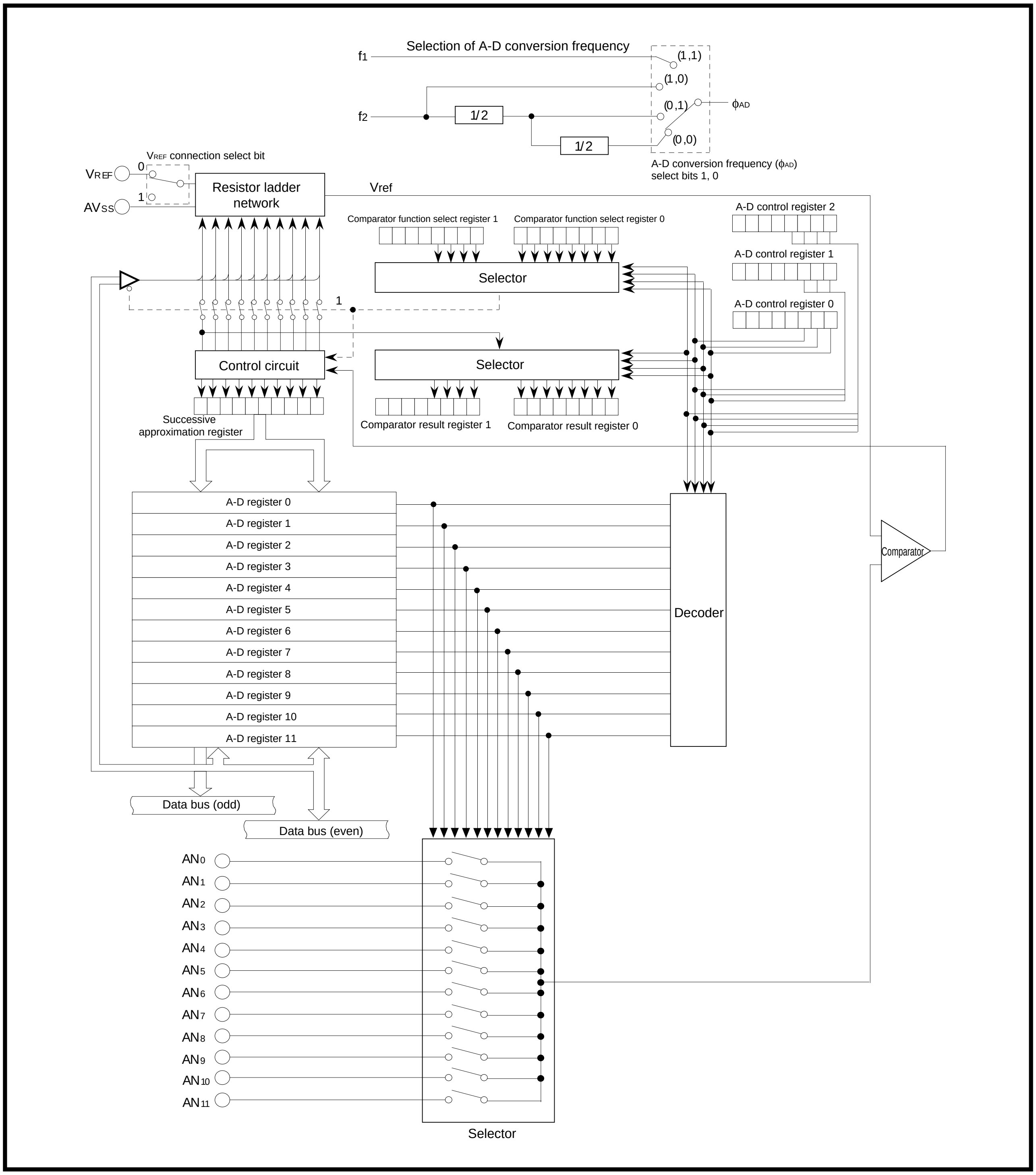


Fig. 12.2.1 Block diagram of A-D converter

12.2.1 A-D control registers 0, 1, and 2

Figures 12.2.2, 12.2.3 and 12.2.4 show the structures of the A-D control registers 0, 1 and 2, respectively.

A-D control register 0 (Address $1E_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Bit	Bit name	Function	At reset	R/W
0	Analog input pin select bits (Valid in the one-shot and repeat modes.) **(Note 1)**	b2 b1 b0 0 0 0 : AN_0 is selected. 0 0 1 : AN_1 is selected. 0 1 0 : AN_2 is selected. 0 1 1 : AN_3 is selected. 1 0 0 : AN_4 is selected. 1 0 1 : AN_5 is selected. 1 1 0 : AN_6 is selected. 1 1 1 : AN_7 is selected. **(Note 2)**	Undefined	RW
1			Undefined	RW
2			Undefined	RW
3	A-D operation mode select bits 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or 1	0	RW
4			0	RW
5	Fix this bit to “0.”		0	RW
6	A-D conversion start bit	0 : A-D conversion halts. 1 : A-D conversion starts.	0	RW **(Note 3)**
7	A-D conversion frequency (ϕ_{AD}) select bit 0	See Table 12.2.1.	0	RW

Notes 1: When using pins AN_0 to AN_7, be sure to fix bit 3 of the analog input pin select bits 1 (bits 3 to 0 at address DB_{16}) to “0.” Setting bit 3 of the analog input pin select bits 1 to “1” invalidates these bits.
Also, these bits are invalid in the single sweep mode, repeat sweep mode 0 and repeat sweet mode 1. (Each may be either “0” or “1.”)

2: When using pin AN_7, be sure that the D-A_0 output enable bit (bit 0 at address 96_{16}) = “0” (output disabled).

3: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

4: Writing to each bit (except writing of “0” to bit 6) of the A-D control register 0 must be performed while the A-D converter halts, regardless of the A-D operation mode.

Fig. 12.2.2 Structure of A-D control register 0

A-D control register 1 (Address $1F_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Bit	Bit name	Function	At reset	R/W
0	A-D sweep pin select bits (Valid in the single sweep mode, repeat sweep mode 0 and repeat sweep mode 1.) **(Note 1)**	Single sweep mode/Repeat sweep mode 0 b1 b0 0 0 : Pins AN_0 and AN_1 (2 pins) 0 1 : Pins AN_0 to AN_3 (4 pins) 1 0 : Pins AN_0 to AN_5 (6 pins) 1 1 : Pins AN_0 to AN_7 (8 pins) **(Note 2)**	1	RW
1		Repeat sweep mode 1 **(Note 3)** b1 b0 0 0 : Pin AN_0 (1 pin) 0 1 : Pins AN_0 and AN_1 (2 pins) 1 0 : Pins AN_0 to AN_2 (3 pins) 1 1 : Pins AN_0 to AN_3 (4 pins)	1	RW
2	A-D operation mode select bit 1 (Used in the repeat sweep mode 0 and repeat sweep mode 1.) **(Note 4)**	0 : Repeat sweep mode 0 1 : Repeat sweep mode 1	0	RW
3	Resolution select bit	0 : 8-bit resolution mode 1 : 10-bit resolution mode	0	RW
4	A-D conversion frequency (ϕ_{AD}) select bit 1	See Table 12.2.1.	0	RW
5	Fix this bit to “0.”		0	RW
6	V_{REF} connection select bit **(Note 5)**	0 : Pin V_{REF} is connected. 1 : Pin V_{REF} is disconnected.	0	RW
7	The value is “0” at reading.		0	–

Notes 1: These bits are invalid in the one-shot and repeat modes. (They may be either “0” or “1.”)
2: When using pin AN_7, be sure that the D-A_0 output enable bit (bit 0 at address 96_{16}) = “0” (output disabled).
3: Be sure to select frequently-used analog input pins in the repeat sweep mode 1.
4: Fix this bit to “0” in the one-shot mode, repeat mode, and single sweep mode.
5: When this bit is cleared from “1” to “0,” be sure to start the A-D conversion after an interval of 1 µs or more has elapsed.
6: Writing to each bit of the A-D control register 1 must be performed while the A-D converter halts, regardless of the A-D operation mode.

Fig. 12.2.3 Structure of A-D control register 1

A-D control register 2 (Address DB_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0				

Bit	Bit name	Function	At reset	R/W
0	Analog input pin select bits 1 (Note 1)	b3 b2 b1 b0 0 X X X : Pins AN_0 to AN_7 are selected. **(Note 2)** 1 0 0 0 : Pin AN_8 is selected. **(Note 3)** 1 0 0 1 : Pin AN_9 is selected. **(Note 4)** 1 0 1 0 : Pin AN_{10} is selected. **(Note 5)** 1 0 1 1 : Pin AN_{11} is selected. **(Note 6)** 1 1 0 0 : Do not select. ⋮ 1 1 1 1 : Do not select.	0	RW
1			0	RW
2			0	RW
3			0	RW
7 to 4	Fix these bits to “0000.”		0	RW

X : They may be either “0” or “1.”

Note 1: When using pins AN_0 to AN_7, regardless of the A-D operation mode, be sure to fix bit 3 to “0.” Also, pins AN_8 to AN_{11} are used only in the one-shot mode and repeat mode.

2: Select pins AN_0 to AN_7 at bits 2 to 0 of A-D control register 0 (address $1E_{16}$).

3: When using pin AN_8, be sure that the D-A_1 output enable bit (bit 1 at address 96_{16}) = “0” (output disabled). Also, be sure not to use pin $\overline{CTS_2}/\overline{RTS_2}$.

4: When using pin AN_9, be sure not to use pin $\overline{CTS_2}/CLK_2$.

5: When using pin AN_{10}, be sure not to use pin R_XD_2.

6: When using pin AN_{11}, be sure not to use pin T_XD_2.

7: Writing to each bit of A-D control register 2 must be performed while the A-D conversion halts, regardless of the A-D operation mode.

Fig. 12.2.4 Structure of A-D control register 2

(1) Analog input pin select bits 0 (bits 0 to 2 at address $1E_{16}$), analog input pin select bits 1 (bits 3 to 0 at address DB_{16})

These bits are used to select an analog input pin. Pins which are not selected as analog input pins serve as programmable I/O port pins or I/O pins of other internal peripheral devices, which are multiplexed.

When using pins AN_0 to AN_7, be sure to fix bit 3 of the analog input pin select bits 1 (bits 3 to 0 at address DB_{16}) to "0," regardless of the A-D operation mode.

Pins AN_8 to AN_{11} can be used only in the one-shot mode and repeat mode.

Also, these bits must be specified again if the user switches the operation mode to the one-shot mode or repeat mode after the operation is performed in the single sweep mode, repeat sweep mode 0, or repeat sweep mode 1.

(2) A-D operation mode select bits 0 (bits 3 and 4 at address $1E_{16}$), A-D operation mode select bit 1 (bit 2 at address $1F_{16}$)

These bits are used to select the operation mode of the A-D converter.

When using the one-shot mode, repeat mode, or single sweep mode, be sure to fix the A-D operation mode select bit 1 to "0".

(3) A-D conversion start bit (bit 6 at address $1E_{16}$)

Setting this bit to "1" generates a trigger, causing the A-D converter to start its operation. Clearing this bit to "0" causes the A-D converter to halt its operation.

In the one-shot mode or single sweep mode, this bit is cleared to "0" when the operation is completed.

In the repeat mode, repeat sweep mode 0, or repeat sweep mode 1, the A-D converter continues its operation until this bit is cleared to "0" by software.

(4) A-D conversion frequency (ϕ_{AD}) select bit 0 (bit 7 at address $1E_{16}$), A-D conversion frequency (ϕ_{AD}) select bit 1 (bit 4 at address $1F_{16}$)

These bits are used to select the operation clock (ϕ_{AD}) of the A-D converter. Table 12.2.1 lists the conversion time per one analog input pin.

Since the A-D converter's comparator consists of capacity coupling amplifiers, be sure to keep that

Table 12.2.1 Conversion time per one analog input pin

A-D conversion frequency (ϕ_{AD}) select bit 1	A-D conversion frequency (ϕ_{AD}) select bit 0	ϕ_{AD}	Conversion time (µs) **(Note)** f_{sys} = 20 MHz 8-bit resolution mode	10-bit resolution mode	Comparator function
0	0	f_2 divided by 4	19.60	23.60	5.60
0	1	f_2 divided by 2	9.80	11.80	2.80
1	0	f_2	4.90	5.90	1.40
1	1	f_1	2.45	Do not select.	0.70

Note: This applies when the peripheral devices' clock select bits 0, 1 (bits 6, 7 at address BC_{16}) = "00_2."
$\phi_{AD} \geq$ 250 kHz while the A-D converter is active.

(5) A-D sweep pin select bits (bits 0 and 1 at address $1F_{16}$)

These bits are used to select analog input pins in the single sweep mode, repeat sweep mode 0, or repeat sweep mode 1. Pins which are not selected as analog input pins serve as programmable I/O port pins or as I/O pins of other internal peripheral devices, which are multiplexed.

(6) Resolution select bit (bit 3 at address $1F_{16}$)

This bit is used to select a resolution.

(7) V_{REF} connection select bit (bit 6 at address $1F_{16}$)

When the A-D converter is not used, this bit is used to disconnect the resistor ladder network of the A-D converter from the reference voltage input pin (V_{REF}).
When the resistor ladder network is disconnected from pin V_{REF}, the current is not flowed from pin V_{REF} to resistor ladder network. Accordingly, the power dissipation can be saved.
When this bit changes from "1" (V_{REF} disconnected) to "0" (V_{REF} connected), start of the operation must be 1 µs or more later.

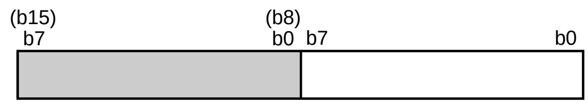
12.2.2 A-D register i (i = 0 to 11)

Figures 12.2.5 and 12.2.6 show the structures of the A-D register i. When the A-D conversion is completed, the conversion result (contents of the successive approximation register) is stored into this register. When the comparator function is selected, the value to be compared is stored in this register.

Each A-D register i corresponds to an analog input pin (AN_i).

■ When 8-bit resolution mode is selected

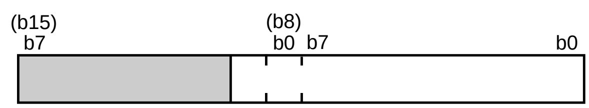
A-D register 0 (Addresses 21_{16}, 20_{16})
A-D register 1 (Addresses 23_{16}, 22_{16})
A-D register 2 (Addresses 25_{16}, 24_{16})
A-D register 3 (Addresses 27_{16}, 26_{16})
A-D register 4 (Addresses 29_{16}, 28_{16})
A-D register 5 (Addresses $2B_{16}$, $2A_{16}$)
A-D register 6 (Addresses $2D_{16}$, $2C_{16}$)
A-D register 7 (Addresses $2F_{16}$, $2E_{16}$)
A-D register 8 (Addresses $E1_{16}$, $E0_{16}$)
A-D register 9 (Addresses $E3_{16}$, $E2_{16}$)
A-D register 10 (Addresses $E5_{16}$, $E4_{16}$)
A-D register 11 (Addresses $E7_{16}$, $E6_{16}$)



Bit	Function	At reset	R/W
7 to 0	Reads an A-D conversion result.	Undefined	RO
15 to 8	The value is “0” at reading.	0	–

■ When 10-bit resolution mode is selected

A-D register 0 (Addresses 21_{16}, 20_{16})
A-D register 1 (Addresses 23_{16}, 22_{16})
A-D register 2 (Addresses 25_{16}, 24_{16})
A-D register 3 (Addresses 27_{16}, 26_{16})
A-D register 4 (Addresses 29_{16}, 28_{16})
A-D register 5 (Addresses $2B_{16}$, $2A_{16}$)
A-D register 6 (Addresses $2D_{16}$, $2C_{16}$)
A-D register 7 (Addresses $2F_{16}$, $2E_{16}$)
A-D register 8 (Addresses $E1_{16}$, $E0_{16}$)
A-D register 9 (Addresses $E3_{16}$, $E2_{16}$)
A-D register 10 (Addresses $E5_{16}$, $E4_{16}$)
A-D register 11 (Addresses $E7_{16}$, $E6_{16}$)

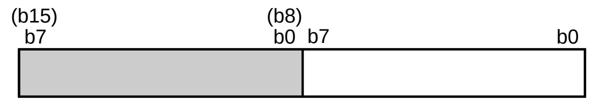


Bit	Function	At reset	R/W
9 to 0	Reads an A-D conversion result.	Undefined	RO
15 to 10	The value is “0” at reading.	0	–

Fig. 12.2.5 Structure of A-D register i (1)

■ When comparator function is selected

A-D register 0 (Addresses 21_{16}, 20_{16})
A-D register 1 (Addresses 23_{16}, 22_{16})
A-D register 2 (Addresses 25_{16}, 24_{16})
A-D register 3 (Addresses 27_{16}, 26_{16})
A-D register 4 (Addresses 29_{16}, 28_{16})
A-D register 5 (Addresses $2B_{16}$, $2A_{16}$)
A-D register 6 (Addresses $2D_{16}$, $2C_{16}$)
A-D register 7 (Addresses $2F_{16}$, $2E_{16}$)
A-D register 8 (Addresses $E1_{16}$, $E0_{16}$)
A-D register 9 (Addresses $E3_{16}$, $E2_{16}$)
A-D register 10 (Addresses $E5_{16}$, $E4_{16}$)
A-D register 11 (Addresses $E7_{16}$, $E6_{16}$)



Bit	Function	At reset	R/W
7 to 0	Any value in the range from "00_{16}" to "FF_{16}" can be set. The set value is compared with the input voltage. The value is undefined at reading.	Undefined	RO
15 to 8	The value is "0" at reading.	0	–

Note: When the comparator function is selected, writing to and reading from A-D register i must be performed while the A-D converter halts.

Fig. 12.2.6 Structure of A-D register i (2)

12.2.3 Comparator function select register 0, 1 and comparator result register 0, 1

Figure 12.2.7 shows the structures of comparator function select register 0 and 1; Figure 12.2.8 shows the structures of comparator result register 0 and 1.

When the AN_i pin comparator function select bit is set to "1," the comparator function is selected. When the A-D conversion is performed, be sure to clear the corresponding bit to "0."

For details of the comparator function, refer to section **"12.6 Comparator function."**

Comparator function select register 0 (Address DC_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	AN_0 pin comparator function select bit	0 : The comparator function is not selected. 1 : The comparator function is selected.	0	RW
1	AN_1 pin comparator function select bit		0	RW
2	AN_2 pin comparator function select bit		0	RW
3	AN_3 pin comparator function select bit		0	RW
4	AN_4 pin comparator function select bit		0	RW
5	AN_5 pin comparator function select bit		0	RW
6	AN_6 pin comparator function select bit		0	RW
7	AN_7 pin comparator function select bit		0	RW

Note: Writing to comparator function select register 0 must be performed while the A-D converter halts.

Comparator function select register 1 (Address DD_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0				

Bit	Bit name	Function	At reset	R/W
0	AN_8 pin comparator function select bit	0 : The comparator function is not selected. 1 : The comparator function is selected.	0	RW
1	AN_9 pin comparator function select bit		0	RW
2	AN_{10} pin comparator function select bit		0	RW
3	AN_{11} pin comparator function select bit		0	RW
7 to 4	Fix these bits to "0000."		0	RW

Note: Writing to comparator function select register 1 must be performed while the A-D converter halts.

Fig. 12.2.7 Structures of comparator function select register 0 and 1

Comparator result register 0 (Address DE_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	AN_0 pin comparator result bit	0 : The set value > The input level at pin AN_i 1 : The set value < The input level at pin AN_i	0	RW
1	AN_1 pin comparator result bit		0	RW
2	AN_2 pin comparator result bit		0	RW
3	AN_3 pin comparator result bit		0	RW
4	AN_4 pin comparator result bit		0	RW
5	AN_5 pin comparator result bit		0	RW
6	AN_6 pin comparator result bit		0	RW
7	AN_7 pin comparator result bit		0	RW

Note: Writing to comparator result register 0 must be performed while the A-D converter halts.

Comparator result register 1 (Address DF_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0				

Bit	Bit name	Function	At reset	R/W
0	AN_8 pin comparator result bit	0 : The set value > The input level at pin AN_i 1 : The set value < The input level at pin AN_i	0	RW
1	AN_9 pin comparator result bit		0	RW
2	AN_{10} pin comparator result bit		0	RW
3	AN_{11} pin comparator result bit		0	RW
7 to 4	Fix these bits to “0000.”		0	RW

Note: Writing to comparator result register 1 must be performed while the A-D converter halts.

Fig. 12.2.8 Structures of comparator result register 0 and 1

12.2.4 A-D conversion interrupt control register

Figure 12.2.9 shows the structure of the A-D conversion interrupt control register. For details about interrupts, refer to **"CHAPTER 6. INTERRUPTS."**

A-D conversion interrupt control register (Address 70_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt requested 1 : Interrupt requested	Undefined **(Note 1)**	RW **(Note 2)**
7 to 4	Nothing is assigned.		Undefined	—

Notes 1: Before using an A-D conversion interrupt, be sure to clear this bit to "0" by software.
2: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

Fig. 12.2.9 Structure of A-D conversion interrupt control register

(1) Interrupt priority level select bits (bits 2 to 0)

These bits are used to select an A-D conversion interrupt's priority level. When using an A-D conversion interrupt, be sure to select one of the priority levels (1 to 7). When an A-D conversion interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL). The requested interrupt is enabled only when its priority level is higher than the IPL. (However, this applies when the interrupt disable flag (I) = "0.")

To disable an A-D conversion interrupt, set these bits to "000_2" (level 0).

(2) Interrupt request bit (bit 3)

This bit is set to "1" when an A-D conversion interrupt request has occurred. This bit is automatically cleared to "0" when the A-D conversion interrupt request has accepted. This bit can be set to "1" or cleared to "0" by software.

12.2.5 Port P7 direction register, port P8 direction register

The A-D converter's input pins are multiplexed with the port P7 and P8 pins. When using these pins as A-D converter's input pins, be sure to clear the port P7, P8 direction registers' bits, corresponding to the A-D converter's input pins, in order to set these pins to the input mode. Figure 12.2.10 shows the correspondence between the port P7, P8 direction registers and the A-D converter's input pins.

Port P7 direction register (Address 11_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	Pin AN_0	0 : Input mode 1 : Output mode When using any of these pins as A-D converter's input pin, be sure to clear its corresponding bit to "0."	0	RW
1	Pin AN_1		0	RW
2	Pin AN_2		0	RW
3	Pin AN_3		0	RW
4	Pin AN_4		0	RW
5	Pin AN_5		0	RW
6	Pin AN_6		0	RW
7	Pin AN_7 (Pin DA_0) **(Note 1)**		0	RW

Notes 1: When using pin AN_7, be sure to clear the D-A_0 output enable bit (bit 0 at address 96_{16}) = "0" (output disabled).
2: The pins in ( ) is I/O pins of other internal peripheral devices, which are multiplexed with the corresponding port P7 pin.

Port P8 direction register (Address 14_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	Pin AN_8 (Pin $\overline{CTS_2}$/$\overline{RTS_2}$/DA_1) **(Note 1)**	0 : Input mode 1 : Output mode When using any of these pins as A-D converter's input pin, be sure to clear its corresponding bit to "0."	0	RW
1	Pin AN_9 (Pin $\overline{CTS_2}$/CLK_2) **(Note 2)**		0	RW
2	Pin AN_{10} (Pin R_XD_2) **(Note 3)**		0	RW
3	Pin AN_{11} (Pin T_XD_2) **(Note 4)**		0	RW
7 to 5	Nothing is assigned.		Undefined	–

Notes 1: When using pin AN_8 be sure to clear the D-A_1 output enable bit (bit 1 at address 96_{16}) = "0" (output disabled). Also, be sure not to use pin $\overline{CTS_2}$/$\overline{RTS_2}$.
2: When using pin AN_9, be sure not to use pin $\overline{CTS_2}$/CLK_2.
3: When using pin AN_{10}, be sure not to use pin R_XD_2.
4: When using pin AN_{11}, be sure not to use pin T_XD_2.
5: The pins in ( ) are I/O pins of other internal peripheral devices, which are multiplexed with the corresponding port P8 pins.

Fig. 12.2.10 Correspondence between port P7, P8 derection registers and A-D converter's input pins

12.3 A-D conversion method

The A-D converter compares the comparison voltage (V_{ref}), which is internally generated according to the contents of the successive approximation register, with the analog input voltage (V_{IN}), which is input from the analog input pin (AN_i). By reflecting the comparison result on the successive approximation register, V_{IN} is converted into a digital value. When a trigger is generated, the A-D converter performs the following processing:

① Determining bit 9 of the successive approximation register
The A-D converter compares V_{ref} with V_{IN}. At this time, the contents of the successive approximation register is "1000000000_2" (initial value).
Bit 9 of the successive approximation register depends on the comparison result as follows:
When $V_{ref} < V_{IN}$, bit 9 = "1"
When $V_{ref} > V_{IN}$, bit 9 = "0"

② Determining bit 8 of the successive approximation register
After setting bit 8 of the successive approximation register to "1," the A-D converter compares V_{ref} with V_{IN}. Bit 8 depends on the comparison result as follows:
When $V_{ref} < V_{IN}$, bit 8 = "1"
When $V_{ref} > V_{IN}$, bit 8 = "0"

③ Determining bits 7 to LSB of the successive approximation register
Operation ② is performed for each of bits 7 to 0 in the 10-bit resolution mode.
Operation ② is performed for each of bits 7 to 2 in the 8-bit resolution mode.
When the LSB is determined, the contents of the successive approximation register (in order words, conversion result) are transferred to the A-D register i.

V_{ref} is generated according to the latest contents of the successive approximation register. Table 12.3.1 lists the relationship between the successive approximation register's contents and V_{ref}. Tables 12.3.2 and 12.3.3 list the changes of the successive approximation register and V_{ref} during the A-D conversion, respectively. Figure 12.3.1 shows the ideal A-D conversion characteristics in the 10-bit resolution mode.

Table 12.3.1 Relationship between successive approximation register's contents and V_{ref}

Successive approximation register's contents: n	V_{ref} (V)
0	0
1 to 1023	$\frac{V_{REF}}{1024} \times (n - 0.5)$

V_{REF}: Reference voltage

Table 12.3.2 Change of successive approximation register and V_{ref} during A-D conversion (8-bit resolution)

	Successive approximation register (b9 … b0)	Change of V_{ref}
A-D converter halt	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$ [V]
1st comparison	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{2048}$ [V]
↓ 2nd comparison	n9 1 0 0 0 0 0 0 0 0 (n9: 1st comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{2048}$ [V] (•n9 = 1: $+\frac{V_{REF}}{4}$; •n9 = 0: $-\frac{V_{REF}}{4}$)
↓ 3rd comparison	n9 n8 1 0 0 0 0 0 0 0 (n8: 2nd comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{2048}$ [V] (•n8 = 1: $+\frac{V_{REF}}{8}$; •n8 = 0: $-\frac{V_{REF}}{8}$)
⋮	⋮	⋮
↓ 8th comparison	n9 n8 n7 n6 n5 n4 n3 1 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm ...... \pm \frac{V_{REF}}{256} - \frac{V_{REF}}{2048}$ [V]
↓ Conversion completed	n9 n8 n7 n6 n5 n4 n3 n2 0 0	

Table 12.3.3 Change of successive approximation register and V_{ref} during A-D conversion (10-bit resolution)

	Successive approximation register (b9 … b0)	Change of V_{ref}
A-D converter halt	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$ [V]
1st comparison	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{2048}$ [V]
↓ 2nd comparison	n9 1 0 0 0 0 0 0 0 0 (n9: 1st comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{2048}$ [V] (•n9 = 1: $+\frac{V_{REF}}{4}$; •n9 = 0: $-\frac{V_{REF}}{4}$)
↓ 3rd comparison	n9 n8 1 0 0 0 0 0 0 0 (n8: 2nd comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{2048}$ [V] (•n8 = 1: $+\frac{V_{REF}}{8}$; •n8 = 0: $-\frac{V_{REF}}{8}$)
⋮	⋮	⋮
↓ 10th comparison	n9 n8 n7 n6 n5 n4 n3 n2 n1 1	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm ...... \pm \frac{V_{REF}}{1024} - \frac{V_{REF}}{2048}$ [V]
↓ Conversion completed	n9 n8 n7 n6 n5 n4 n3 n2 n1 n0	

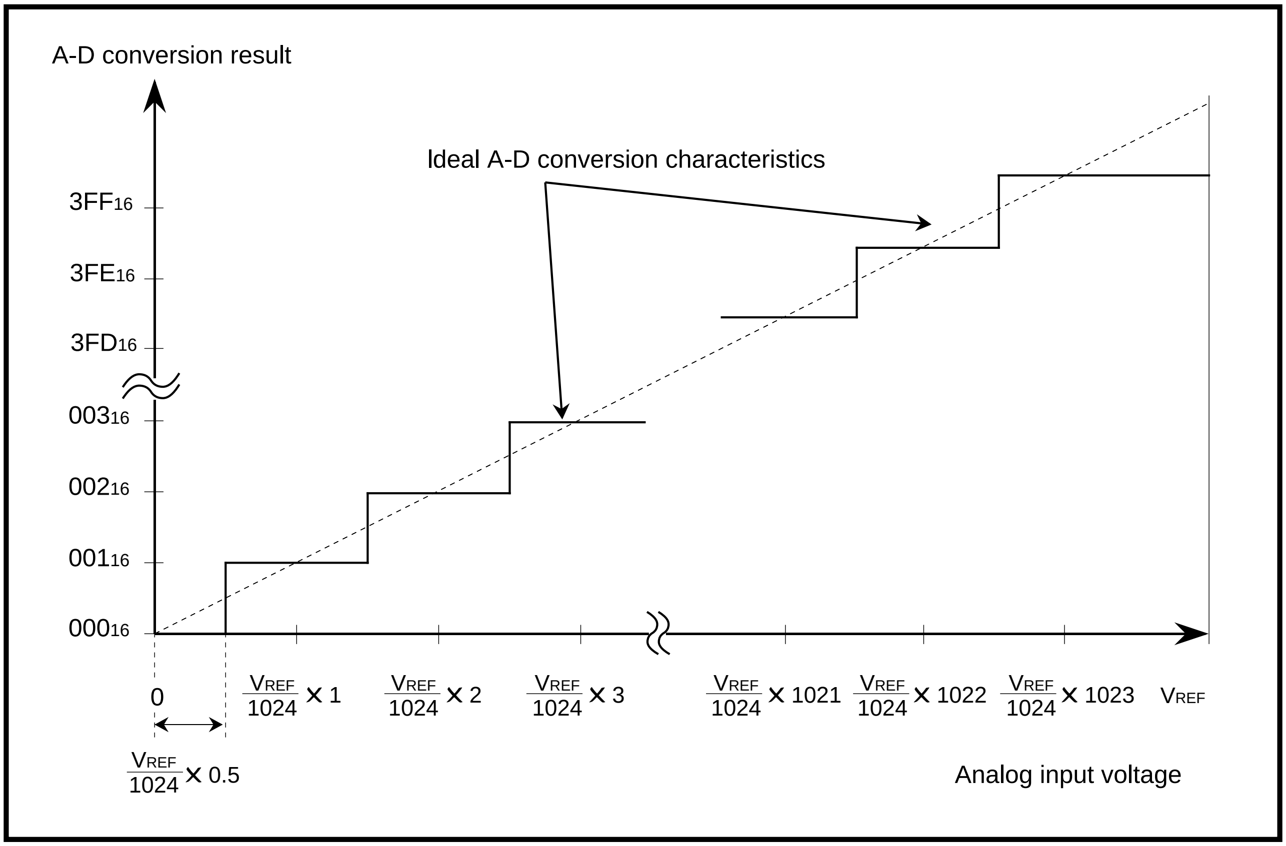


Fig. 12.3.1 Ideal A-D conversion characteristics in 10-bit resolution mode

12.4 Absolute accuracy and Differential non-linearity error

The A-D converter's accuracy is described below. Refer to section **"Appendix 10. 4. A-D converter standard characteristics,"** also.

12.4.1 Absolute accuracy

The absolute accuracy is the difference expressed in the LSB between the actual A-D conversion result and the output code of an A-D converter with ideal characteristics. (See Figure 12.4.1 for more details.)
The analog input voltage at measurement of the absolute accuracy is assumed to be the mid point of the analog input voltage width that outputs the same output code from an A-D converter with ideal characteristics. For example, in the case of the 10-bit resolution mode, when V_{REF} = 5.12 V, 1 LSB width is 5 mV, and 0 mV, 5 mV, 10 mV, 15 mV, 20 mV, ... are selected as the analog input voltages.
The absolute accuracy = ±3 LSB indicates that when the analog input voltage is 25 mV, the output code expected from an ideal A-D conversion characteristics is "005_{16}," but the actual A-D conversion result is between "002_{16}" to "008_{16}."
The absolute accuracy includes the zero error and the full-scale error.
The absolute accuracy degrades when V_{REF} is lowered. Any of the output codes for analog input voltages in the range from V_{REF} to Vcc is "$3FF_{16}$."

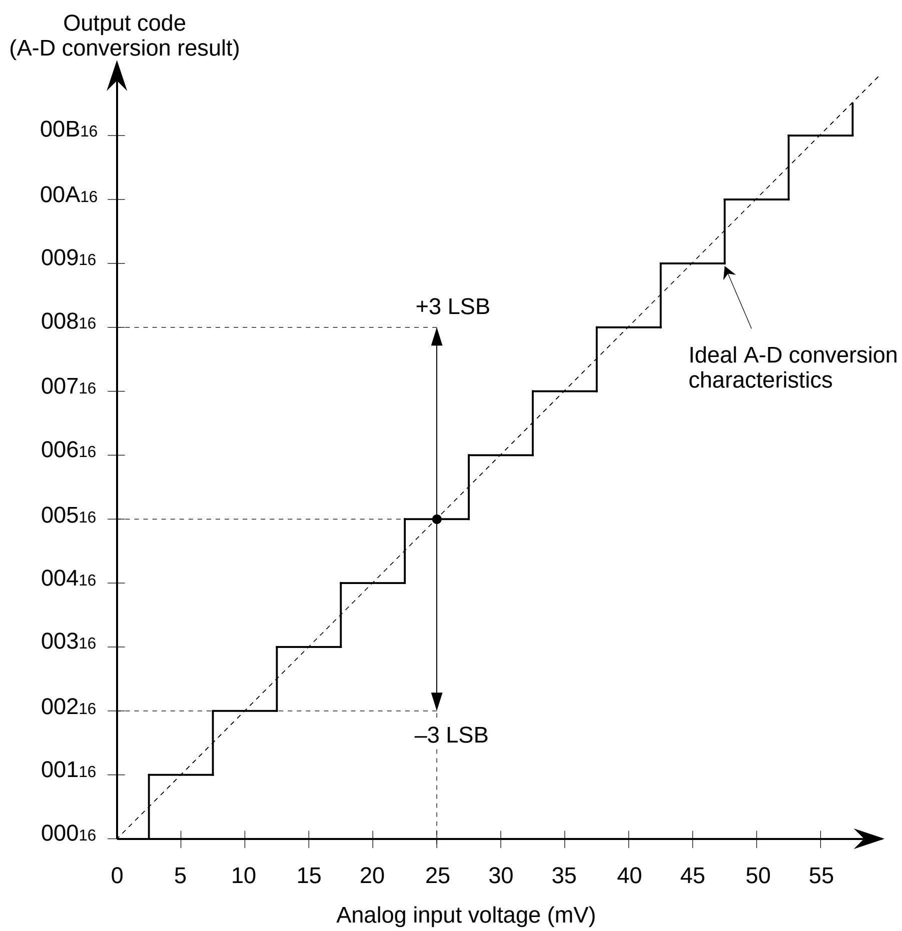


Fig. 12.4.1 Absolute accuracy of A-D converter (10-bit resolution mode)

12.4 Absolute accuracy and Differential non-linearity error

12.4.2 Differential non-linearity error

The differential non-linearity error indicates the difference between the 1 LSB step width (the ideal analog input voltage width while the same output code is expected to output) of an A-D converter with ideal characteristics and the actual measured step width (the actual analog input voltage width while the same output code is output). (See Figure 12.4.2 for more details.) For example, in the case of the 10-bit resolution mode and V_{REF} = 5.12 V, the 1 LSB width of an A-D converter with ideal characteristics is 5 mV; but if the differential non-linearity error is ±1 LSB, the actual measured 1 LSB width is in the range from 0 to 10 mV.

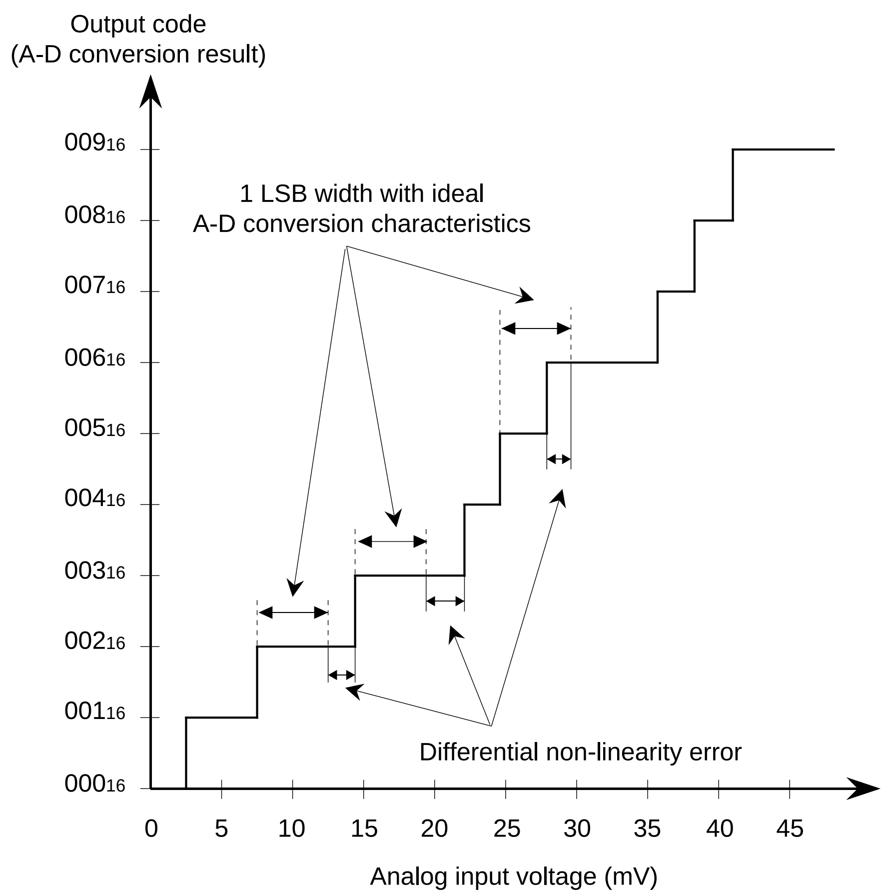


Fig. 12.4.2 Differential non-linearity error (10-bit resolution mode)

12.5 Comparison voltage in 8-bit resolution mode

In the 8-bit resolution mode, which is selected by the resolution select bit, the high-order 8 bits of the 10-bit successive approximation register are treated as the A-D conversion result. Accordingly, when compared with the 8-bit A-D converter, a comparison reference voltage is different by $3V_{REF}/2048$. (Refer to the underlined portions in Table 12.5.1). The difference of the output code change point is generated as shown in Figure 12.5.1.

Table 12.5.1 Comparison voltage

	M37905's 8-bit resolution mode	8-bit A-D converter
Comparison voltage V_{ref}	$\frac{V_{REF}}{2^8} \times n - \underline{\underline{\frac{V_{REF}}{2^{10}} \times 0.5}}$	$\frac{V_{REF}}{2^8} \times n - \underline{\underline{\frac{V_{REF}}{2^8} \times 0.5}}$

V_{REF} : Reference voltage

n : Contents of successive approximation register

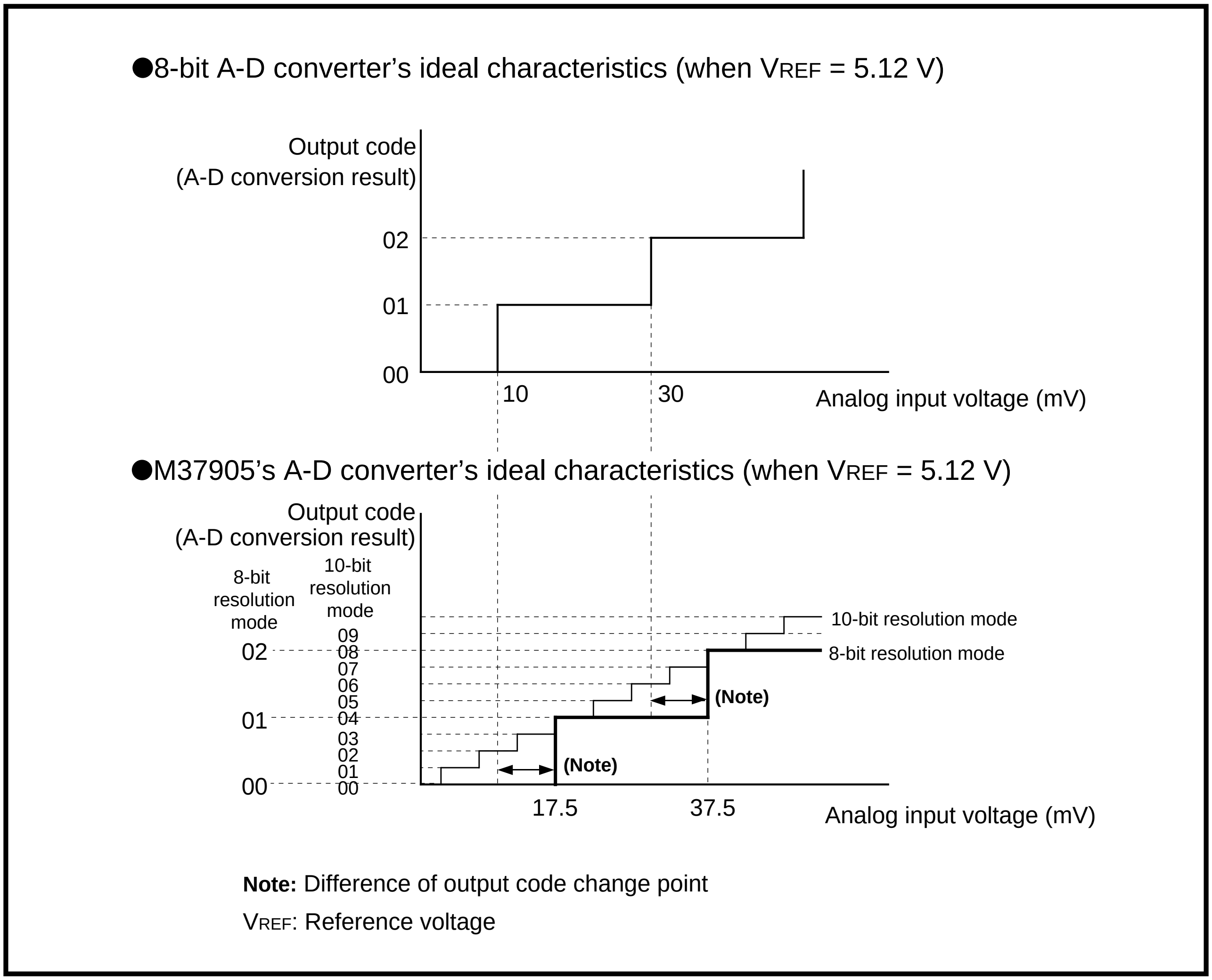


Fig. 12.5.1 Difference of output code change point

12.6 Comparator function

By setting the AN_i pin comparator function select bit (See Figure 12.2.7.) to “1,” the comparator function can be selected for each pin AN_i.
For pin AN_i where the comparator function is selected, the following comparison operation is performed.

① A 10-bit value (a set value), of which high-order 8 bits consist of the corresponding A-D register i (at an even-numbered address)’s contents and of which low-order 2 bits = “10_2,” is D-A converted.

② The result of the D-A conversion (that is to say, comparison voltage V_{ref}) is compared with an analog voltage input from an analog input pin.

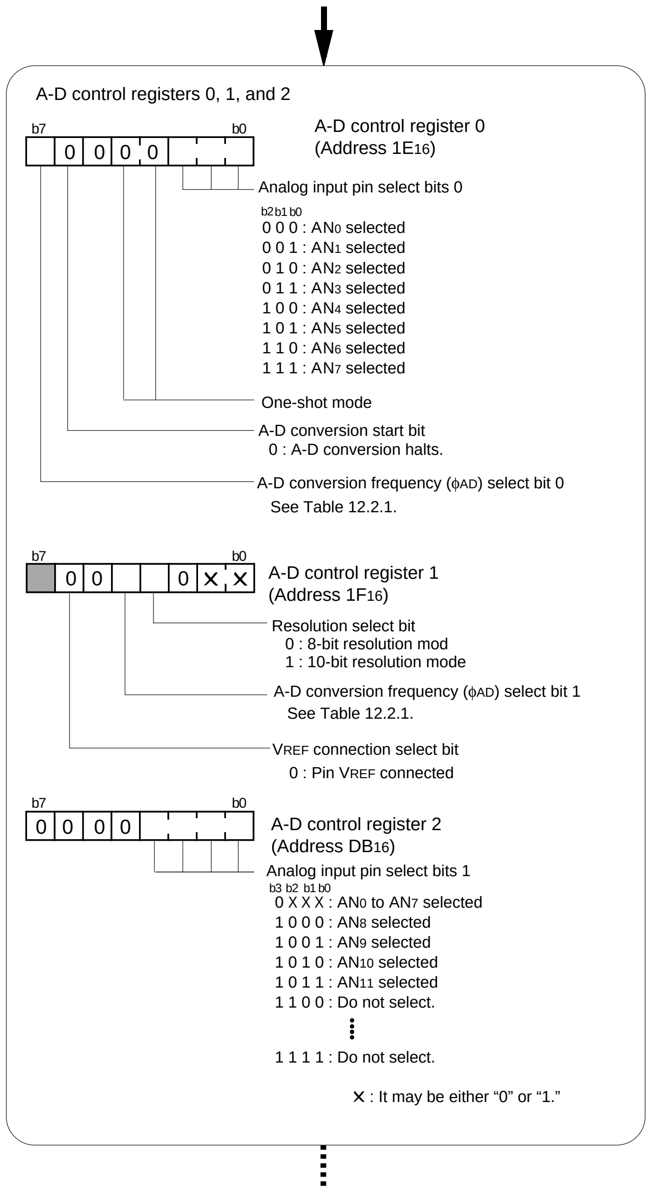
③ The value to be stored into the AN_i pin comparator result bit (see Figure 12.2.8.) depends on the comparison result as follows:
When V_{ref} > analog input voltage, “0” is stored.
When V_{ref} < analog input voltage, “1” is stored.

12.7 One-shot mode

In the one-shot mode, the operation for an input voltage from one selected analog input pin is performed once, and an A-D conversion interrupt request occurs at completion of the operation.
This mode can be used with analog input pin AN_i (i = 0 to 11).

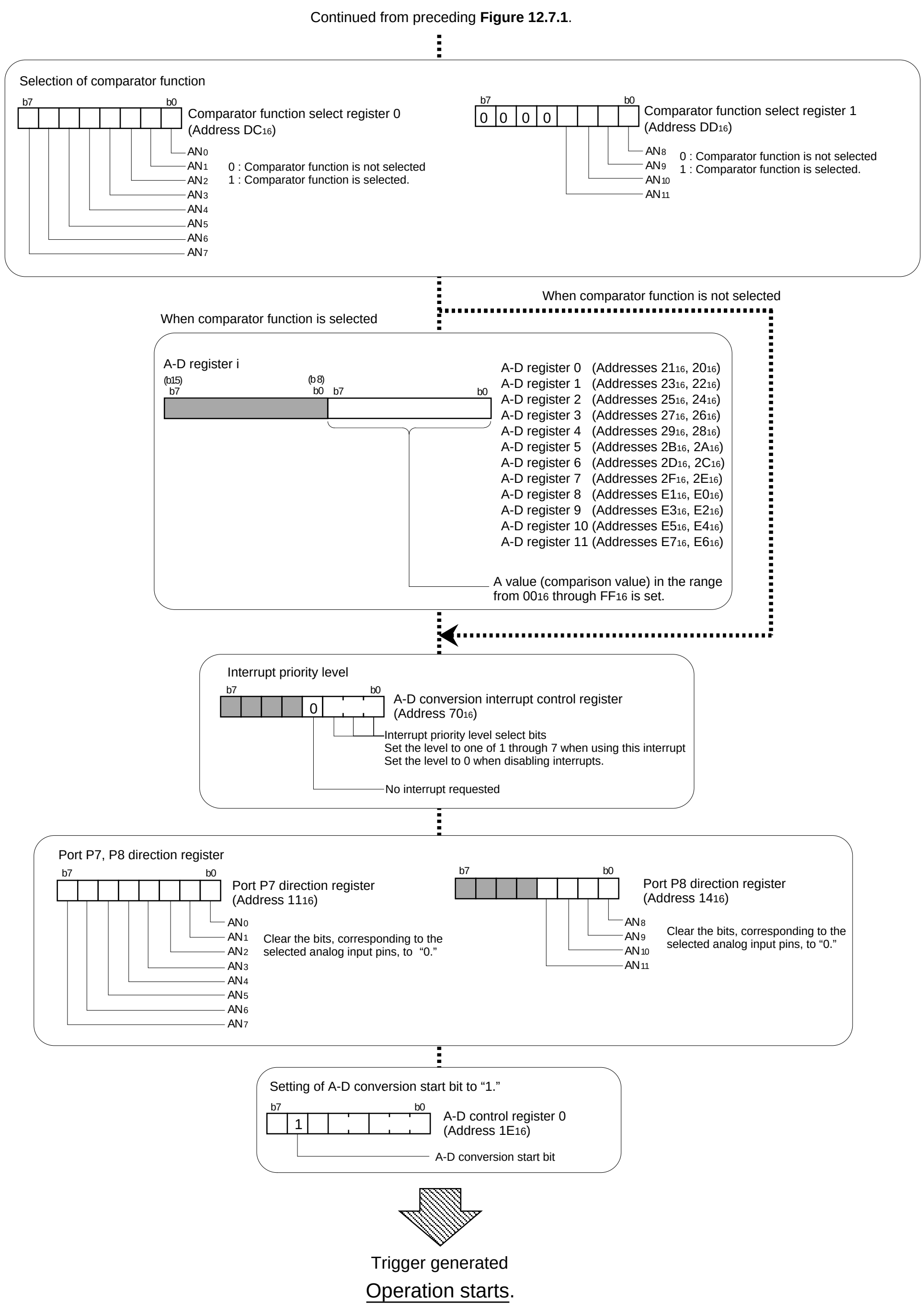
12.7.1 Settings for one-shot mode

Figures 12.7.1 and 12.7.2 show initial setting examples for related registers in the one-shot mode.
When using an interrupt, it is necessary to set the related registers to enable an interrupt. Refer to **"CHAPTER 6. INTERRUPTS"** for more details.



Continued on **Figure 12.7.2**.

Fig. 12.7.1 Initial setting example for related registers in one-shot mode (1)



Note: Writing to the following must be performed while the A-D converter halts (in other words, before a trigger is generated); this must be done independent of the operation mode of the A-D converter.
- Each bit of the A-D control register 0, except writing of "0" to bit 6
- Each bit of the A-D control register 1
- Each bit of the A-D control register 2
- A-D register i (when the comparator function is selected)
- Comparator function select register 0
- Comparator function select register 1

Especially, when the VREF connection select bit is cleared from "1" to "0," an interval of 1 µs or more must be elapsed before occurrence of a trigger.

Fig. 12.7.2 Initial setting example for related registers in one-shot mode (2)

12.7.2 One-shot mode operation

① The A-D converter starts its operation when the A-D conversion start bit is set to “1.”

② The A-D conversion is completed after 49 cycles of ϕAD in the 8-bit resolution mode, or 59 cycles of ϕAD in the 10-bit resolution mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register i.
When the comparator function is selected, the comparison is completed after 14 cycles of ϕ_{AD}. Then, the result of the comparison is stored into the AN_i pin comparator result bit.

③ At the same time as step ②, the A-D conversion interrupt request bit is set to “1.”

④ The A-D conversion start bit is cleared to “0,” and the A-D converter halts.

Figure 12.7.3 shows the operation in the one-shot mode.

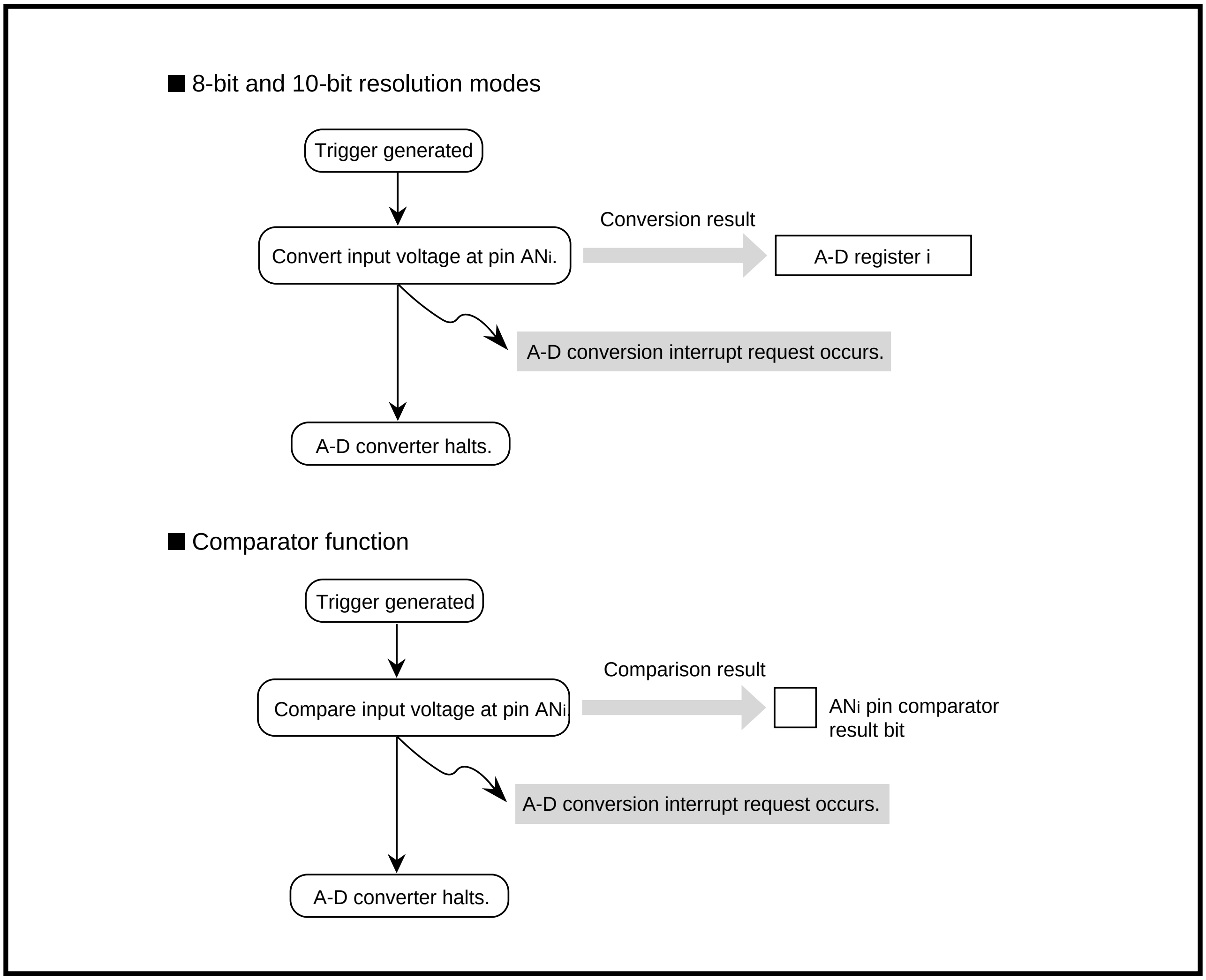


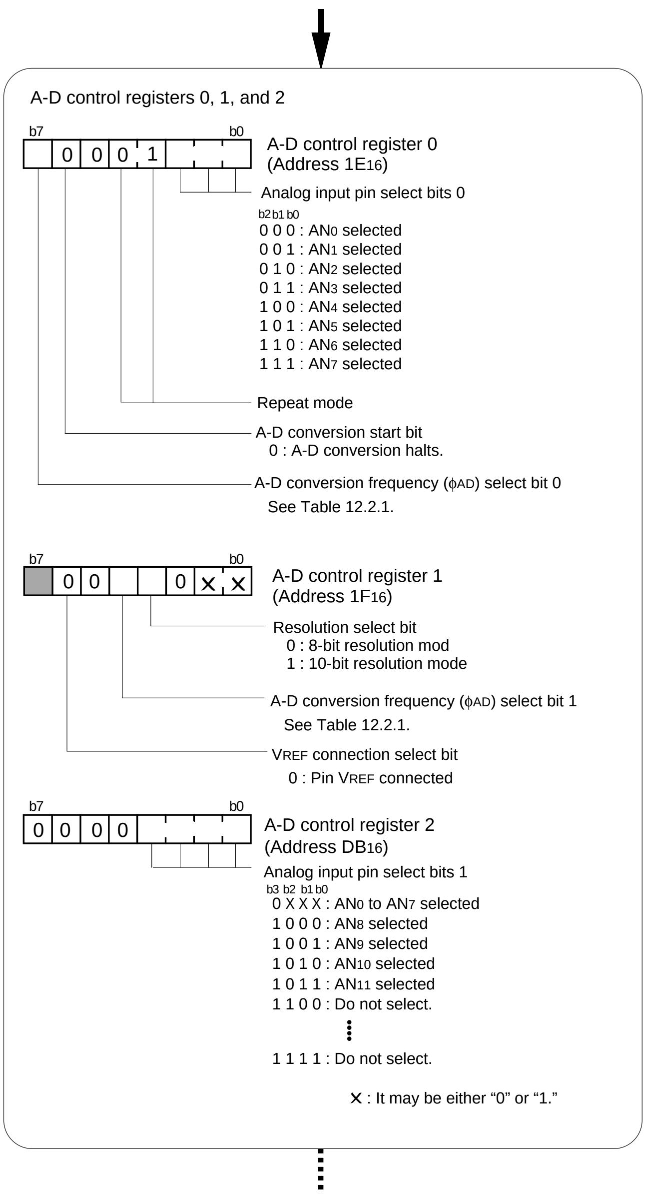
Fig. 12.7.3 Operation in one-shot mode

12.8 Repeat mode

In the repeat mode, the A-D conversion for an input voltage from one selected analog input pin is performed repeatedly.
In this mode, no A-D conversion interrupt request occurs. Additionally, the A-D conversion start bit (bit 6 at address $1E_{16}$) remains set to "1" until it is cleared to "0" by software, and the A-D converter repeats its operation while the A-D conversion start bit = "1."
This mode can be used with analog input pin AN_i (i = 0 to 11).

12.8.1 Settings for repeat mode

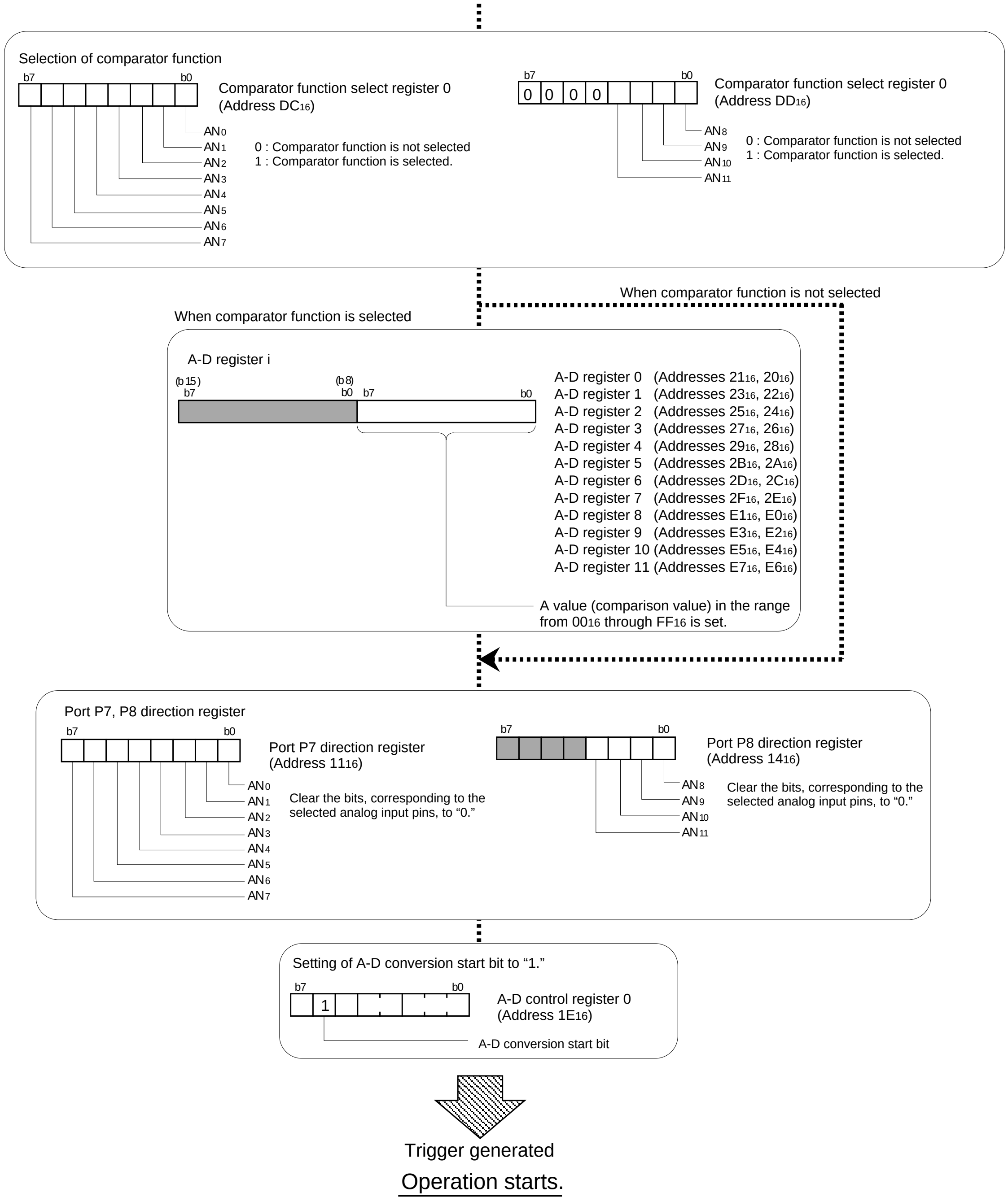
Figures 12.8.1 and 12.8.2 show initial setting examples for related registers in the repeat mode.



Continued on **Figure 12.8.2**.

Fig. 12.8.1 Initial setting example for related registers in repeat mode (1)

Continued from preceding **Figure 12.8.1**.



Note: Writing to the following must be performed while the A-D converter halts (in other words, before a trigger is generated); this must be done independent of the operation mode of the A-D converter.
- Each bit of the A-D control register 0, except writing of "0" to bit 6
- Each bit of the A-D control register 1
- Each bit of the A-D control register 2
- A-D register i (when the comparator function is selected)
- Comparator function select register 0
- Comparator function select register 1

Especially, when the VREF connection select bit is cleared from "1" to "0," an interval of 1 µs or more must be elapsed before occurrence of a trigger.

Fig. 12.8.2 Initial setting example for related registers in repeat mode (2)

12.8.2 Repeat mode operation

① The A-D converter starts its operation when the A-D conversion start bit is set to "1."

② The 1st A-D conversion is completed after 49 cycles of ϕ_{AD} in the 8-bit resolution mode, or 59 cycles of ϕAD in the 10-bit resolution mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register i.
When the comparator function is selected, the 1st comparison is completed after 14 cycles of ϕ_{AD}. Then, the result of the comparison is stored into the AN_i pin comparator result bit.

③ The A-D converter repeats its operation until the A-D conversion start bit is cleared to "0" by software. The conversion result is transferred to the A-D register i each time the conversion is completed. When the comparator function is selected, the comparison result is stored into the AN_i pin comparator result bit each time the comparison is completed.

Figure 12.8.3 shows the operation in the repeat mode.

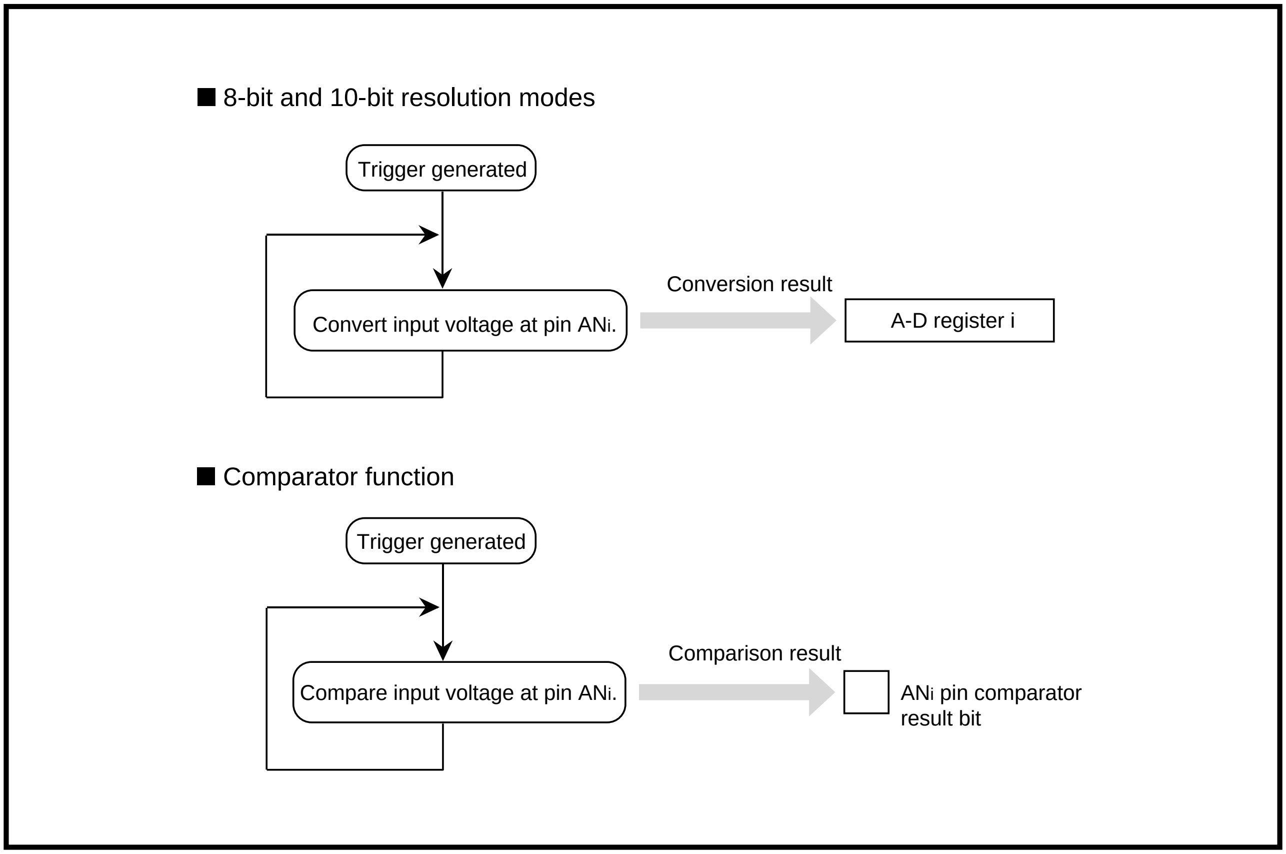


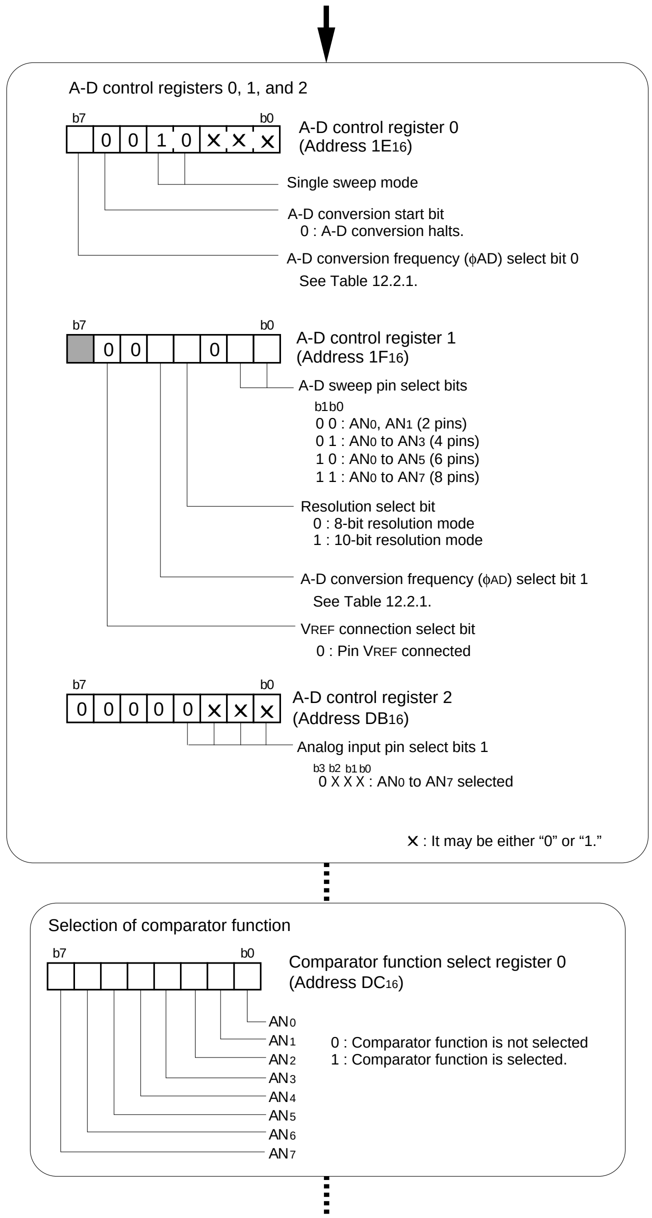
Fig. 12.8.3 Operation in repeat mode

12.9 Single sweep mode

In the single sweep mode, the operation for the input voltages from multiple selected analog input pins are performed, one at a time. The operation is performed in ascending sequence from pin AN_0 to pin AN_7. An A-D conversion interrupt request occurs when the operations for all selected analog input pins are completed. This mode can be used with analog input pins AN_j (j = 0 to 7).

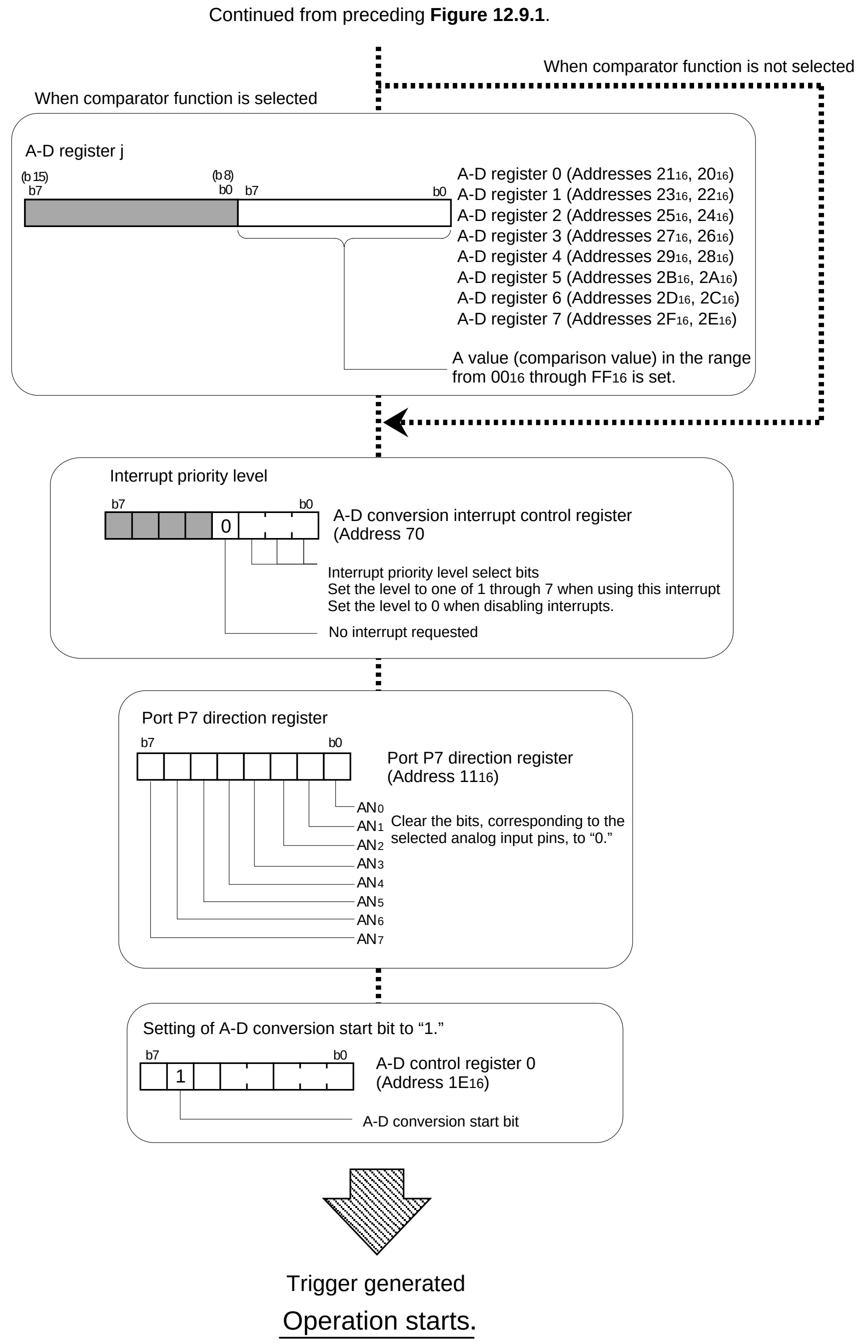
12.9.1 Settings for single sweep mode

Figures 12.9.1 and 12.9.2 show initial setting examples for related registers in the single sweep mode. When using an interrupt, it is necessary to set the related registers to enable an interrupt. Refer to **"CHAPTER 6. INTERRUPTS"** for more details.



Continued on **Figure 12.9.2**.

Fig. 12.9.1 Initial setting example for related registers in single sweep mode (1)



Note: Writing to the following must be performed while the A-D converter halts (in other words, before a trigger is generated); this must be done independent of the operation mode of the A-D converter.
- Each bit of the A-D control register 0, except writing of "0" to bit 6
- Each bit of the A-D control register 1
- Each bit of the A-D control register 2
- A-D register j (when the comparator function is selected)
- Comparator function select register 0

Especially, when the VREF connection select bit is cleared from "1" to "0," an interval of 1 µs or more must be elapsed before occurrence of a trigger.

Fig. 12.9.2 Initial setting example for related registers in single sweep mode (2)

12.9.2 Single sweep mode operation

① The A-D converter starts its operation for the input voltage at pin AN_0 when the A-D conversion start bit is set to "1."

② The A-D conversion for the input voltage at pin AN_0 is completed after 49 cycles of ϕ_{AD} in the 8-bit resolution mode, or 59 cycles of ϕ_{AD} in the 10-bit resolution mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.
When the comparator function is selected, the comparison for pin AN_0 is completed after 14 cycles of ϕ_{AD}. Then, the result of the comparison is stored into the AN_0 pin comparator result bit.

③ The operations for all selected analog input pins are performed.
In the 8-bit and 10-bit resolution modes, the conversion result is transferred to the corresponding A-D register j each time when the A-D conversion per one pin is completed. When the comparator function is selected, the comparison result is stored into the AN_j pin comparator result bit each time the comparison for one pin is completed.

④ When step ③ is completed, the A-D conversion interrupt request bit is set to "1."

⑤ The A-D conversion start bit is cleared to "0," and the A-D converter halts.

Figure 12.9.3 shows the operation in the single sweep mode.

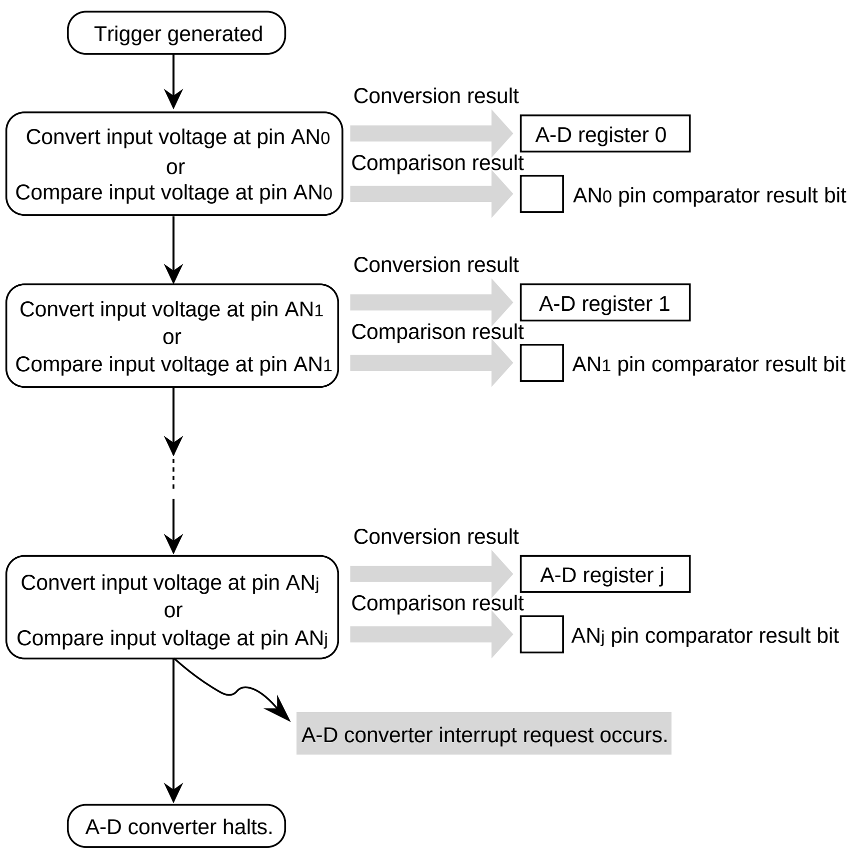


Fig. 12.9.3 Operation in single sweep mode

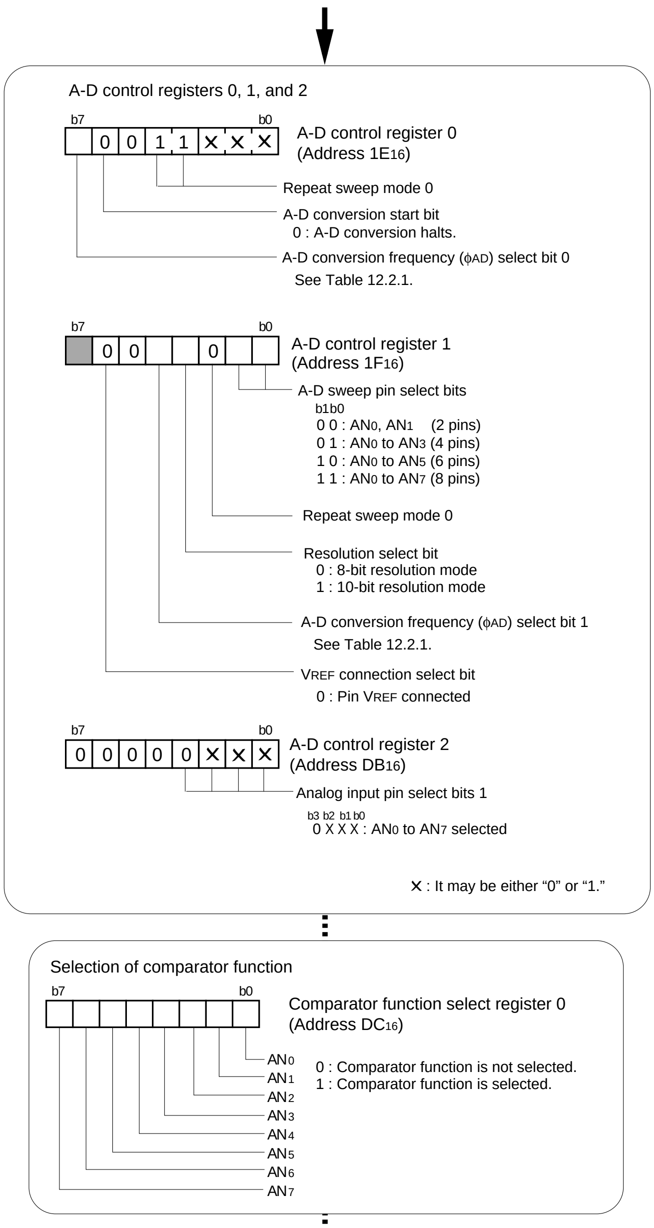
12.10 Repeat sweep mode 0

In the repeat sweep mode, the A-D conversions for input voltages from multiple selected analog input pins are performed repeatedly. The A-D conversion is performed in ascending sequence from pin AN_0 to pin AN_7. In this mode, no A-D conversion interrupt request occurs. Additionally, the A-D conversion start bit (bit 6 at address $1E_{16}$) remains set to "1" until it is cleared to "0" by software, and the A-D converter repeats its operation while the A-D conversion start bit = "1."

This mode can be used with analog input pins AN_j (j = 0 to 7).

12.10.1 Settings for repeat sweep mode 0

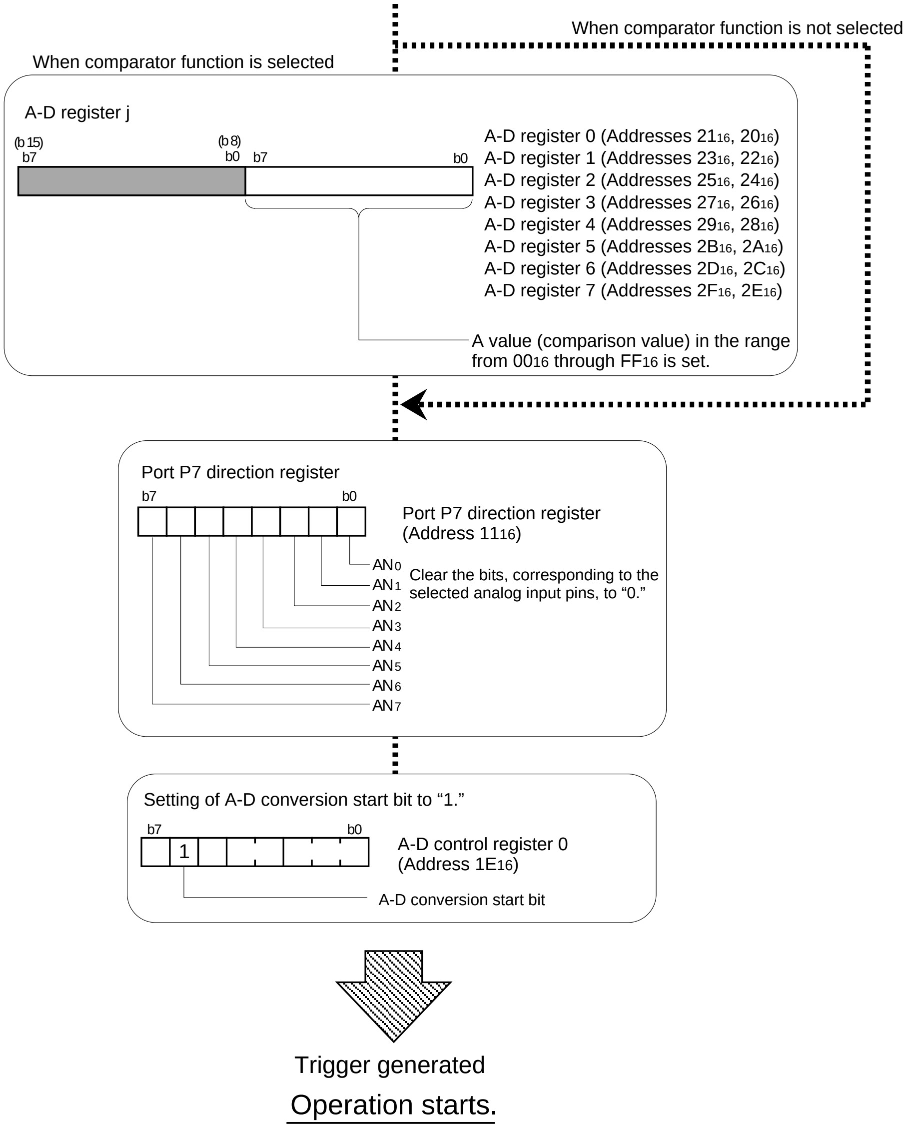
Figures 12.10.1 and 12.10.2 show initial setting examples for related registers in the repeat sweep mode 0.



Continued on **Figure 12.10.2**.

Fig. 12.10.1 Initial setting example for related registers in repeat sweep mode 0 (1)

Continued from preceding **Figure 12.10.1**.



Note: Writing to the following must be performed while the A-D converter halts (in other words, before a trigger is generated); this must be done independent of the operation mode of the A-D converter.
- Each bit of the A-D control register 0, except writing of "0" to bit 6
- Each bit of the A-D control register 1
- Each bit of the A-D control register 2
- A-D register j (when the comparator function is selected)
- Comparator function select register 0

Especially, when the VREF connection select bit is cleared from "1" to "0," an interval of 1 μs or more must be elapsed before occurrence of a trigger.

Fig. 12.10.2 Initial setting example for related registers in repeat sweep mode 0 (2)

12.10.2 Repeat sweep mode 0 operation

① The A-D converter starts its operation for the input voltage at pin AN_0 when the A-D conversion start bit is set to "1."

② The A-D conversion for the input voltage at pin AN_0 is completed after 49 cycles of ϕ_{AD} in the 8-bit resolution mode, or 59 cycles of ϕ_{AD} in the 10-bit resolution mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.
When the comparator function is selected, the comparison for pin AN_0 is completed after 14 cycles of ϕ_{AD}. Then, the result of the comparison is stored into the AN_0 pin comparator result bit.

③ The operations for all selected analog input pins are performed.
The conversion result is transferred to the corresponding A-D register j each time when the A-D conversion per one pin is completed. When the comparator function is selected, the comparison result is stored into the AN_j pin comparator result bit each time the comparison for one pin is completed.

④ The operations for all selected analog input pins are performed again.

⑤ The A-D converter repeats its operation until the A-D conversion start bit is cleared to "0" by software.

Figure 12.10.3 shows the operation in the repeat sweep mode 0.

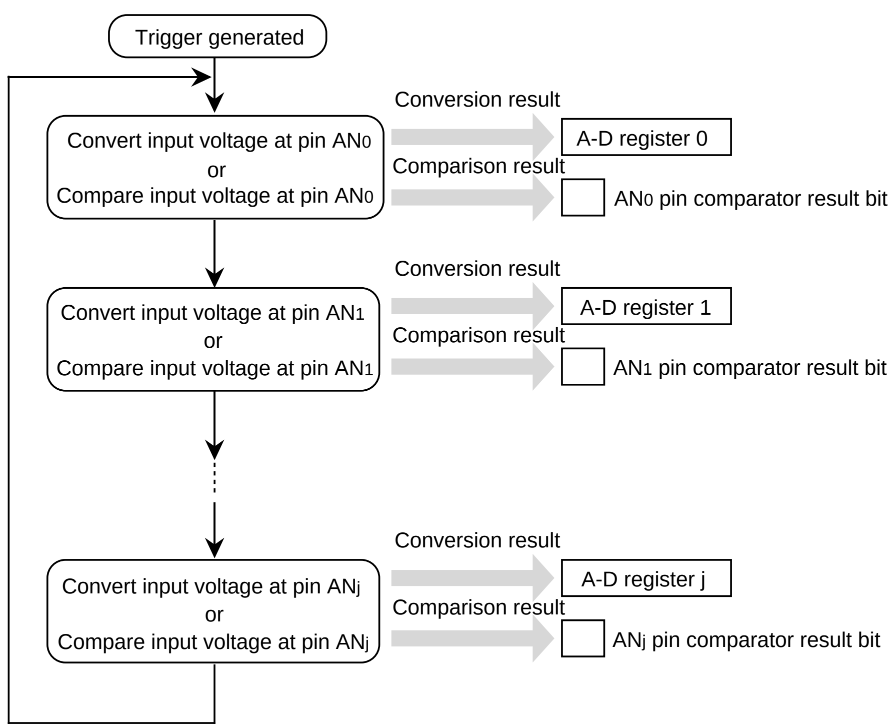


Fig. 12.10.3 Operation in repeat sweep mode 0

12.11 Repeat sweep mode 1

In the repeat sweep mode 1, the A-D conversions for input voltages from multiple selected analog input pins AN_j (j = 0 to 7) are performed repeatedly. In this mode, analog input pins AN_j are divided into two groups: frequently-used pins and non-frequently-used pins. Then, the operation for all of the frequently-used pins is performed. Next, the operation for one of the non-frequently-used pins is performed. Figure 12.11.1 shows the operation sequence in the repeat sweep mode 1. As shown in Figure 12.11.1, the non-frequently-used pin changes sequentially.
In this mode, no A-D conversion interrupt request occurs. Additionally, the A-D conversion start bit (bit 6 at address $1E_{16}$) remains set to "1" until it is cleared to "0" by software, and the A-D converter repeats its operation while the A-D conversion start bit = "1."
This mode can be used with analog input pins AN_j (j = 0 to 7).

12.11.1 Settings for repeat sweep mode 0

Figures 12.11.2 and 12.10.3 show initial setting examples for related registers in the repeat sweep mode 1. Be sure to select the frequently-used analog input pins by the A-D sweep pin select bits (bits 1 and 0 at address $1F_{16}$). All pins that are not selected by the A-D sweep pin select bits become the non-frequently-used pins.

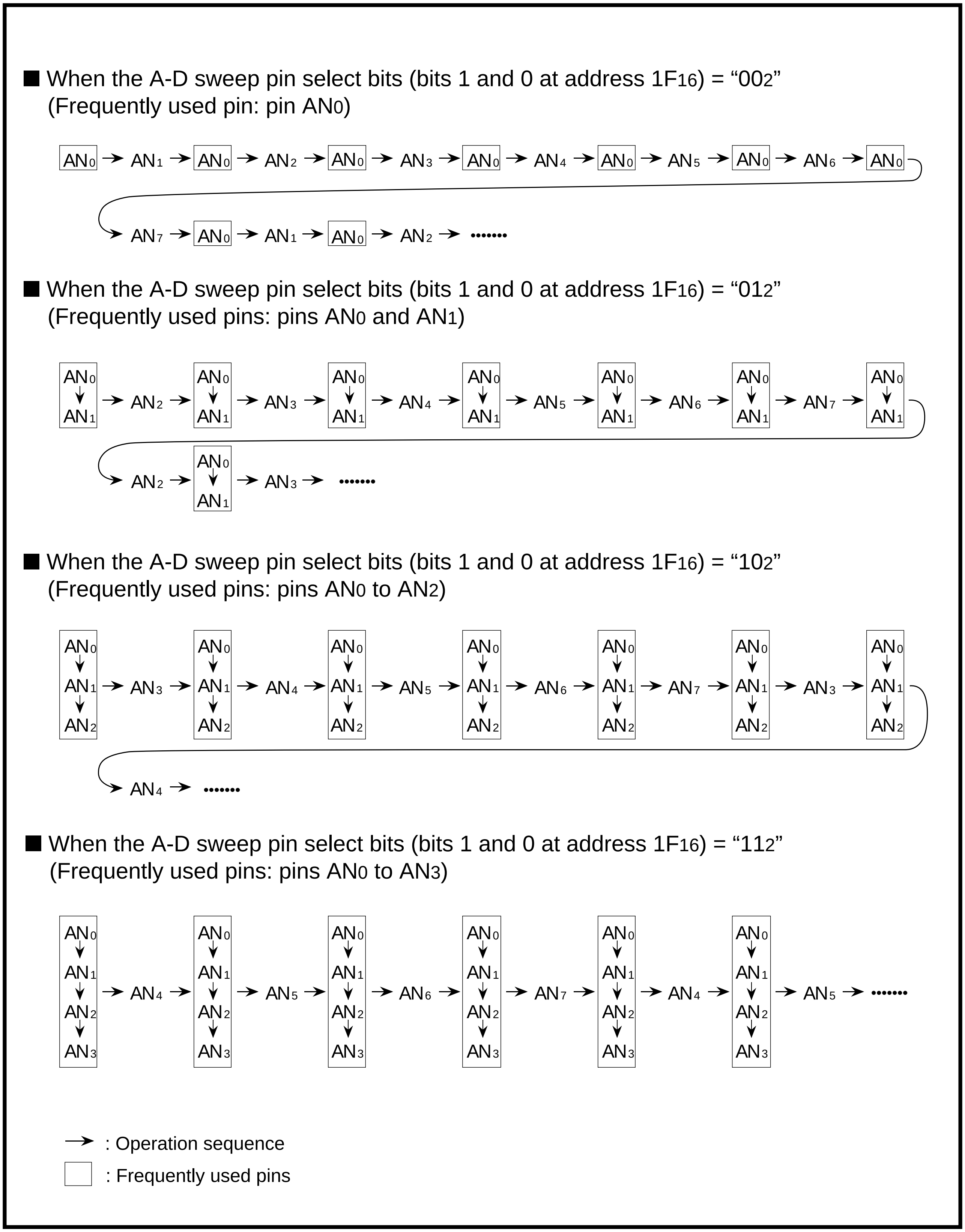
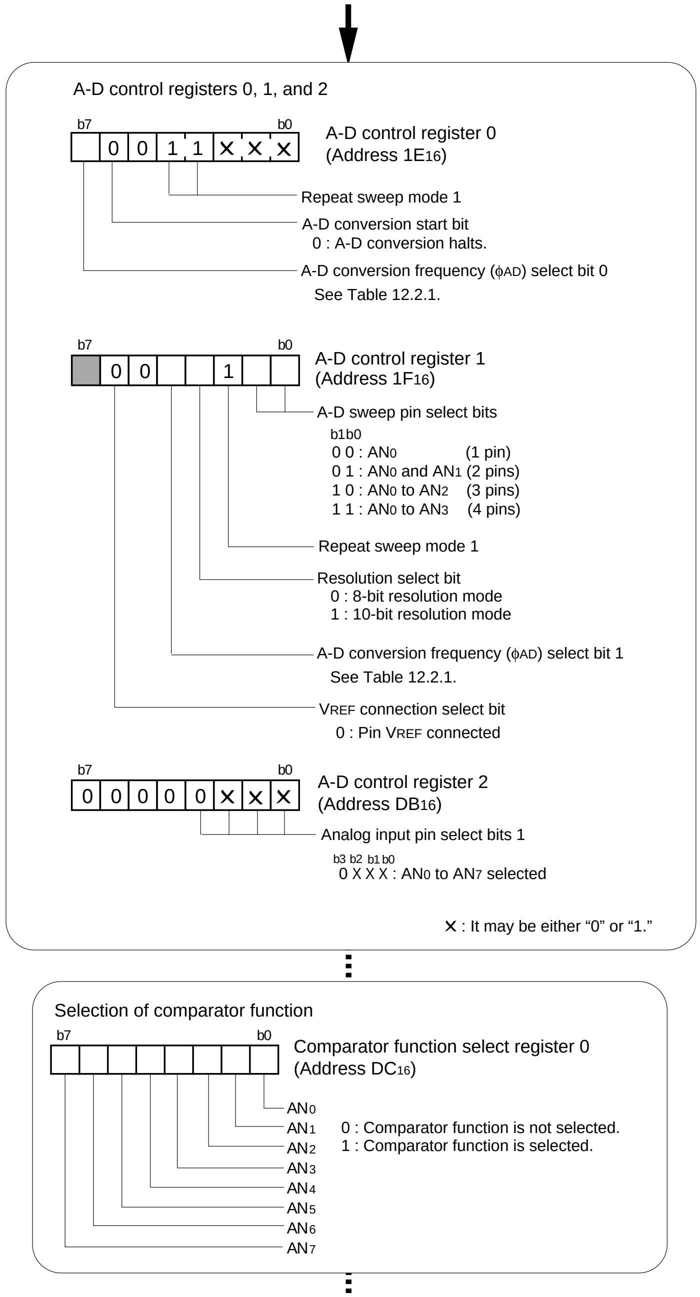


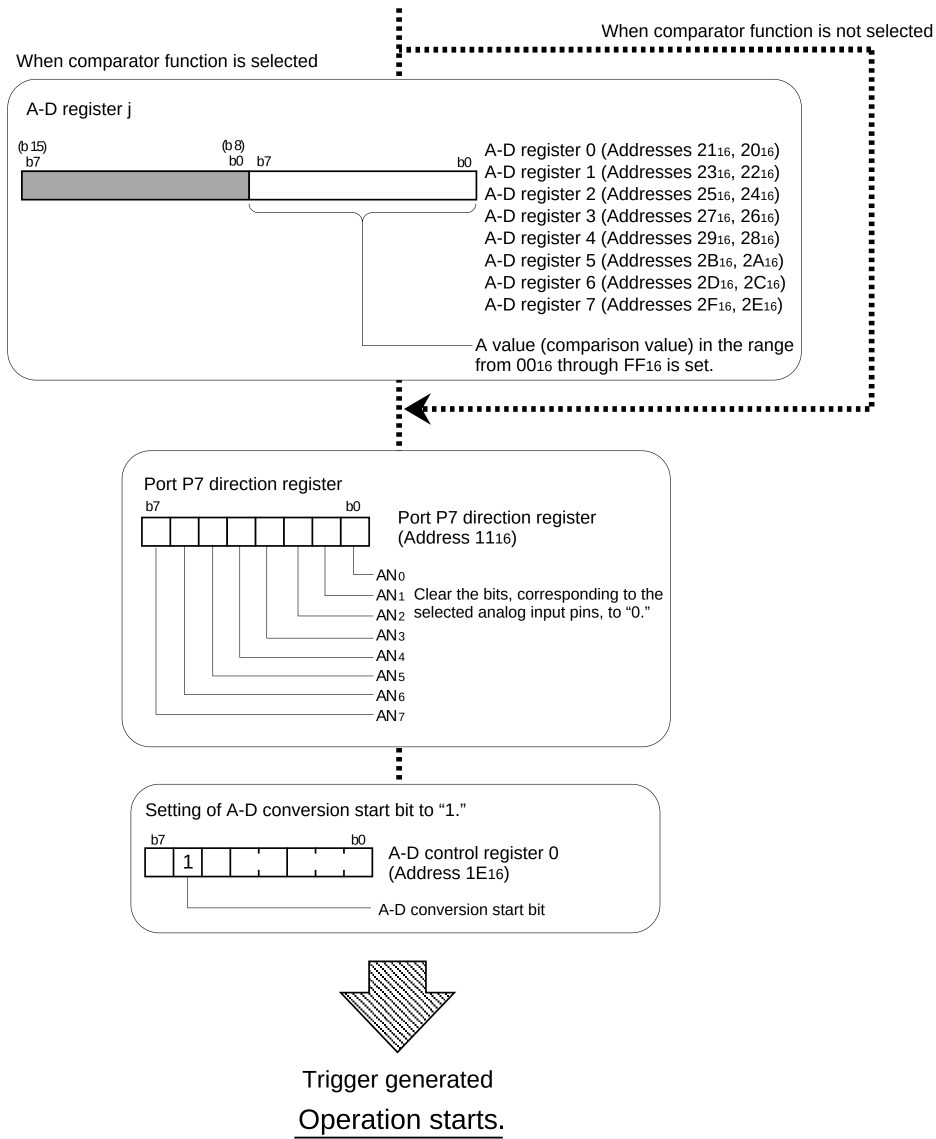
Fig. 12.11.1 Operation sequence in repeat sweep mode 1



Continued on **Figure 12.11.3**.

Fig. 12.11.2 Initial setting example for related registers in repeat sweep mode 1 (1)

Continued from preceding **Figure 12.11.2**.



Note: Writing to the following must be performed while the A-D converter halts (in other words, before a trigger is generated); this must be done independent of the operation mode of the A-D converter.

- Each bit of the A-D control register 0, except wrinting of "0" to bit 6
- Each bit of the A-D control register 1
- Each bit of the A-D control register 2
- A-D register j (when the comparator function is selected)
- Comparator function select register 0

Especially, when the VREF connection select bit is cleared from "1" to "0," an interval of 1 µs or more must be elapsed before occurrence of a trigger.

Fig. 12.11.3 Initial setting example for related registers in repeat sweep mode 1 (2)

12.11.2 Repeat sweep mode 1 operation

① The A-D converter starts its operation for the input voltage at pin AN_0 when the A-D conversion start bit is set to "1."

② The A-D conversion for the input voltage at pin AN_0 is completed after 49 cycles of ϕ_{AD} in the 8-bit resolution mode, or 59 cycles of ϕ_{AD} in the 10-bit resolution mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.
When the comparator function is selected, the comparison for pin AN_0 is completed after 14 cycles of ϕ_{AD}. Then, the result of the comparison is stored into the AN_0 pin comparator result bit.

③ The operations for all of the frequently-used analog input pins is performed. The conversion result is transferred to the corresponding A-D register j each time when the A-D conversion per one pin is completed. When the comparator function is selected, the comparison result is stored into the AN_j pin comparator result bit each time the comparison for one pin is completed.

④ The operation for one of the non-frequently-used analog input pins is performed. (See Figure 12.11.1.)

⑤ The operation for all of the frequently-used analog input pins is performed again.

⑥ The operation for one of the non-frequently-used analog input pins is performed. This pin differs from the pin used in ④. (See Figure 12.11.1.)

⑦ The A-D converter repeats its operation until the A-D conversion start bit is cleared to "0" by software.

[Precautions for A-D converter]

1. Be sure to clear the V_{REF} connection select bit to "0."

2. Writing to the following must be performed before a trigger is generated (in other words, while the A-D converter halts); this must be done independent of the operation mode of the A-D converter.
   - Each bit of the A-D control register 0, except writing of "0" to bit 6
   - Each bit of the A-D control register 1
   - Each bit of the A-D control register 2
   - A-D register i (when the comparator function is selected)
   - Comparator function select register 0
   - Comparator function select register 1
   - Comparator result register 0
   - Comparator result register 1

   Especially, when any instruction which clears the V_{REF} connection select bit from "1" to "0" has been executed (in other words, the resistor ladder network is connected with pin V_{REF} by this instruction), an interval of 1 µs or more must be elapsed before occurrence of a trigger.

3. When using pins AN_0 to AN_7, regardless of the A-D operation mode, be sure to fix bit 3 of the analog input pin select bits 1 (bits 3 to 0 at address DB_{16}) to "0."

4. Pins AN_8 to AN_{11} can be used only in the one-shot mode or repeat mode.

5. The analog input pin select bits 0 (bits 2 to 0 at address $1E_{16}$) and the analog input pin select bits 1 (bits 3 to 0 at address DB_{16}) must be specified again if the user switches the operation mode to the one-shot mode or repeat mode after the operation is performed in the single sweep mode, repeat sweep mode 0, or repeat sweep mode 1.

6. Reading from A-D register i (when the comparator function is selected) must be performed before occurrence of a trigger (in other words, while the A-D converter halts.). The value undefined at reading.

7. When using pin AN_7, be sure that the D-A_0 output enable bit (bit 0 at address 96_{16}) = "0" (output disabled).
   When using pin AN_8, be sure that the D-A_1 output enable bit (bit 1 at address 96_{16}) = "0" (output disabled). Also, be sure not to use pin $\overline{CTS_2}/\overline{RTS_2}$.
   When using pin AN_9, be sure not to use pin $\overline{CTS_2}/CLK_2$.
   When using pin AN_{10}, be sure not to use pin R_XD_2.
   When using pin AN_{11}, be sure not to use pin T_XD_2.

8. Setting of bit 3 of the analog input pin select bits 1 (bits 3 to 0 at address DB_{16}) to "1" invalidates the analog input pin select bits 0 (bits 2 to 0 at address $1E_{16}$).

9. Refer to section **"Appendix 7. Countermeasures against noise"** when using the A-D converter.

CHAPTER 13
D-A CONVERTER

13.1 Overview

The M37905 is provided with two independent D-A converters of the R-2R type with 8-bit resolution. These D-A converters convert the values loaded in D-A register i (i = 0, 1) to analog voltages and output them from pin DA_i.

13.2 Block description

Figure 13.2.1 shows the block diagram of the D-A converter. The registers related to the D-A converter are described below.

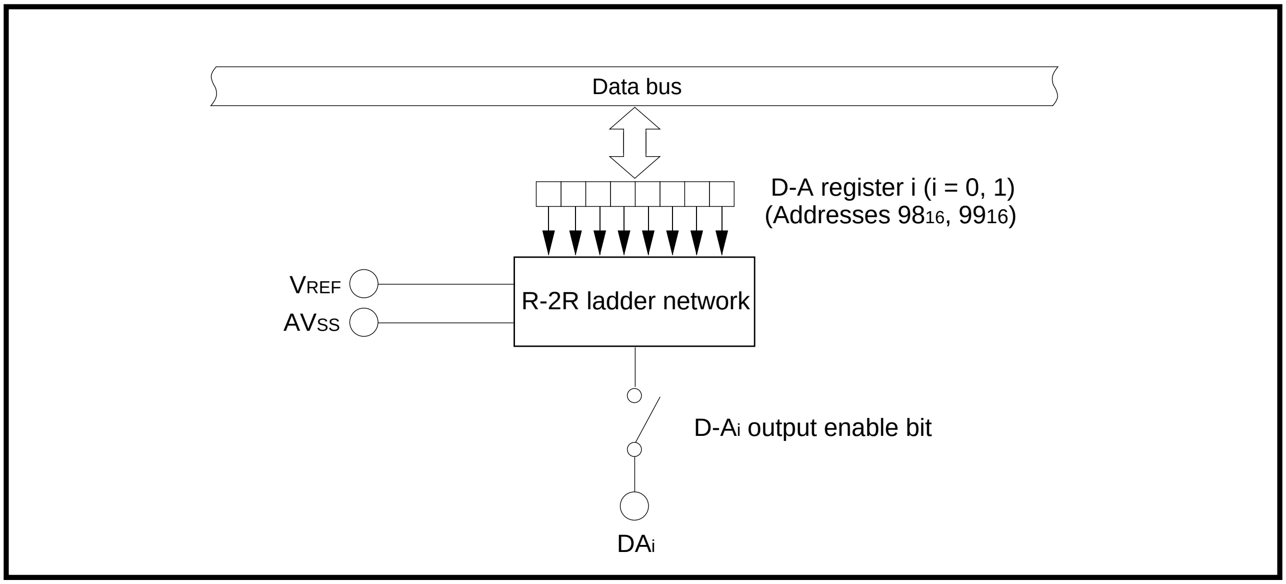


Fig. 13.2.1 D-A converter block diagram

13.2.1 D-A control register

Figure 13.2.2 shows the structure of the D-A control register.
Pin DA_i (i = 0, 1) serves as the analog voltage output pin of the D-A converter. Since pin DA_i is equipped with no internal buffer amplifier, it is necessary to connect a buffer amplifier externally to pin DA_i, if this pin is needed to be connected with a low-impedance load.
Pin DA_i is multiplexed with an analog input pin and I/O pins for serial I/O. When any of the D-A_i output enable bits is set to “1” (output enabled), the corresponding pin is used only as pin DA_i, not as any other multiplexed input/output pin (including a programmable I/O port pin).

D-A control register (Address 96_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	D-A_0 output enable bit	0: Output is disabled. 1: Output is enabled. **(Notes 1, 2)**	0	RW
1	D-A_1 output enable bit	0: Output is disabled. 1: Output is enabled. **(Notes 1, 2)**	0	RW
7 to 2	Nothing is assigned.		Undefined	—

Notes 1: Pin DA_i is multiplexed with an analog input pin and I/O pins for serial I/O. When a D-A_i output enable bit = “1” (in other words, output is enabled.), however, the corresponding pin cannot function as any other multiplexed input/output pin (including a programmable I/O port pin).
2: When not using the D-A converter, be sure to clear this bit to “0.”

Fig. 13.2.2 Structure of D-A control register

(1) D-A_i output enable bits (Bits 0, 1)

Setting any of the D-A_i output enable bits to “1” (output enabled) allows the corresponding pin DA_i to output D-A converted analog voltage, regardless of the contents of the corresponding bits of the port P7 and port P8 direction registers.

13.2.2 D-A register i (i = 0, 1)

Each pin DA_i outputs the analog voltage corresponding to the value loaded in D-A register i. Figure 13.2.3 shows the structure of D-A register i.

D-A register i (i = 0, 1) (Addresses 98_{16}, 99_{16})

b7 b0

Bit	Function	At reset	R/W
7 to 0	Any value in the range from 00_{16} through FF_{16} can be set **(Note)**, and this value will be D-A converted and will be output.	0	RW

Note: When not using the D-A converter, be sure to clear the contents of these bits to “00_{16}.”

Fig. 13.2.3 Structure of D-A register i

13.3 D-A conversion method

The reference voltage V_{REF} is divided according to the value loaded in D-A register i, and it is output as an analog voltage from pin DA_i.

Figure 13.3.1 shows the equivalent circuit diagram of the D-A converter.

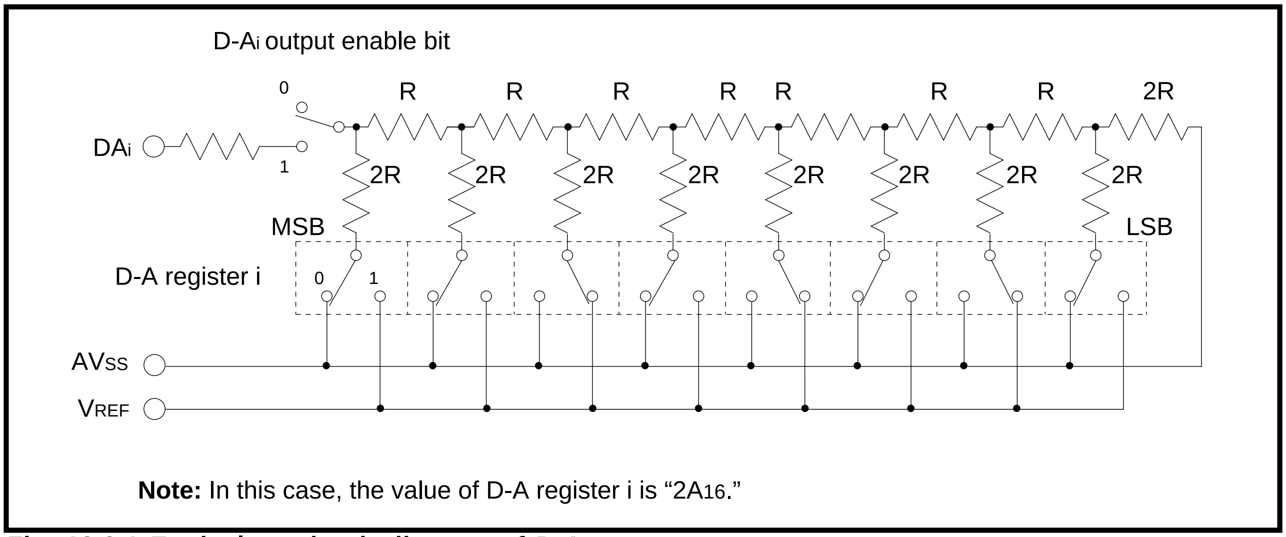


Fig. 13.3.1 Equivalent circuit diagram of D-A converter

13.4 Setting method

Figure 13.4.1 shows an initial setting example of registers related to the D-A converter.

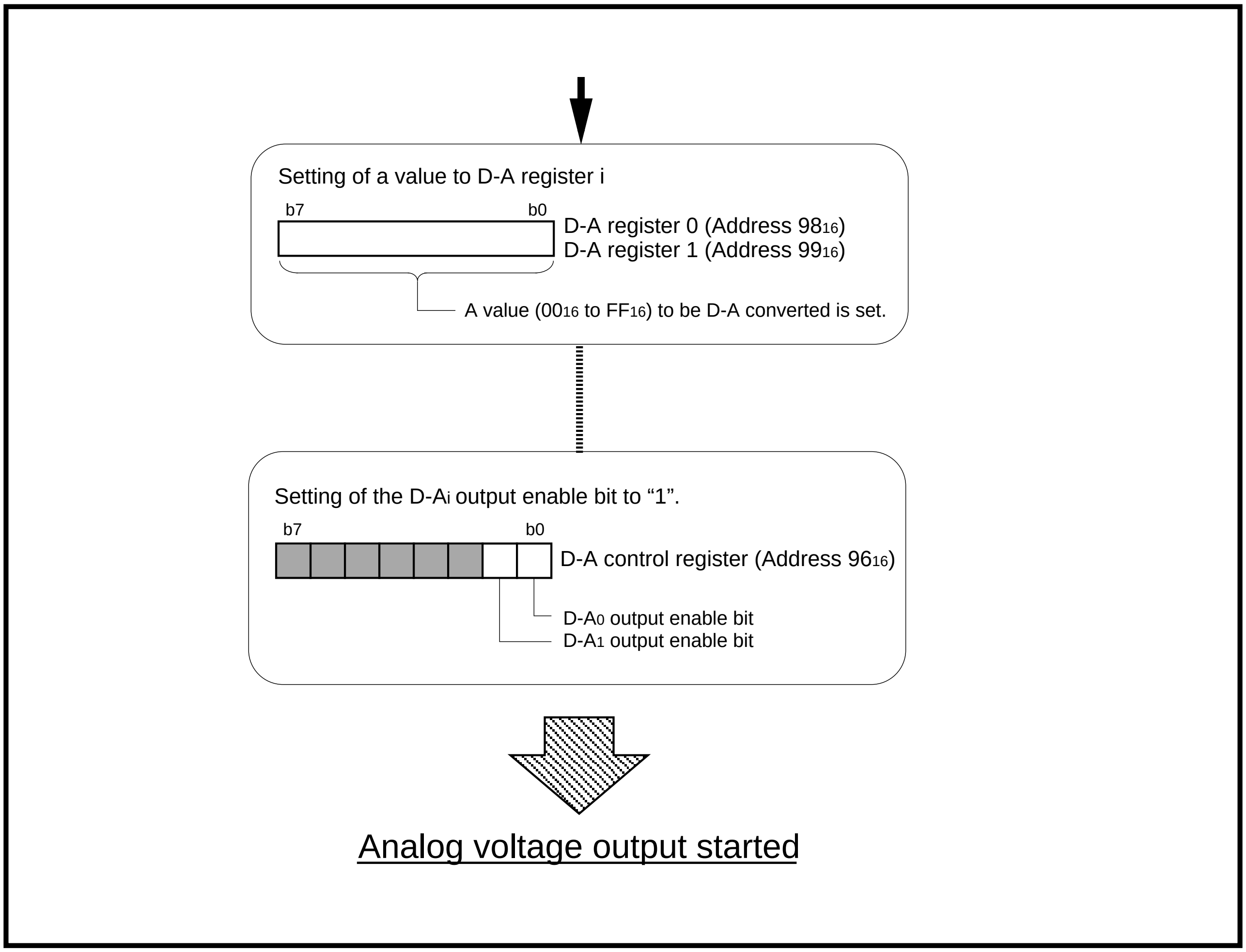


Fig. 13.4.1 Initial setting example of registers related to D-A converter

13.5 Operation description

When any of the D-A$_i$ output enable bits is set to "1," the value loaded in D-A register i is converted to an analog voltage, and the analog voltage is output from pin DA$_i$.

The relationship between analog output voltage V and value n, which has been loaded in D-A register i, can be expressed as follows :

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

V_{REF} : Reference voltage

[Precautions for D-A converter]

1. Pin DA_i is multiplexed with an analog input pin and I/O pins for serial I/O. When any of the D-A_i output enable bits is set to "1" (output enabled), the corresponding pin is used as pin DA_i, not as any other multiplexed input/output pin (including a programmable I/O port pin).

2. When not using the D-A converter, be sure to do as follows:
   - Clear the D-A_i (i = 0, 1) output enable bit (bits 0, 1 at address 96_{16}) to "0."
   - Clear the contents of D-A register i (addresses 98_{16}, 99_{16}) to "00_{16}."

CHAPTER 14

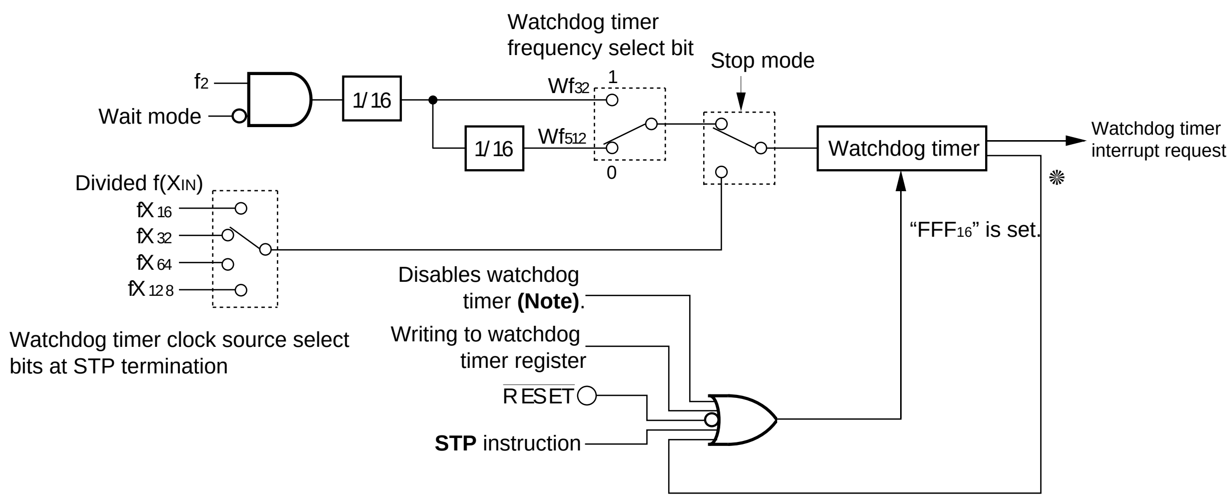
WATCHDOG TIMER

The watchdog timer functions as follows:

- Detects a program runaway.
- At stop mode termination, measures a certain time after oscillation starts. (Refer to section **“15.3 Stop mode.”**)

14.1 Block description

Figure 14.1.1 shows the block diagram of the watchdog timer, and registers relevant to the watchdog timer are described below.



- Watchdog timer register: address 60_{16}
- Watchdog timer frequency select bit: bit 0 at address 61_{16}
- Watchdog timer clock source select bits at STP termination: bits 7, 6 at address 61_{16}

❋ When the most significant bit of the watchdog timer becomes “0,” this signal will be generated.

Note: During the stop mode and until the stop mode is terminated, setting for disabling the watchdog timer is ignored. (Refer to section **“14.1.3 Particular function select register 2.”**)

Fig. 14.1.1 Block diagram of watchdog timer

14.1.1 Watchdog timer

Figure 14.1.2 shows the structure of the watchdog timer register.
The watchdog timer is a 12-bit counter where the count source which is selected with the watchdog timer frequency select bit (bit 0 at address 61_{16}) is counted down. A value of "FFF_{16}" is automatically set in the watchdog timer if any of the following conditions is satisfied. An arbitrary value cannot be set to the watchdog timer.

- When dummy data is written to the watchdog timer register. (See Figure 14.1.2.)
- When the most significant bit of watchdog timer becomes "0."
- When the **STP** instruction is executed. (Refer to section **"15.3 Stop mode."**)
- At reset

Watchdog timer register (Address 60_{16}) — b7 … b0

Bit	Function	At reset	R/W
7 to 0	Initializes the watchdog timer. When dummy data has been written to this register, the watchdog timer's value is initialized to "FFF_{16}" (dummy data: 00_{16} to FF_{16}).	Undefined	—

Fig. 14.1.2 Structure of watchdog timer register

14.1.2 Watchdog timer frequency select register

Figure 14.1.3 shows the structure of the watchdog timer frequency select register.

Watchdog timer frequency select register (Address 61_{16}) — b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Watchdog timer frequency select bit	0 : Wf_{512} 1 : Wf_{32}	0	RW
5 to 1	Nothing is assigned.		Undefined	—
6	Watchdog timer clock source select bits at STP termination	b7 b6 0 0 : fX_{32} 0 1 : fX_{16} 1 0 : fX_{128} 1 1 : fX_{64}	0	RW
7			0	RW

Fig. 14.1.3 Structure of watchdog timer frequency select register

(1) Watchdog timer frequency select bit (bit 0)
This bit is used to select a count source of the watchdog timer.

(2) Watchdog timer clock source select bits at STP termination (bits 7, 6)
These bits are used to select a count source at stop mode termination.
For details of the operation at stop mode termination, refer to section **"15.3 Stop mode."**

14.1.3 Particular function select register 2

When not using the watchdog timer, this register can be used to disable the watchdog timer. Figure 14.1.4 shows the structure of the particular function select register 2.

Particular function select register 2 (Address 64_{16})

b7 b0

Bit	Function	At reset	R/W
7 to 0	Disables the watchdog timer. When values of "79_{16}" and "50_{16}" succeedingly in this order, the watchdog timer will stop its operation.	Undefined	—

Note: After reset, this register can be set only once. Writing to this register requires the following procedure:

- Write values of "79_{16}" and "50_{16}" to this register succeedingly in this order.
- For the above writing, be sure to use the **MOVMB** (**MOVM** when m = 1) instruction or the **STAB** (**STA** when m = 1).

Note that the following: if an interrupt occurs between writing of "79_{16}" and next writing of "50_{16}," the watchdog timer does not stop its operation.

If any of the following has been performed after reset, writing to this register is disabled from that time:

- If this register is read out.
- If writing to this register is performed by the procedure other than the above procedure.

Fig. 14.1.4 Structure of particular function select register 2

In addition, even when the watchdog timer is disabled by this register, the watchdog timer can be active only at the stop mode termination if the external clock input select bit (bit 1 at address 62_{16}) = "0." (Refer to section **"15.3 Stop mode."**)

14.2 Operation description

The operations of the watchdog timer are described below.

14.2.1 Basic operation

① Watchdog timer starts counting down from “FFF_{16}.”
② When the watchdog timer’s most significant bit becomes “0” (counted 2048 times), a watchdog timer interrupt request occurs. (See Table 14.2.1.)
③ When the interrupt request occurs in above ②, a value of “FFF_{16}” is set to the watchdog timer.

A watchdog timer interrupt is a non-maskable interrupt. When a watchdog timer interrupt request is accepted, the processor interrupt priority level (IPL) is set to “111_2.”

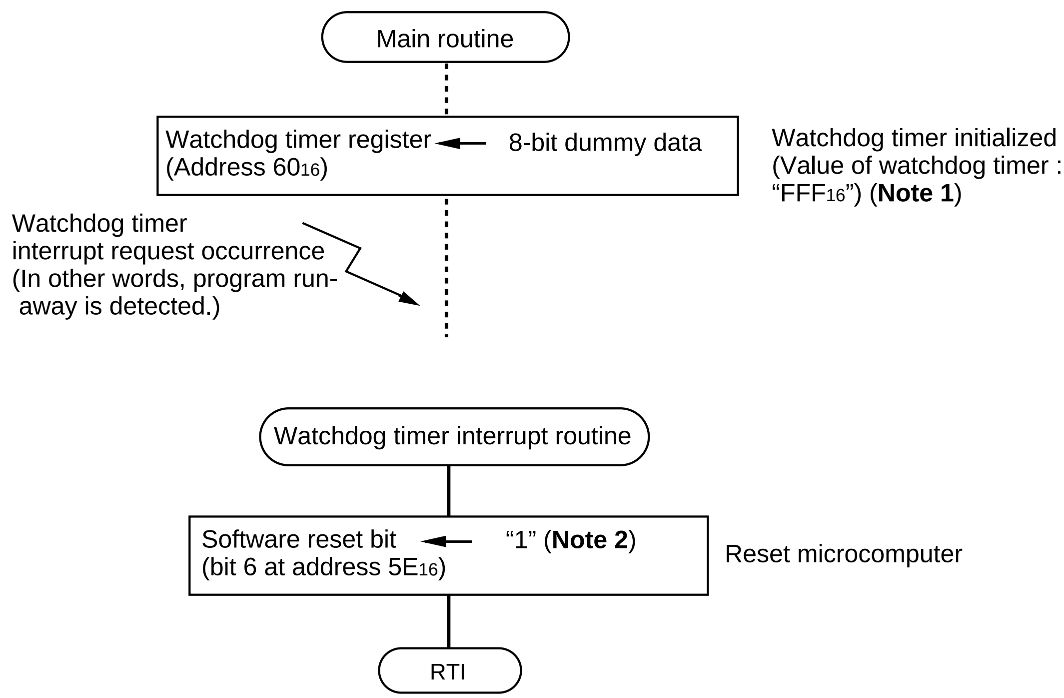
Table 14.2.1 Occurrence interval of watchdog timer interrupt request

Watchdog timer frequency select bit	f(f_{sys}) = 20 MHz	
	Count source	Occurrence interval (Note)
0	Wf_{512}	52.43 ms
1	Wf_{32}	3.28 ms

Note: This applies when the peripheral device’s clock select bits 1, 0 (bits 7, 6 at address BC_{16}) = “00_2.”

14.2 Operation description

Be sure to write dummy data to the watchdog timer register (address 60_{16}) before the most significant bit of the watchdog timer becomes "0." When writing to the watchdog timer is not performed owing to a program runaway and the watchdog timer's most significant bit becomes "0," a watchdog timer interrupt request occurs. This informs that a program runaway has occurred.
In order to reset the microcomputer when a program runaway has been detected, write "1" to the software reset bit (bit 6 at address $5E_{16}$) in the watchdog timer interrupt routine.
Figure 14.2.1 shows an example of a program runaway detected by the watchdog timer.



Notes 1: Be sure to initialize the watchdog timer before the most significant bit of the watchdog timer becomes "0." (In other words, be sure to write dummy data to address 60_{16} before a watchdog timer interrupt request occurs).
2: When a program runaway occurs, values of the data bank register (DT), direct page register (DPRi), etc., may be changed. When "1" is written to the software reset bit by an addressing mode using DT, DPRi, etc., be sure to set values to DT and DPRi, etc. again.

Fig. 14.2.1 Example of program runaway detection by watchdog timer

14.2.2 Stop period

The watchdog timer stops its operation in any of the following cases:

① During Wait mode (Refer to section **"15.4 Wait mode."**)

② During Stop mode (Refer to section **"15.3 Stop mode."**)

When state ① has been terminated, the watchdog timer restarts counting from the state immediately before it stops its operation. For the watchdog timer's operation at termination of state ②, refer to section **"14.2.3 Operation in stop mode."**

14.2.3 Operations in stop mode

When the **STP** instruction has been executed, a value of "FFF_{16}" is set to the watchdog timer, and the watchdog timer stops its operation in the stop mode. Immediately after the stop mode termination, the watchdog timer operates as follows.

(1) When stop mode is terminated by hardware reset

Supply of ϕ_{CPU} and ϕ_{BIU} starts immediately after the stop mode termination, and the microcomputer performs "operation after reset." (Refer to **"CHAPTER 3. RESET."**) The watchdog timer frequency select bit becomes "0," and the watchdog timer starts counting of Wf_{512} from "FFF_{16}."

(2) When stop mode is terminated by interrupt occurrence (with watchdog timer used) (Note)

Immediately after the stop mode termination, the watchdog timer starts counting the count source selected by the watchdog timer clock source select bits at STP termination (bits 6, 7 at address 61_{16}), starting from "FFF_{16}." It is independent of the watchdog timer frequency select bit (bit 0 at address 61_{16}). When the most significant bit of the watchdog timer becomes "0," supply of ϕ_{CPU} and ϕ_{BIU} starts. (At this time, no watchdog timer interrupt request occurs.)

When supply of ϕ_{CPU} and ϕ_{BIU} starts, the routine of the interrupt which the microcomputer used to terminate the stop mode is executed. The watchdog timer restarts counting of the count source (Wf_{32} or Wf_{512}), which was counted immediately before execution of the **STP** instruction, starting from "FFF_{16}."

Note: For the setting of the usage of the watchdog timer, refer to section **"15.3 Stop mode."**

(3) When stop mode is terminated by interrupt occurrence (with watchdog timer not used) (Note)

Supply of ϕ_{CPU} and ϕ_{BIU} starts immediately after the stop mode termination, and the routine of the interrupt which the microcomputer used to terminate the stop mode is executed. The watchdog timer restarts counting of the count source (Wf_{32} or Wf_{512}), which was counted immediately before execution of the **STP** instruction, starting from "FFF_{16}."

Note: For the setting of the usage of the watchdog timer, refer to section **"15.3 Stop mode."**

[Precautions for watchdog timer]

1. When dummy data has been written to address 60_{16} with the 16-bit data length, writing to address 61_{16} is simultaneously performed. Accordingly, when the user does not want to change the contents of the watchdog timer frequency select bit (bit 0 at address 61_{16}) and watchdog timer clock source select bits at STP termination (bits 6, 7 at address 61_{16}), be sure to write again the values which are currently set in these bits, simultaneously with writing to address 60_{16}.

2. When the **STP** instruction is executed, the watchdog timer stops its operation. If the **STP** instruction's code (31_{16}, 30_{16}) has accidentally been executed owing to a program runaway, the watchdog timer stops its operation. Therefore, in the system where the watchdog timer is used to detect a program runaway, we recommend that the STP instruction invalidity select bit (bit 0 at address 62_{16}) = "1." (**STP** instruction is invalid.) Refer to section **"15.3 Stop mode."**

CHAPTER 15

STOP AND WAIT MODES

15.1 Overview

When there is no need for operation of the central processing unit (CPU), the stop and wait modes are used to stop oscillation or internal clock. As a result, the power consumption can be saved. The microcomputer enters the stop mode when the **STP** instruction has been executed; the microcomputer enters the wait mode when the **WIT** instruction has been executed.
The stop and wait modes are terminated by an interrupt request occurrence or hardware reset.
Table 15.1.1 lists the states in the stop and wait modes and operations after these modes are terminated.

Table 15.1.1 States in stop and wait modes and operations after these modes are terminated

Item			Stop mode		Wait mode	
			When watchdog timer is used at termination (See Figure 15.3.1.)	When watchdog timer is not used at termination (See Figure 15.3.1.)	System clock is active. (Bit 3 at address 63_{16} = "0")	System clock is inactive. (Bit 3 at address 63_{16} = "1")
States		Oscillation	Inactive.		Active.	
		PLL frequency multiplier	Stopped.		Operates **(Note 1).**	
		ϕ_{CPU}, ϕ_{BIU}	Inactive.		Inactive.	
		f_{sys}, clock ϕ_1, f_1 to f_{4096}	Inactive.		Active.	Inactive.
		Wf_{32}, Wf_{512}	Inactive.		Inactive.	
	Internal peripheral	Timers A, B	Can operate only in the event counter mode.		Operates.	Can operate only in the event counter mode.
		Serial I/O	Can operate only when an external clock is selected.		Operates.	Can operate only when an external clock is selected.
		A-D converter	Stopped.		Operates.	Stopped.
		D-A converter	Stopped.		Operates.	Stopped.
		Watchdog timer	Stopped.		Stopped.	
		Pins	Retains the state at the **STP** instruction execution.		Retains the state at the **WIT** instruction execution.	
Operation after termination		Termination due to interrupt request occurrence	Supply of ϕ_{CPU}, ϕ_{BIU} starts after a certain time has been measured by using the watchdog timer.	Supply of ϕ_{CPU}, ϕ_{BIU} starts immediately after termination **(Note 2)**.	Supply of ϕ_{CPU}, ϕ_{BIU} starts immediately after termination.	
		Termination due to hardware reset	Operation after hardware reset		Operation after hardware reset	

Notes 1: This applies when the PLL circuit operation enable bit (bit 1 at address BC_{16}) = "1."
2: See Table 15.3.2.

15.2 Block description

Figure 15.2.1 shows the block diagram of the clock generating circuit with the **STP** and **WIT** instructions. Also, registers relevant to these modes are described below.

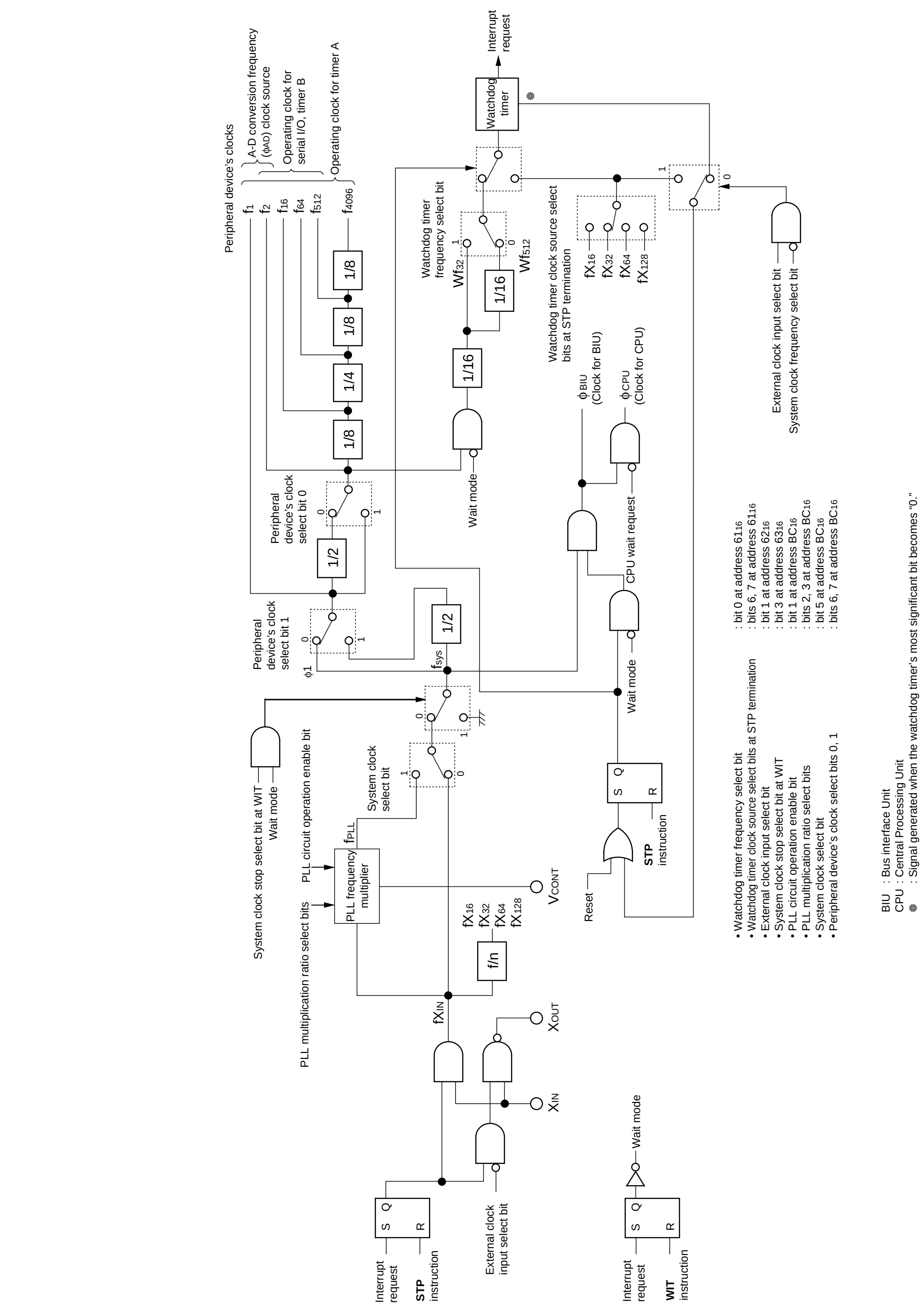


Fig. 15.2.1 Block diagram of clock generating circuit with STP and WIT instructions

15.2 Block description

15.2.1 Particular function select register 0

Figure 15.2.2 shows the structure of the particular function select register 0, and Figure 15.2.3 shows the writing procedure for the particular function select register 0.

Particular function select register 0 (Address 62_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0		

Bit	Bit name	Function	At reset	R/W
0	STP instruction invalidity select bit	0 : STP instruction is valid. 1 : STP instruction is invalid.	0	RW **(Note)**
1	External clcok input select bit	0 : Oscillation circuit is active. (Oscillator is connected.) Watchdog timer is used at stop mode termination. 1 : Oscillation circuit is inactive. (External clock is input.) When the system clock select bit (bit 5 at address BC_{16}) = “0,” watchdog timer is not used at stop mode termination. When the system clock select bit = “1,” watchdog timer is used at stop mode termination.	0	RW **(Note)**
7 to 2	Fix these bits to “000000.”		0	RW

Note: Writing to these bits requires the following procedure:
- Write “55_{16}” to this register. (The bit status does not change only by this writing.)
- Succeedingly, write “0” or “1” to each bit.

Also, use the **MOVMB** (**MOVM** when m = 1) instruction or **STAB** (**STA** when m = 1) instruction.
If an interrupt occurs between writing of “55_{16}” and next writing of “0” or “1,” latter writing may be ignored. When there is a possibility that an interrupt occurs at the above timing, be sure to read this bit’s contents after writing of “0” or “1,” and verify whether “0” or “1” has correctly been written or not.

Fig. 15.2.2 Structure of particular function select register 0

(1) STP instruction invalidity select bit (bit 0)

Setting this bit to “1” invalidates the **STP** instruction. When using the stop mode, be sure to clear this bit to “0.”

Writing to this bit requires the following procedure:
- Write “55_{16}” to address 62_{16}.
- Succeedingly, write “0” or “1” to this bit. (See Figure 15.2.3.)

If an interrupt occurs between writing of “55_{16}” and next writing of “0” or “1,” latter writing may be ignored. When there is a possibility that an interrupt occurs at the above timing, be sure to read this bit’s contents after writing of “0” or “1,” and verify whether “0” or “1” has correctly been written or not.

(2) External clock input select bit (bit 1)

When this bit = "0," the oscillation driver circuit between pins X_{IN} and X_{OUT} is operating. At the stop mode termination owing to an interrupt occurrence, the watchdog timer is used.
Setting this bit to "1" stops the oscillation driver circuit between pins X_{IN} and X_{OUT} and keeps the output level at pin X_{OUT} being "H." (Refer to section **"16.3 Stop of oscillation circuit."**) At the stop mode termination owing to an interrupt occurrence, the watchdog timer is not used if the system clock select bit (bit 5 at address BC_{16}) = "0," where as the watchdog timer is used if the system clock select bit = "1."
To rewrite this bit, write "0" or "1" just after writing of "55_{16}" to address 62_{16}. (See Figure 15.2.3.)
Note that if an interrupt occurs between writing of "55_{16}" and next writing of "0" or "1," latter writing may be ignored. When there is a possibility that an interrupt occurs at the above timing, be sure to read this bit's contents after writing of "0" or "1," and verify whether "0" or "1" has correctly been written or not.

In addition, even when the watchdog timer is disabled by the particular function select register 2 (address 64_{16}), the watchdog timer can be active only at the stop mode termination if this bit = "0." (Refer to section **"15.3 Stop mode."**)

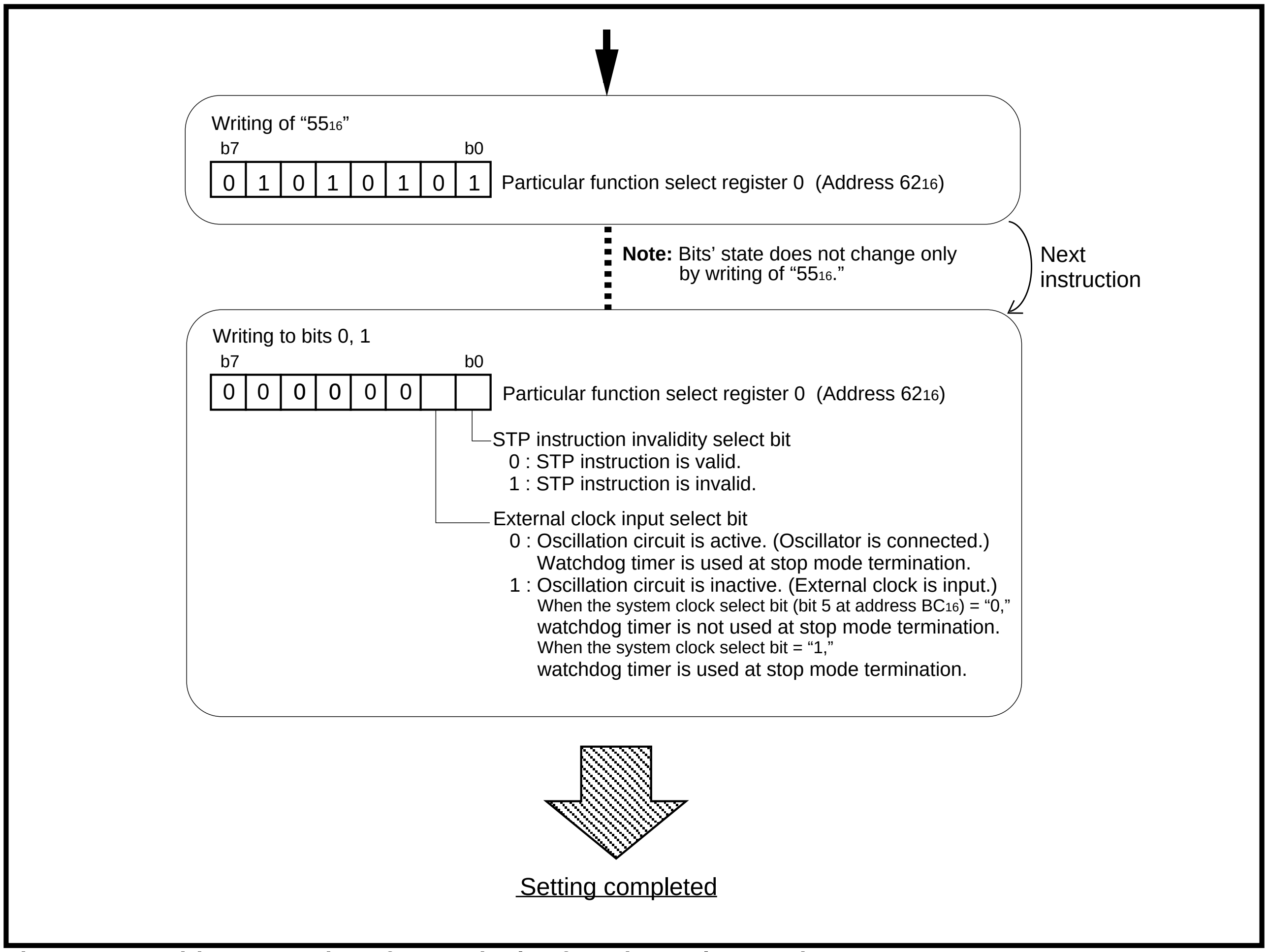


Fig. 15.2.3 Writing procedure for particular function select register 0

15.2.2 Particular function select register 1

Figure 15.2.4 shows the structure of the particular function select register 1.

Particular function select register 1 (Address 63_{16})

b7	b6	b5	b4	b3	b2	b1	b0
			0		0		

Bit	Bit name	Function	At reset	R/W
0	STP-instruction-execution status bit	0 : Normal operation. 1 : During execution of STP instruction	**(Note 1)**	RW **(Note 2)**
1	WIT-instruction-execution status bit	0 : Normal operation. 1 : During execution of WIT instruction	**(Note 1)**	RW **(Note 2)**
2	Fix this bit to “0.”		0	RW
3	System clock stop select bit at WIT **(Note 3)**	0 : In the wait mode, system clock f_{sys} is active. 1 : In the wait mode, system clock f_{sys} is inactive.	0	RW
4	Fix this bit to “0.”		0	RW
5	The value is “0” at reading.		0	—
6	Timer B2 clock source select bit (Valid in event counter mode.) **(Note 4)**	0 : External signal input to the $TB2_{IN}$ pin is counted. 1 : fX_{32} is counted.	0	RW
7	The value is “0” at reading.		0	—

Notes 1: At power-on reset, this bit becomes “0.” At hardware reset or software reset, this bit retains the value just before reset.
2: Even when “1” is written, the bit status will not change.
3: Setting this bit to “1” must be performed just before execution of the WIT instruction. Also, after the wait state is terminated, this bit must be cleared to “0” immediately.
4: When using timer B2 in the pulse period/pulse width measurement mode, be sure to clear this bit to “0.”

Fig. 15.2.4 Structure of particular function select register 1

(1) STP-instruction-execution status bit (bit 0)

When the microcomputer enters the stop mode, this bit becomes “1,” indicating that the **STP** instruction has been executed.

This bit becomes “0” at power-on reset. At hardware reset and software reset, this bit retains the value immediately before reset. Therefore, this bit is used for the following verification:
• Which of the power-on reset and hardware reset has been used to reset the system?
• Has the hardware reset been used for the stop mode termination?

This bit is cleared to “0” by writing “0” to this bit. Although, even when “1” is written to this bit, this bit does not change.

At the stop mode termination, be sure to clear this bit to “0” by software.

(2) WIT-instruction-execution status bit (bit 1)

When the microcomputer enters the wait mode, this bit becomes “1,” indicating that the **WIT** instruction has been executed.

This bit becomes “0” at power-on reset. At hardware reset and software reset, this bit retains the value immediately before reset. Therefore, this bit is used for the following verification:
• Which of the power-on reset and hardware reset has been used to reset the system?
• Has the hardware reset been used for the wait mode termination?

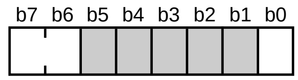
This bit is cleared to “0” by writing “0” to this bit. Although, even when “1” is written to this bit, this bit does not change.

At the wait mode termination, be sure to clear this bit to “0” by software.

15.2.3 Watchdog timer frequency select register

Figure 15.2.5 shows the structure of the watchdog timer frequency select register.

Watchdog timer frequency select register (Address 61_{16})



Bit	Bit name	Function	At reset	R/W
0	Watchdog timer frequency select bit	0 : Wf_{512} 1 : Wf_{32}	0	RW
5 to 1	Nothing is assigned.		Undefined	—
6	Watchdog timer clock source select bits at STP termination	b7 b6 0 0 : fX_{32} 0 1 : fX_{16} 1 0 : fX_{128} 1 1 : fX_{64}	0	RW
7			0	RW

Fig. 15.2.5 Structure of watchdog timer frequency select register

(1) Watchdog timer clock source select bits at STP termination (bits 7, 6)

These bits are used to select a count source at stop mode termination.

For details of the operation at stop mode termination, refer to section **"15.3 Stop mode."**

15.3 Stop mode

When the **STP** instruction has been executed, each of the oscillation and the PLL frequency multiplier's operation becomes inactive. This state is called "stop mode." (See Table 15.1.1)
In the stop mode, even when oscillation becomes inactive, the contents of the internal RAM can be retained if Vcc (the power source voltage) ≥ V_{RAM} (RAM hold voltage). Furthermore, since the CPU and internal peripheral devices which use any of clocks f_1 to f_{4096}, Wf_{32}, Wf_{512} stop their operations, the power consumption can be saved.
The stop mode is terminated owing to an interrupt request occurrence or hardware reset.

When terminated owing to an interrupt request occurrence, an instruction can be executed immediately after termination if all of the following conditions are satisfied. (Refer to section **"15.3.2 Terminate operation at interrupt request occurrence (when not using watchdog timer)."**):
• An stable clock is input from the external. (The external clock input select bit (bit 1 at address 62_{16}) = "1.")
• The PLL frequency multiplier is not used. (The system clock select bit (bit 5 at address BC_{16}) = "0.")

When terminated owing to an interrupt request occurrence, an instruction will be executed after the oscillation stabilizing time has been measured by using the watchdog timer if any of the following conditions is satisfied. (Refer to section **"15.3.1 Terminate operation at interrupt request occurrence (when using watchdog timer)."**):
• An oscillator is used. (The external clock input select bit (bit 1 at address 62_{16}) = "0.")
• The PLL frequency multiplier is used. (The system clock select bit (bit 5 at address BC_{16}) = "1.")

15.3.1 Terminate operation at interrupt request occurrence (when using watchdog timer)

At the stop mode termination, execution of an instruction is started after a certain time has been measured by using the watchdog timer. (See Figure 15.3.1.)

① When an interrupt request occurs, an oscillator starts its operation. Also, when the PLL circuit operation enable bit (bit 1 at address BC_{16}) = "1," the PLL frequency multiplier starts its operation. Simultaneously with this, each supply of clocks f_{sys}, ϕ_1, f_1 to f_{4096}, Wf_{32}, Wf_{512} starts.
② By start of oscillation in ①, the watchdog timer starts its operation. Regardless of the watchdog timer frequency select bit (bit 0 at address 61_{16}), the watchdog timer counts a count source (fX_{16} to fX_{128}), which is selected by the watchdog timer clock source select bits at STP termination (bits 7, 6 at address 61_{16}). This counting is started from a value of "FFF_{16}."
③ When the most significant bit (MSB) of the watchdog timer becomes "0," each supply of ϕ_{CPU}, ϕ_{BIU} starts. (At this time, no watchdog timer interrupt request occurs.) Also, the count source of the watchdog timer returns to the count source selected by the watchdog timer frequency select bits (in order words, Wf_{32} or Wf_{512}).
④ The interrupt request which occurred in ① is accepted.

For the watchdog timer, refer to **"CHAPTER 14. WATCHDOG TIMER."**
Table 15.3.1 lists the interrupts which can be used to terminate the stop mode.

Table 15.3.1 Interrupts which can be used to terminate stop mode

Interrupt	Usage condition for interrupt request occurrence
$\overline{INT_i}$ interrupt (i = 0 to 7)	————
Timer Ai interrupt (i = 0 to 9)	In event counter mode
Timer Bi interrupt (i = 0 to 2)	
UARTi transmit interrupt (i = 0 to 2)	When an external clock is selected.
UARTi receive interrupt (i = 0 to 2)	

Notes 1: When multiple interrupts are enabled, the stop mode is terminated owing to the interrupt request which occurs first.
2: For interrupts, refer to **"CHAPTER 6. INTERRUPTS"** and each peripheral device's chapter.

Before executing the **STP** instruction, be sure to enable an interrupt which is to be used for the stop mode termination.
Also, make sure that the interrupt priority level of an interrupt, which is to be used for the termination, is higher than the processor interrupt priority level (IPL) of a routine where the **STP** instruction is executed.
After oscillation starts (①), there is a possibility that each interrupt request occurs until the supply of ϕ_{CPU}, ϕ_{BIU} starts (③). The interrupt requests which occurred during this period are accepted in order of priority after the watchdog timer's MSB becomes "0." (When the level sense of an $\overline{INT_i}$ interrupt is used, however, no interrupt request is retained. Therefore, if pin $\overline{INT_i}$ is at the invalid level when the watchdog timer's MSB becomes "0," no interrupt request is accepted.) For an interrupt which has no need to be accepted, be sure to set its interrupt priority level to "0" (Interrupt disabled) before executing the **STP** instruction.

15.3.2 Terminate operation at interrupt request occurrence (when not using watchdog timer)

At the stop mode termination, an instruction is executed without use of the watchdog timer. (See Figure 15.3.1.)

① When an interrupt request occurs, clock input from pin X_{IN} starts. Simultaneously, supply of clocks f_{sys}, ϕ_1, f_1 to f_{4096}, Wf_{32}, Wf_{512} starts.
② Supply of ϕ_{CPU}, ϕ_{BIU} starts after the time listed in Table 15.3.2 has elapsed.
③ The interrupt request which occurred in ① is accepted.

Table 15.3.2 Time after stop mode is terminated until supply of ϕ_{CPU}, ϕ_{BIU} starts

Watchdog timer clock source select bits at STP termination (bits 7, 6 at address 61_{16})	Time until supply of ϕ_{CPU} and ϕ_{BIU} starts
00	fX_{IN} × 19 cycles
01	fX_{IN} × 11 cycles
10	fX_{IN} × 67 cycles
11	fX_{IN} × 35 cycles

Before executing the **STP** instruction, be sure to set as follows:

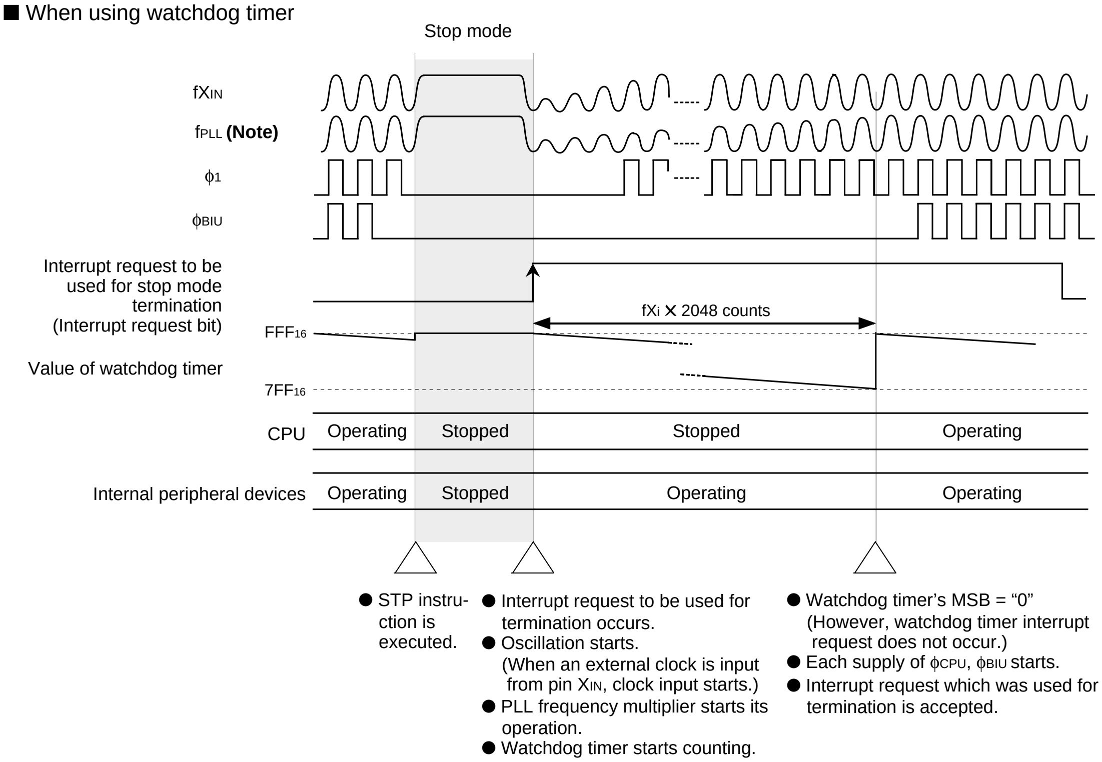
- Enable an interrupt which is to be used for the stop mode termination.
  Also, make sure that the interrupt priority level of an interrupt, which is to be used for the termination, is higher than the processor interrupt priority level (IPL) of a routine where the **STP** instruction is executed.

- The external clock input select bit (bit 1 at address 62_{16}) = "1" **(Note)**

- The system clock select bit (bit 5 at address BC_{16}) = "0" **(Note)**

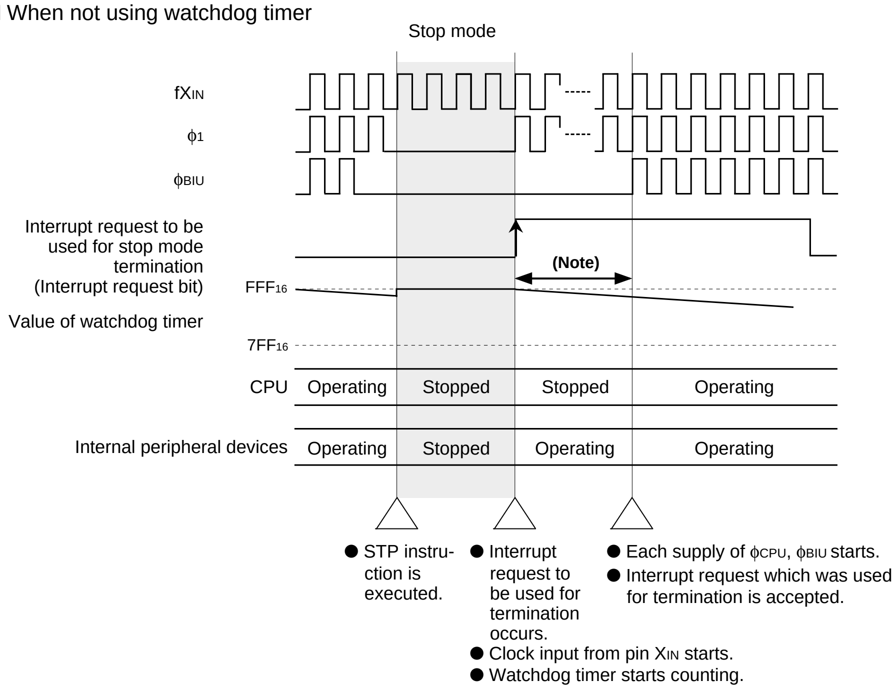
Note: Simultaneously, the oscillation driver circuit between pins X_{IN} and X_{OUT} stops, and the output level at pin X_{OUT} is kept "H." (Refer to section **"16.3 Stop of oscillation circuit."**)

■ When using watchdog timer



Note: This applies when the PLL circuit operation enable bit (bit 1 at address BC_{16}) = "1."
fX_i : fX_{16}, fX_{32}, fX_{64}, fX_{128}.
These are clocks selected by the watchdog timer clock source select bits at STP termination (bits 7, 6 at address 61_{16}.)

■ When not using watchdog timer



Note: Time listed in Table 15.3.2. See Figure 19.1.3 for the built-in flash memory version.

Fig. 15.3.1 Stop mode terminate sequence owing to interrupt request occurrence

15.3.3 Terminate operation at hardware reset

Although each of the CPU and SFR area is initialized, the contents of the internal RAM immediately before the **STP** instruction execution are retained. The terminate sequence is the same as the internal processing sequence after reset.

For reset, refer to **"CHAPTER 3. RESET."**

Also, the STP-instruction-execution status bit (bit 0 at address 63_{16}) is used for the following verification:

- Which of the power-on reset and hardware reset has been used to reset the system?
- Has the hardware reset been used for the stop mode termination?

15.4 Wait mode

When the **WIT** instruction is executed, both of ϕ_{CPU} and ϕ_{BIU} become inactive. (The oscillation does not become inactive.) This state is called "wait mode." (See Table 15.1.1.)
In the wait mode, the power consumption can be saved with Vcc (the power source voltage) retained. When using no internal peripheral device in the wait mode, the power consumption can be saved furthermore since each of f_{sys} and internal peripheral device's operation clock can be inactive. (Refer to section **"16.2 Stop of system clock in wait mode."**)
The wait mode is terminated owing to an interrupt request occurrence or hardware reset.
The wait mode terminate operation is described below.

15.4.1 Terminate operation at interrupt request occurrence

① When an interrupt request occurs, each supply of ϕ_{CPU} and ϕ_{BIU} starts.
② The interrupt request which occurred in ① is accepted.

Table 15.4.1 lists the interrupts which can be used for the wait mode termination.

Table 15.4.1 Interrupts which can be used for wait mode termination

Interrupt	Usage conditions for interrupt request occurrences	
	System clock in action	System clock out of action
$\overline{INT_i}$ interrupt (i = 0 to 7)	————	
Timer Ai interrupt (i = 0 to 9)	————	In event counter mode
Timer Bi interrupt (i = 0 to 2)		
UARTi transmit interrupt (i = 0 to 2)	————	When an external clock is selected.
UARTi receive interrupt (i = 0 to 2)		
A-D conversion interrupt	————	Do not use.

Notes 1: When multiple interrupts are enabled, the wait mode is terminated owing to the interrupt request which occurs first.
2: For interrupts, refer to **"CHAPTER 6. INTERRUPTS"** and each peripheral device's chapter.

Before executing the **WIT** instruction, be sure to enable an interrupt which is to be used for the wait mode termination.
Also, make sure that the interrupt priority level of an interrupt, which is to be used for termination, is higher than the processor interrupt priority level (IPL) of a routine where the **WIT** instruction is executed.
Also, when multiple interrupts in Table 15.4.1 are enabled, the wait mode is terminated owing to the interrupt request which occurs first.

15.4.2 Terminate operation at hardware reset

Although each of the CPU and SFR area is initialized, the contents of the internal RAM immediately before the **WIT** instruction execution are retained. The terminate sequence is the same as the internal processing sequence after reset.
For reset, refer to **"CHAPTER 3. RESET."**
Also, the WIT-instruction-execution status bit (bit 1 at address 63_{16}) is used for the following verification:
• Which of the power-on reset and hardware reset has been used to reset the system?
• Has the hardware reset been used for the wait mode termination?

CHAPTER 16

POWER SAVING FUNCTIONS

POWER SAVING FUNCTIONS

16.1 Overview

This chapter explains the functions to save the power consumption of the microcomputer and the total system including the microcomputer.

16.1 Overview

Table 16.1.1 lists the overview of the power saving functions. Each of these functions saves the power consumption of the total system. The registers related to the power saving functions are explained in the following.

Table 16.1.1 Overview of power saving functions

Item	Function	Reference
Inactivity of system clock in wait mode	In the wait mode, operating clocks for the internal peripheral devices and f_{sys} can be inactive.	**CHAPTER 15. STOP AND WAIT MODES**
Stop of oscillation circuit	When a stable clock externally generated is used, the drive circuit for oscillation between pins X_{IN} and X_{OUT} can be stopped. (The output level at pin X_{OUT} is fixed to "H.")	**CHAPTER 4. CLOCK GENERATING CIRCUIT**, Section **15.3 Stop mode**
Pin V_{REF} disconnection	The V_{REF} input can be disconnected when the A-D converter is not used	**CHAPTER 12. A-D CONVERTER**

16.1.1 Particular function select register 0

Figure 16.1.1 shows the structure of the particular function select register 0, and Figure 16.1.2 shows the writing procedure for the particular function select register 0.

Particular function select register 0 (Address 62_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0		

Bit	Bit name	Function	At reset	R/W
0	STP instruction invalidity select bit	0 : STP instruction is valid. 1 : STP instruction is invalid.	0	RW **(Note)**
1	External clcok input select bit	0 : Oscillation circuit is active. (Oscillator is connected.) Watchdog timer is used at stop mode termination. 1 : Oscillation circuit is inactive. (External clock is input.) When the system clock select bit (bit 5 at address BC_{16}) = "0," watchdog timer is not used at stop mode termination. When the system clock select bit = "1," watchdog timer is used at stop mode termination.	0	RW **(Note)**
7 to 2	Fix these bits to "000000."		0	RW

Note: Writing to these bits requires the following procedure:
- Write "55_{16}" to this register. (The bit status does not change only by this writing.)
- Succeedingly, write "0" or "1" to each bit.

Also, use the **MOVMB** (**MOVM** when m = 1) instruction or **STAB** (**STA** when m = 1) instruction.
If an interrupt occurs between writing of "55_{16}" and next writing of "0" or "1," latter writing may be ignored. When there is a possibility that an interrupt occurs at the above timing, be sure to read this bit's contents after writing of "0" or "1," and verify whether "0" or "1" has correctly been written or not.

Fig. 16.1.1 Structure of particular function select register 0

16.1 Overview

(1) External clock input select bit (bit 1)

When this bit = “0,” the oscillation driver circuit between pins X_{IN} and X_{OUT} is operationg. Also, at the stop mode termination owing to an interrupt request occurrence, the watchdog timer is used.
Setting this bit to “1” stops the oscillation driver circuit between pins X_{IN} and X_{OUT} and keeps the output level at pin X_{OUT} being “H.” (Refer to section **“16.3 Stop of oscillation circuit.”**) At the stop mode termination owing to an interrupt request occurrence, the watchdog timer is not used if the system clock select bit (bit 5 at address BC_{16}) = “0,” where as the watchdog timer is used if the system clock select bit = “1.”
To rewrite this bit, write “0” or “1” just after writing of “55_{16}” to address 62_{16}. (See Figure 16.1.2.)
Note that if an interrupt occurs between writing of “55_{16}” and next writing of “0” or “1,” latter writing may be ignored. When there is a possibility that an interrupt occurs at the above timing, be sure to read this bit’s contents after writing of “0” or “1,” and verify whether “0” or “1” has correctly been written or not.

In addition, even when the watchdog timer is disabled by the particular function select register 2 (address 64_{16}), the watchdog timer can be active only at the stop mode termination if this bit = “0.” (Refer to section **“15.3 Stop mode.”**)

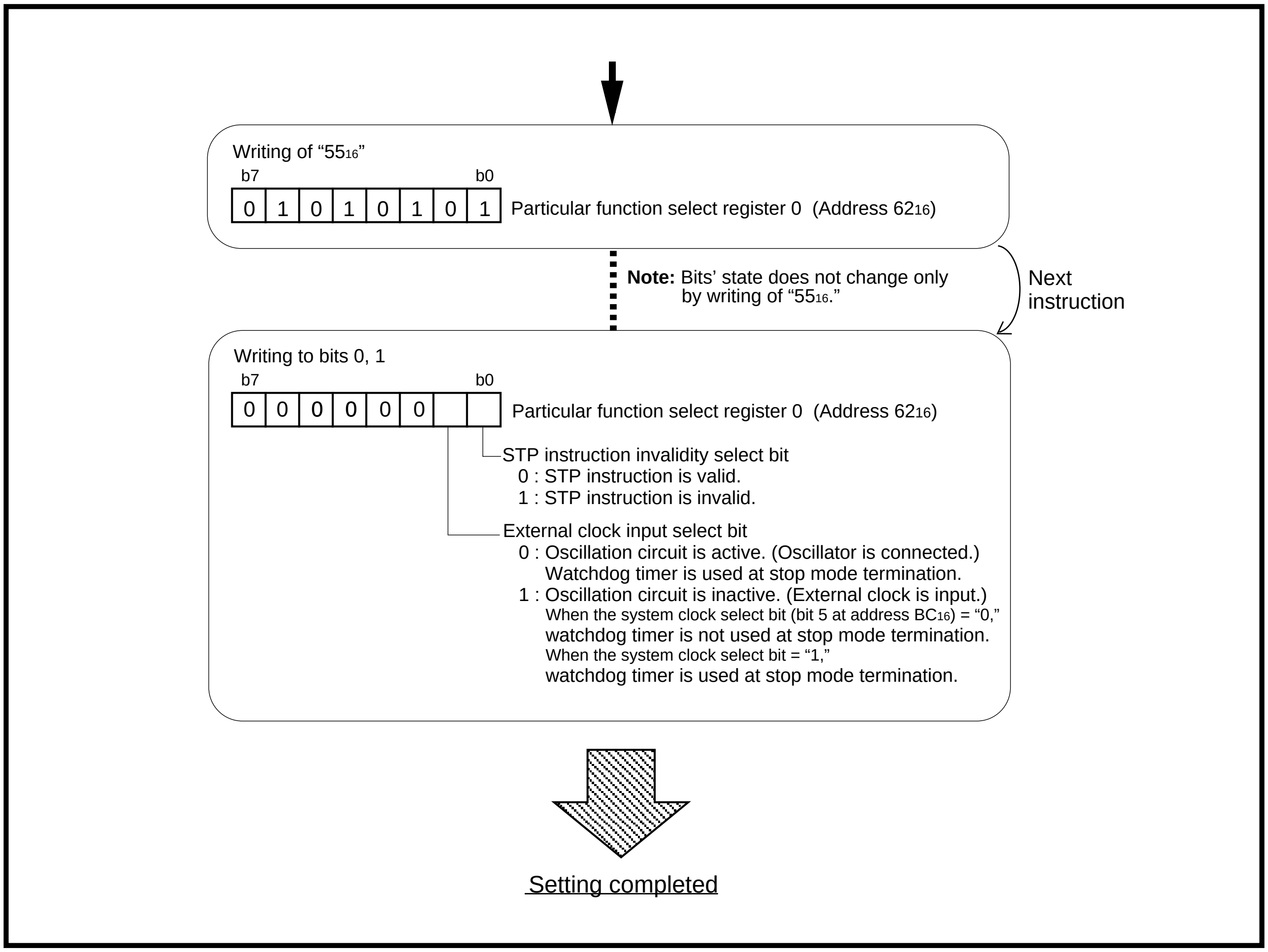


Fig. 16.1.2 Writing procedure for particular function select register 0

16.1.2 Particular function select register 1

Figure 16.1.3 shows the structure of the particular function select register 1.

Particular function select register 1 (Address 63_{16})

b7	b6	b5	b4	b3	b2	b1	b0
			0		0		

Bit	Bit name	Function	At reset	R/W
0	STP-instruction-execution status bit	0 : Normal operation. 1 : During execution of STP instruction	**(Note 1)**	RW **(Note 2)**
1	WIT-instruction-execution status bit	0 : Normal operation. 1 : During execution of WIT instruction	**(Note 1)**	RW **(Note 2)**
2	Fix this bit to "0."		0	RW
3	System clock stop select bit at WIT **(Note 3)**	0 : In the wait mode, system clock f_{sys} is active. 1 : In the wait mode, system clock f_{sys} is inactive.	0	RW
4	Fix this bit to "0."		0	RW
5	The value is "0" at reading.		0	—
6	Timer B2 clock source select bit (Valid in event counter mode.) **(Note 4)**	0 : External signal input to the $TB2_{IN}$ pin is counted. 1 : fX_{32} is counted.	0	RW
7	The value is "0" at reading.		0	—

Notes 1: At power-on reset, this bit becomes "0." At hardware reset or software reset, this bit retains the value just before reset.
2: Even when "1" is written, the bit status will not change.
3: Setting this bit to "1" must be performed just before execution of the WIT instruction. Also, after the wait state is terminated, this bit must be cleared to "0" immediately.
4: When using timer B2 in the pulse period/pulse width measurement mode, be sure to clear this bit to "0."

Fig. 16.1.3 Structure of particular function select register 1

(1) System clock stop select bit at WIT (bit 3)

Setting this bit to "1" makes the following clocks inactive in the wait mode: the operating clocks for the internal peripheral devices and f_{sys}. (Refer to section **"16.2 Inactivity of system clock in wait mode."**)

16.2 Inactivity of system clock in wait mode

In the wait mode, if there is not need to operate the internal peripheral devices, setting the system clock stop select bit at WIT (See Figure 16.1.3.) to “1” makes the following clocks inactive: the operating clocks for the internal peripheral devices and f_{sys}. This saves the power consumption of the microcomputer. Table 16.2.1 lists the states and operations in the wait mode and after this mode is terminated.

Table 16.2.1 States and operations in wait mode and after this mode is terminated

<table>
<tr><th colspan="3">Item</th><th>System clock is active. (bit 3 at address 63_{16} = 0)</th><th>System clock is inactive. (bit 3 at address 63_{16} = 1)</th></tr>
<tr><td rowspan="11">States</td><td colspan="2">Oscillation</td><td colspan="2">Active.</td></tr>
<tr><td colspan="2">PLL frequency multiplier</td><td colspan="2">Operates (Note).</td></tr>
<tr><td colspan="2">ϕ_{CPU}, ϕ_{BIU}</td><td colspan="2">Inactive.</td></tr>
<tr><td colspan="2">f_{sys}, Clock ϕ_1,
f_1 to f_{4096}</td><td>Active.</td><td>Inactive.</td></tr>
<tr><td colspan="2">Wf_{32}, Wf_{512}</td><td colspan="2">Inactive.</td></tr>
<tr><td rowspan="6">Internal peripheral devices</td><td>Timers A, B</td><td>Operates.</td><td>Can operate only in the event counter mode.</td></tr>
<tr><td>Serial I/O</td><td>Operates.</td><td>Can operate only when an external clock is selected.</td></tr>
<tr><td>A-D converter</td><td>Operates.</td><td>Stopped.</td></tr>
<tr><td>D-A converter</td><td>Operates.</td><td>Stopped.</td></tr>
<tr><td>Watchdog timer</td><td colspan="2">Stopped.</td></tr>
<tr><td>Pins</td><td colspan="2">Retains the state at the WIT instruction execution.</td></tr>
<tr><td rowspan="2">Operation after termination</td><td colspan="2">Termination due to interrupt request occurrence</td><td colspan="2">Supply of ϕ_{CPU}, ϕ_{BIU} starts immediately just after termination.</td></tr>
<tr><td colspan="2">Termination due to hardware reset</td><td colspan="2">Operation after hardware reset</td></tr>
</table>

Note: This applies when the PLL circuit operation enable bit (bit 1 at address BC_{16}) = “1.”

16.3 Stop of oscillation circuit

When a stable clock externally generated is input to pin X_{IN}, power consumption can be saved by setting the external clock input select bit to "1" to stop the drive circuit for oscillation between pins X_{IN} and X_{OUT}. (See Figure 16.1.1.) At this time, the output level at pin X_{OUT} is fixed to "H." Also, if the system clock select bit (bit 5 at address BC_{16}) = "0," the watchdog timer is not used when the stop mode is terminated owing to an interrupt request occurrence; therefore, the microcomputer can start instruction execution just after termination of the stop mode. When the system clock select bit = "1," in this case, the watchdog timer is used.

16.4 Pin V_{REF} disconnection

When the A-D converter is not used, power consumption can be saved by setting the V_{REF} connection select bit (See Figure 16.4.1) to "1." It is because the reference voltage input pin (V_{REF}) is disconnected from the ladder resistors of the A-D converter, and there is no current flow between them.
When the V_{REF} connection select bit has been cleared from "1" (V_{REF} disconnected) to "0" (V_{REF} connected), be sure to start the A-D conversion after an interval of 1 µs or more has elapsed.

A-D control register 1 (Address $1F_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Bit	Bit name	Function	At reset	R/W
0	A-D sweep pin select bits (Valid in the single sweep mode, repeat sweep mode 0, and repeat sweep mode 1.) **(Note 1)**	Single sweep mode/Repeat sweep mode 0 b1 b0 0 0 : Pins AN_0 and AN_1 (2 pins) 0 1 : Pins AN_0 to AN_3 (4 pins) 1 0 : Pins AN_0 to AN_5 (6 pins) 1 1 : Pins AN_0 to AN_7 (8 pins) **(Note 2)**	1	RW
1		Repeat sweep mode 1 **(Note 3)** b1 b0 0 0 : Pin AN_0 (1 pin) 0 1 : Pins AN_0 and AN_1 (2 pins) 1 0 : Pins AN_0 to AN_2 (3 pins) 1 1 : Pins AN_0 to AN_3 (4 pins)	1	RW
2	A-D operation mode select bit 1 (Used in the repeat sweep mode 0 and repeat sweep mode 1.) **(Note 4)**	0 : Repeat sweep mode 0 1 : Repeat sweep mode 1	0	RW
3	Resolution select bit	0 : 8-bit resolution mode 1 : 10-bit resolution mode	0	RW
4	A-D conversion frequency (ϕ_{AD}) select bit 1	See Table 12.2.1.	0	RW
5	Fix this bit to "0."		0	RW
6	V_{REF} connection select bit **(Note 5)**	0 : Pin V_{REF} is connected. 1 : Pin V_{REF} is disconnected.	0	RW
7	The value is "0" at reading.		0	–

Notes 1: These bits are invalid in the one-shot and repeat modes. (They may be either "0" or "1.")
2: When using pin AN_7, be sure that the D-A_0 output enable bit (bit 0 at address 96_{16}) = "0" (output disabled).
3: Be sure to select the frequently-used analog input pins in the repeat sweep mode 1.
4: Fix this bit to "0" in the one-shot mode, repeat mode, and single sweep mode.
5: When this bit is cleared from "1" to "0," be sure to start the A-D conversion after an interval of 1 µs or more has elapsed.
6: Writing to each bit of the A-D control register 1 must be performed while the A-D converter halts, regardless of the A-D operation mode.

Fig. 16.4.1 Structure of A-D control register 1

MEMORANDUM

CHAPTER 17

DEBUG FUNCTION

17.1 Overview

When the CPU fetches an op code (op-code fetch), the debug function generates an address matching detection interrupt request if a selected condition is satisfied as a result of comparison between the address where the op code to be fetched is stored (in other words, the contents of PG and PC) and the specified address.
The debug function provides the following 2 modes:

(1) Address matching detection mode
When the contents of PG and PC match with the specified address, an address matching detection interrupt request occurs. This mode can be used for avoiding or modifying a portion of a program.

(2) Out-of-address-area detection mode
When the contents of PG and PC go out of the specified area, an address matching detection interrupt request occurs. This mode can be used for the program runaway detection by specifying the area where a program exists.

Note that an address matching detection interrupt is a non-maskable software interrupt. For details of this interrupt, refer to **"CHAPTER 6. INTERRUPTS."**
In addition, the debug function cannot be evaluated by a debugger. Therefore, do not use a debugger when using the debug function.

17.2 Block description

Figure 17.2.1 shows the block diagram of the debug function, and the registers relevant to this function are described in the following.

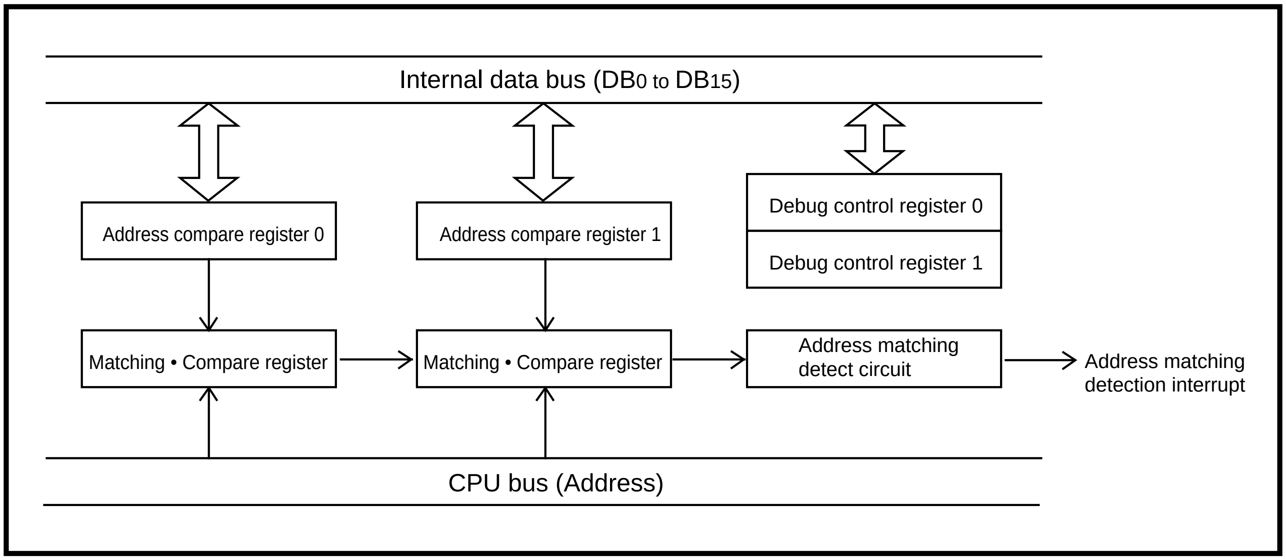


Fig. 17.2.1 Block diagram of debug function

17.2.1 Debug control register 0

Figure 17.2.2 shows the structure of the debug control register 0.

Debug control register 0 (Address 66_{16})

b7	b6	b5	b4	b3	b2	b1	b0
	0		0	0			

Bit	Bit name	Function	At reset	R/W
0	Detect condition select bits **(Note 1)**	b2 b1 b0 0 0 0 : Do not select. 0 0 1 : Address matching detection 0 0 1 0 : Address matching detection 1 0 1 1 : Address matching detection 2 1 0 0 : Do not select. 1 0 1 : Out-of-address-area detection 1 1 0 : } Do not select. 1 1 1 : }	**(Note 2)**	RW
1			**(Note 2)**	RW
2			**(Note 2)**	RW
3	Fix these bits to “00.”		**(Note 2)**	RW
4			**(Note 2)**	RW
5	Detect enable bit	0 : Detection disabled. 1 : Detection enabled.	**(Note 2)**	RW
6	Fix this bit to “0.”		**(Note 2)**	RW
7	The value is “1” at reading.		1	—

Notes 1: These bits are valid when the detect enable bit (bit 5) = “1.” Therefore, these bits must be set before or simultaneously with setting of the detect enable bit to “1.”
2: At power-on reset, each bit becomes “0”; at hardware reset or software reset, each bit retains the value immediately before reset.

Fig. 17.2.2 Structure of debug control register 0

(1) Detect condition select bits (bits 0 to 2)

These bits are used to select an occurrence condition for an address matching detection interrupt request. This condition can be selected from the following:

■ **Address matching detection 0**

An address matching detection interrupt request occurs when the contents of PG and PC match with the address being set in the address compare register 0 (addresses 68_{16} to $6A_{16}$); (Refer to section **“17.3 Address matching detection mode.”**)

■ **Address matching detection 1**

An address matching detection interrupt request occurs when the contents of PG and PC match with the address being set in the address compare register 1 (addresses $6B_{16}$ to $6D_{16}$); (Refer to section **“17.3 Address matching detection mode.”**)

■ **Address matching detection 2**

An address matching detection interrupt request occurs when the contents of PG and PC match with the address being set in the address compare register 0 (addresses 68_{16} to $6A_{16}$) or address compare register 1 (addresses $6B_{16}$ to $6D_{16}$); (Refer to section **“17.3 Address matching detection mode.”**)

■ **Out-of-address-area detection**

An address matching detection interrupt request occurs when the contents of PG and PC are less than the address being set in the address compare register 0 (addresses 68_{16} to $6A_{16}$) or larger than the address compare register 1 (addresses $6B_{16}$ to $6D_{16}$); (Refer to section **“17.4 Out-of-address-area detection mode.”**)

(2) Detect enable bit (bit 5)

If any selected condition is satisfied when this bit = "1," an address matching detection interrupt request occurs.

17.2.2 Debug control register 1

Figure 17.2.3 shows the structure of the debug control register 1.

Debug control register 1 (Address 67_{16})

b7	b6	b5	b4	b3	b2	b1	b0
				1			0

Bit	Bit name	Function	At reset	R/W
0	Fix this bit to "0."		**(Note 1)**	RW
1	The value is "0" at reading.		**(Note 1)**	RO
2	Address compare register access enable bit **(Note 2)**	0 : Disabled. 1 : Enabled.	0	RW
3	Fix this bit to "1" when using the debug function.		0	RW
4	Nothing is assigned.		Undefined	—
5	While a debugger is not used, the value is "0" at reading. While a debugger is used, the value is "1" at reading.		0	RO
6	Address-matching-detection 2 decision bit (Valid when the address matching detection 2 is selected.)	0 : Matches with the contents of the address compare register 0. 1 : Matches with the contents of the address compare register 1.	0	RO
7	The value is "0" at reading.		0	—

Notes 1: At power-on reset, each bit become "0"; at hardware reset or software reset, each bit retains the value immediately before reset.

2: Be sure to set this bit to "1" immediately before the access to the address compare registers 0 and 1 (addresses 68_{16} to $6D_{16}$). Then, be sure to clear this bit to "0" immediately after this access.

Fig. 17.2.3 Structure of debug control register 1

(1) Address compare register access enable bit (bit 2)

Setting this bit to "1" enables reading from or writing to the contents of address compare registers 0 and 1 (addresses 68_{16} to $6D_{16}$), while clearing this bit to "0" disables this reading or writing.

Be sure to set this bit to "1" immediately before reading from or writing to the address compare registers 0 and 1, and then clear it to "0" immediately after this reading or writing.

(2) Address-matching-detection 2 decision bit (bit 6)

When the address matching detection 2 is selected, this bit is used to decide which of the addresses being set in the address compare registers 0 and 1 matches with the contents of PG and PC.

This bit is cleared to "0" when the contents of PG and PC matches with the address being set in address compare register 0 and set to "1" when the contents of PG and PC match with the one being set in the address compare register 1.

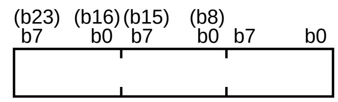
This bit is invalid when the address matching detection 0 and 1 are selected.

17.2.3 Address compare registers 0 and 1

Each of the address compare registers 0 and 1 consists of 24 bits, and the address to be detected is set here.

Figure 17.2.4 shows the structures of the address compare registers 0 and 1.

Address compare register 0 (Addresses $6A_{16}$ to 68_{16})
Address compare register 1 (Addresses $6D_{16}$ to $6B_{16}$)



Bit	Function	At reset	R/W
23 to 0	The address to be detected (in other words, the start address of instructions) is set here.	Undefined	RW

Note: When accessing to these registers, be sure to set the address compare register access enable bit (bit 2 at address 67_{16}) to "1" immediately before the access. Then, be sure to clear this bit to "0" immediately after this access.

Fig. 17.2.4 Structures of address compare registers 0 and 1

At op-code fetch, the contents of PG and PC are compared with the addresses being set in the address compare register 0 or 1. Therefore, be sure to set the start address of an instruction into the address compare register 0 or 1. If such an address as in the middle of instructions or in the data table is set into the address compare register 0 or 1, no address matching detection interrupt request occurs because this address does not match with the contents of PG and PC.

Note that, before the instruction at the address being set in the address compare register 0 or 1 is executed, an address matching detection interrupt request occurs and is accepted.

17.3 Address matching detection mode

When the contents of PG and PC match with the specified address, an address matching detection interrupt request occurs.

17.3.1 Setting procedure for address matching detection mode

Figure 17.3.1 shows an initial setting example for registers relevant to the address matching detection mode.

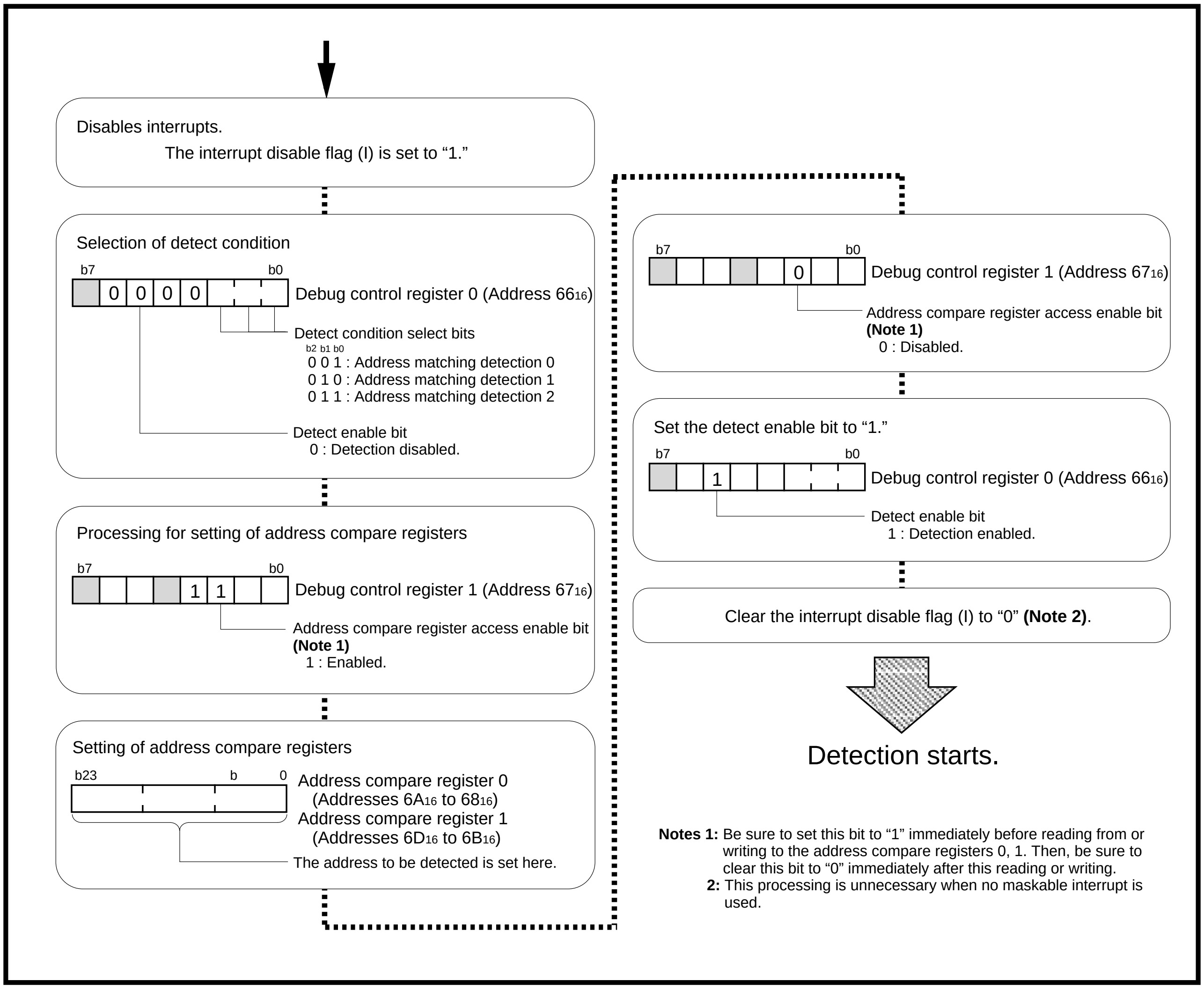


Fig. 17.3.1 Initial setting example for registers relevant to address matching detection mode

17.3.2 Operations in address matching detection mode

① Setting the detect enable bit to "1" initiate to compare the contents of PG and PC with one of the contents of the following registers. This comparison is performed at each op-code fetch:
- When the address matching detection 0 is selected, the contents of the address compare register 0 are used for the above comparison.
- When the address matching detection 1 is selected, the contents of the address compare register 1 are used for the above comparison.
- When the address matching detection 2 is selected, the contents of the address compare register 0 or 1 are used for the above comparison.

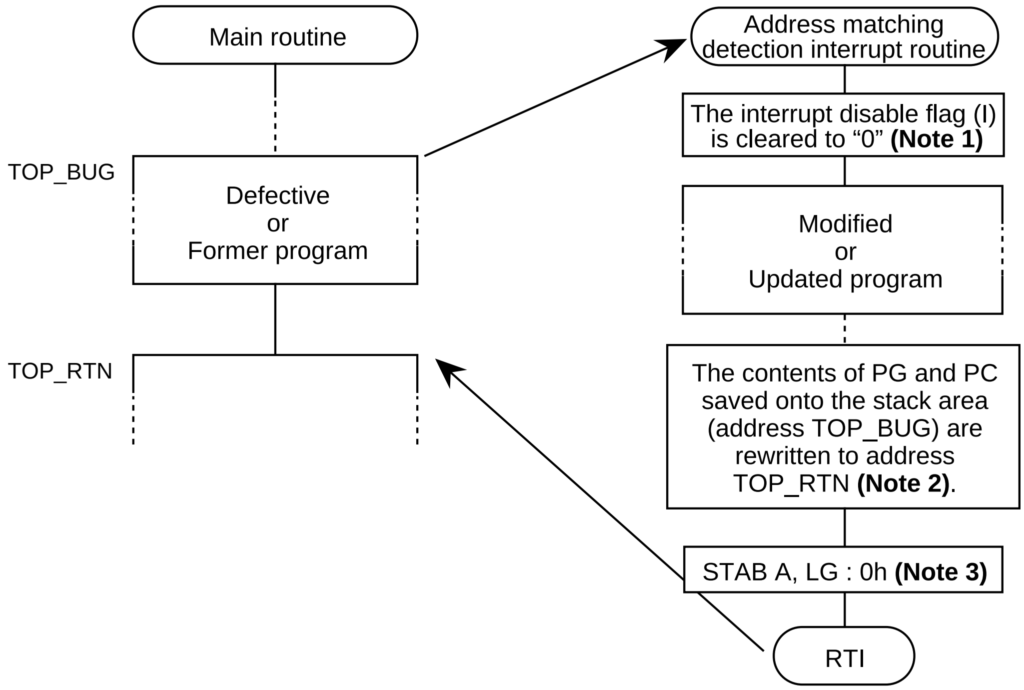
② When the address which matches with the above register's contents is detected, an address matching detection interrupt request occurs, and then, this request will be accepted.

③ Perform the necessary processing with an address matching detection interrupt routine.

④ The contents of PG, PC, and PS at acceptance of the address matching detection interrupt request are saved onto the stack area. Therefore, be sure to rewrite the above contents of PG and PC to a certain return address, and return to the address by using the **RTI** instruction.

When an address matching detection interrupt request has been accepted, the interrupt disable flag (I) is set to "1"; the processor interrupt priority level (IPL) does not change.
Figures 17.3.2 and 17.3.3 show the examples of the ROM correct processing using the address matching detection mode.

■ Address matching detection 0 or 1 selected



TOP_BUG : The start address of defective or former program.
→This address is to be set in the address compare register 0 or 1, in advance.

TOP_RTN : The address next to the defective or former program.

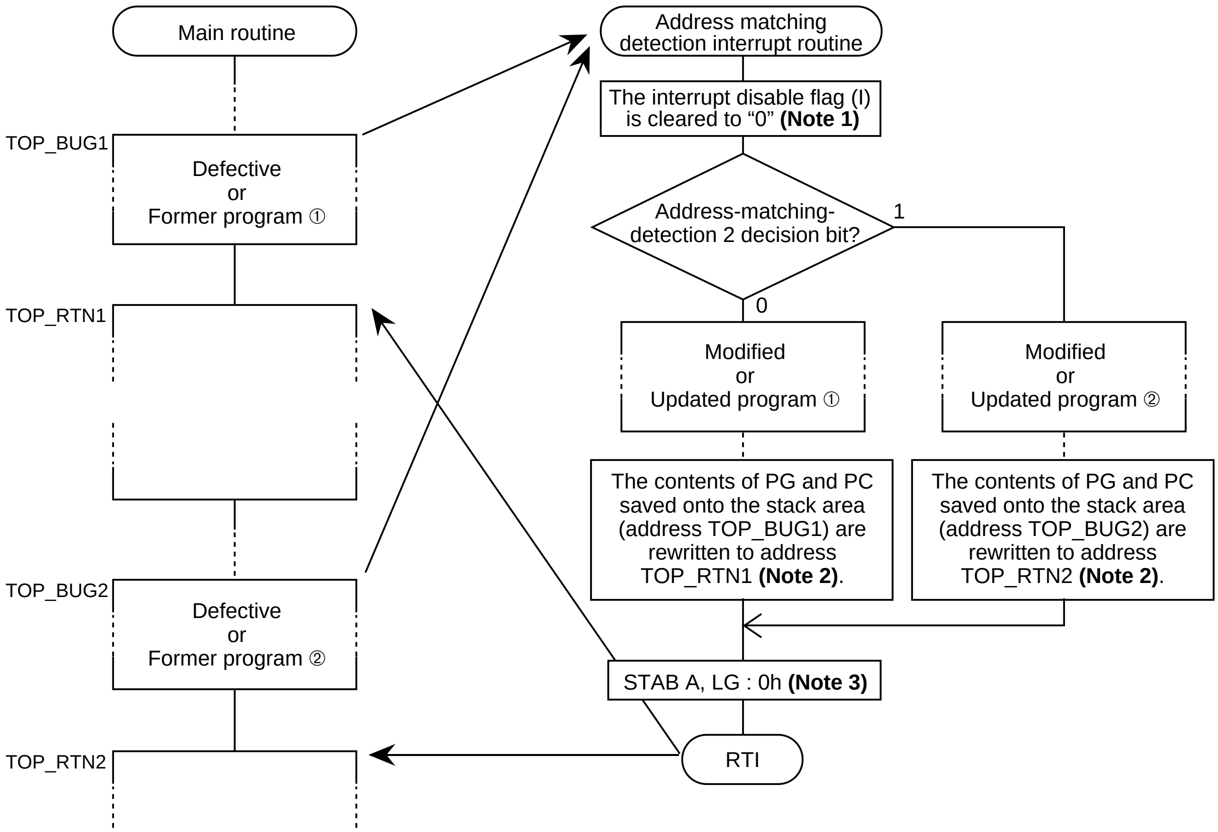
Notes 1: When an address matching detection interrupt request has been accepted, the interrupt disable flag (I) is set to “1.” If another interrupt requests is required to be accepted under the same conditions as those of the defective or former program, be sure to clear the interrupt disable flag (I) to “0” at the start of an address matching detection interrupt routine.

2: Each status of PG, PC, and PS immediately before acceptance of an address matching detection interrupt request is saved onto the stack area. (The contents of PG, PC, and PS are saved onto the stack area in this order.) Refer to section **“6.7 Sequence from acceptance of interrupt request until execution of interrupt routine.”**

3: Make sure that this instruction is executed in the absolute long addressing mode. The above is just an example. In an actual programming, be sure to refer to the format of the assembler description to be used.

Fig. 17.3.2 Example of ROM correct processing using address matching detection mode (1)

■ Address matching detection 2 selected



TOP_BUG1 : The start address of defective or former program ①.
→This address is to be set in the address compare register 0, in advance.
TOP_RTN1 : The address next to the defective or former program ①.
TOP_BUG2 : The start address of defective or former program ②.
→This address is to be set in the address compare register 1, in advance.
TOP_RTN2 : The address next to the defective or former program ②.

Notes 1: When an address matching detection interrupt request has been accepted, the interrupt disable flag (I) is set to “1.” If another interrupt requests is required to be accepted under the same conditions as those of the defective or former program, be sure to clear the interrupt disable flag (I) to “0” at the start of an address matching detection interrupt routine.

2: Each status of PG, PC, and PS immediately before acceptance of an address matching detection interrupt request is saved onto the stack area. (The contents of PG, PC, and PS are saved onto the stack area in this order.) Refer to section **“6.7 Sequence from acceptance of interrupt request until execution of interrupt routine.”**

3: Make sure that this instruction is executed in the absolute long addressing mode. The above is just an example. In an actual programming, be sure to refer to the format of the assembler description to be used.

Fig. 17.3.3 Example of ROM correct processing using address matching detection mode (2)

17.4 Out-of-address-area detection mode

When the contents of PG and PC go out of the range of the specified area, an address matching detection interrupt request occurs.

17.4.1 Setting procedure for out-of-address-area detection mode

Figure 17.4.1 shows an initial setting example for registers relevant to the out-of-address-area detection mode.

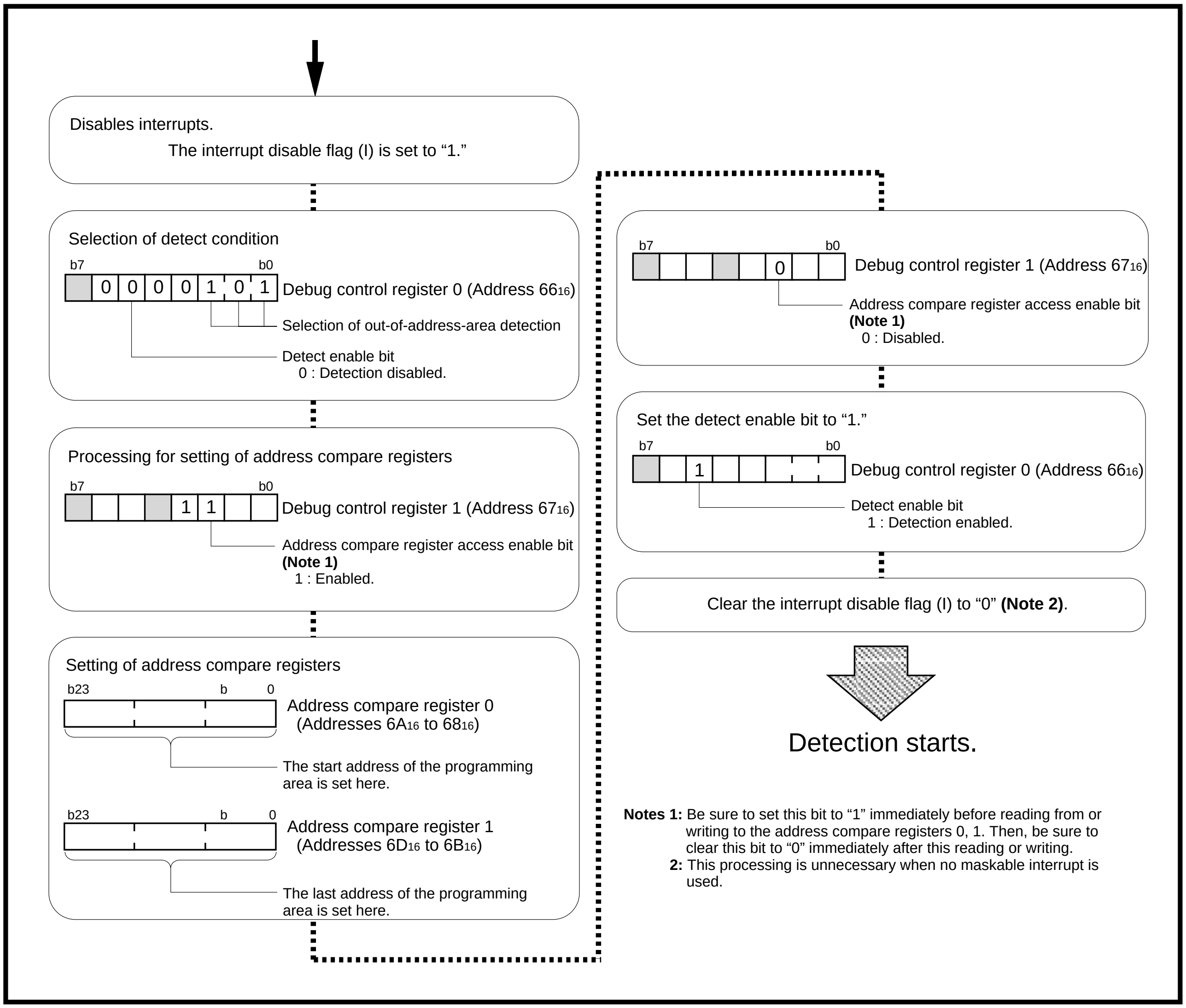


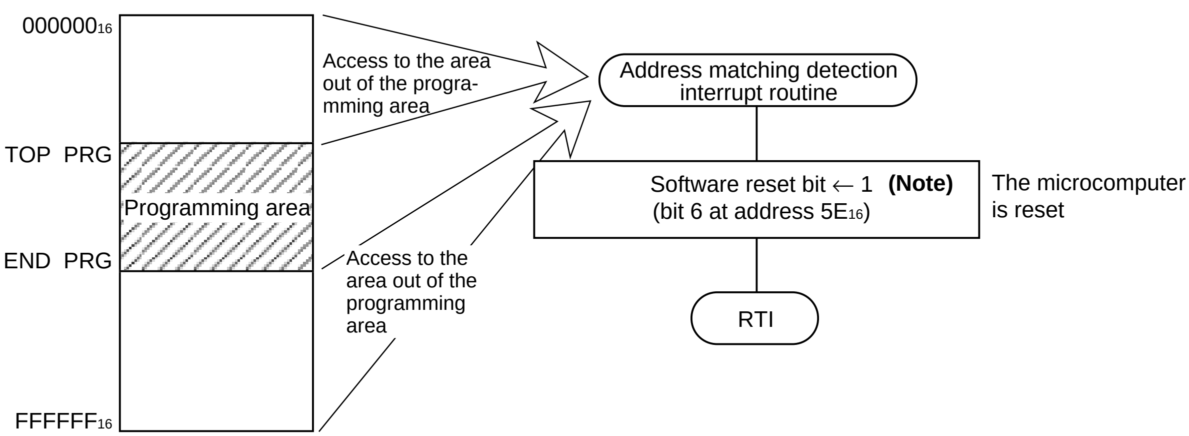
Fig. 17.4.1 Initial setting example for registers relevant to out-of-address-area detection mode

17.4.2 Operations in out-of-address-area detection mode

① Setting the detect enable bit to “1” initiate to compare the contents of PG and PC with the contents of the address compare registers 0 and 1.
② When an address less than the contents of the address compare registers 0 or larger than the one of the address compare register 1 is detected, an address matching detection interrupt request occurs, and then, this request will be accepted.
③ Perform the necessary processing with an address matching detection interrupt routine.
④ The contents of PG, PC, and PS at acceptance of the address matching detection interrupt request are saved onto the stack area. Therefore, be sure to rewrite the above contents of PG and PC to a certain return address, and return there by using the **RTI** instruction.

When an address matching detection interrupt request has been accepted, the interrupt disable flag (I) is set to “1”; the processor interrupt priority level (IPL) does not change.

By setting the start address of the programming area into the address compare register 0 and the last address of the programming area into the address compare register 1, a program runaway (in other words, fetching op codes from the area out of the programming area) can be detected. If any program runaway is detected and reset of the microcomputer is required, be sure to write “1” into the software reset bit (bit 6 at address $5E_{16}$) within an address matching detection interrupt routine.
Figure 17.4.2 shows an example of program runaway detection using the out-of-address-area detection mode.



TOP_PRG : Start address of programming area → This address is to be set into the address compare register 0, in advance.
END_PRG : Last address of programming area → This address is to be set into the address compare register 1, in advance.

Note: A program runaway may affect the contents of the data bank register (DT), the direct page registers (DPRi) etc. Therefore, the contents of these registers must be rewritten in order to write “1” to the software reset bit with an addressing mode using DT, DPRi, etc.

Fig. 17.4.2 Example of program runaway detection using out-of-address-area detection mode

[Precautions for debug function]

1. The debug function cannot be evaluated by a debugger. Therefore, do not use a debugger when using the debug function.

2. When returning from an address matching detection interrupt routine, be sure to rewrite the saved contents of PG and PC to a certain return address, and then return there by using the **RTI** instruction. However, this is unnecessary processing when the software reset is performed within an address matching detection interrupt routine for program runaway detection, etc.

3. Be sure to set the start address of an instruction into the address compare register 0 or 1.

CHAPTER 18
APPLICATIONS

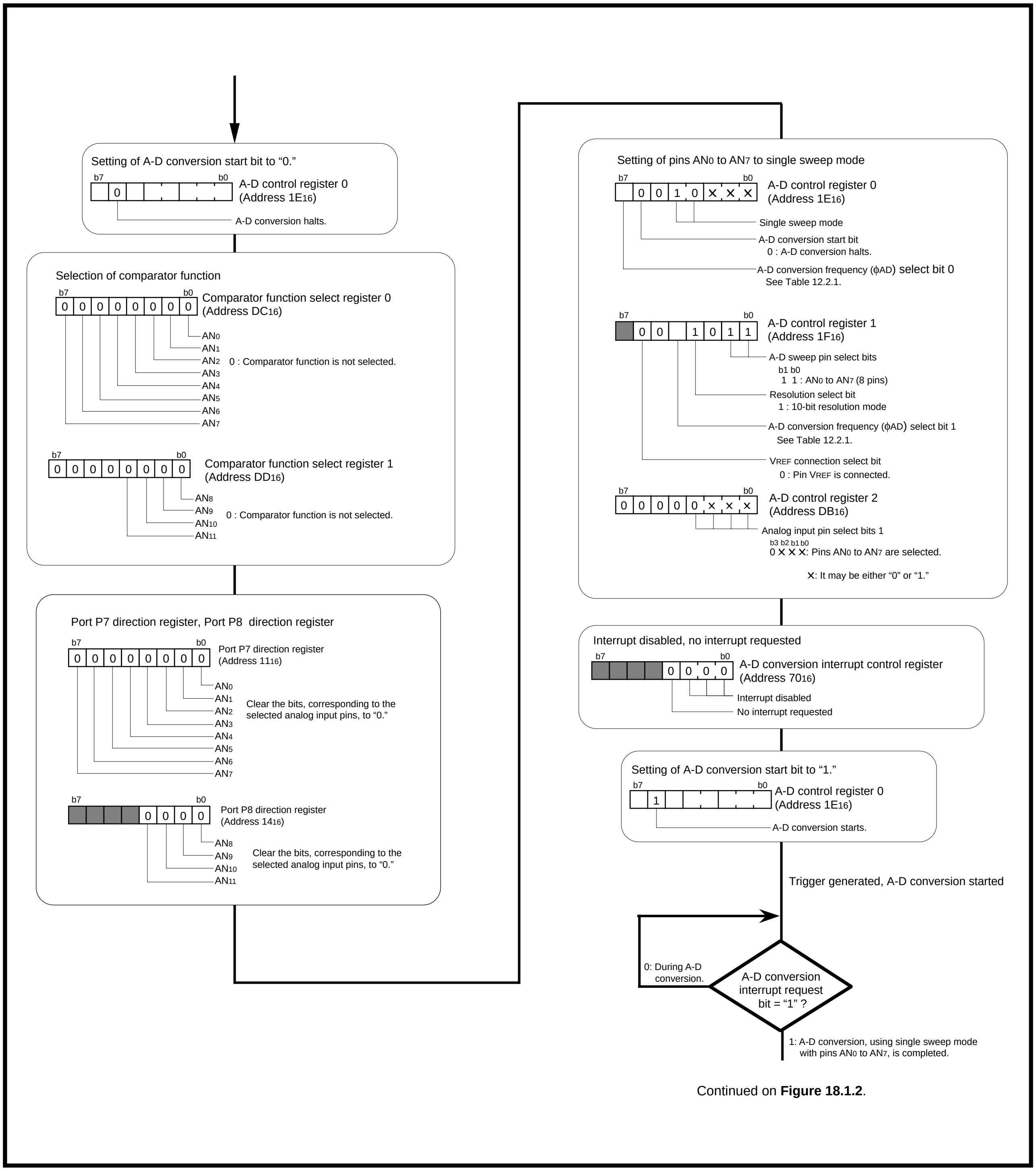
A certain application example is described below.
This application described here is just an example. Therefore, before actual using it, be sure to properly modify it according to the user's system and sufficiently evaluate it.

18.1 Application example of A-D converter

18.1.1 Application example of A-D converter, using single sweep mode with pins AN_0 to AN_{11}

Figures 18.1.1 to 18.1.3 show an application example of the A-D converter, using the single sweep mode with pins AN_0 to AN_{11}. For details, refer to the following specifications:

① For pins AN_0 to AN_7, the single sweep mode is used.
② For pins AN_8 to AN_{11}, the one-shot mode is used.
③ 10-bit resolution mode
④ No A-D conversion interrupt



Continued on **Figure 18.1.2**.

Fig. 18.1.1 Application example of A-D converter, using single sweep mode with pins AN_0 to AN_{11} (1)

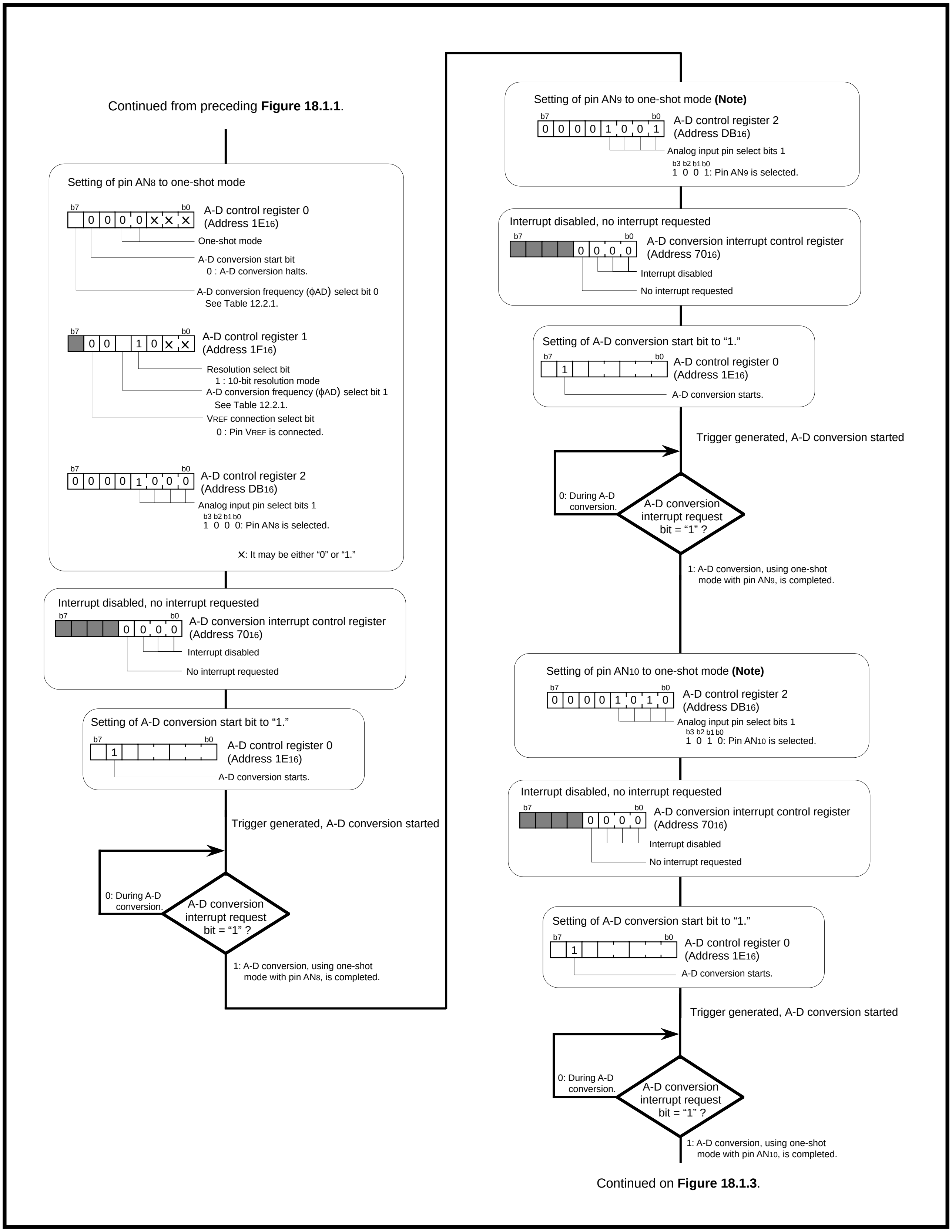


Fig. 18.1.2 Application example of A-D converter, using single sweep mode with pins AN_0 to AN_{11} (2)

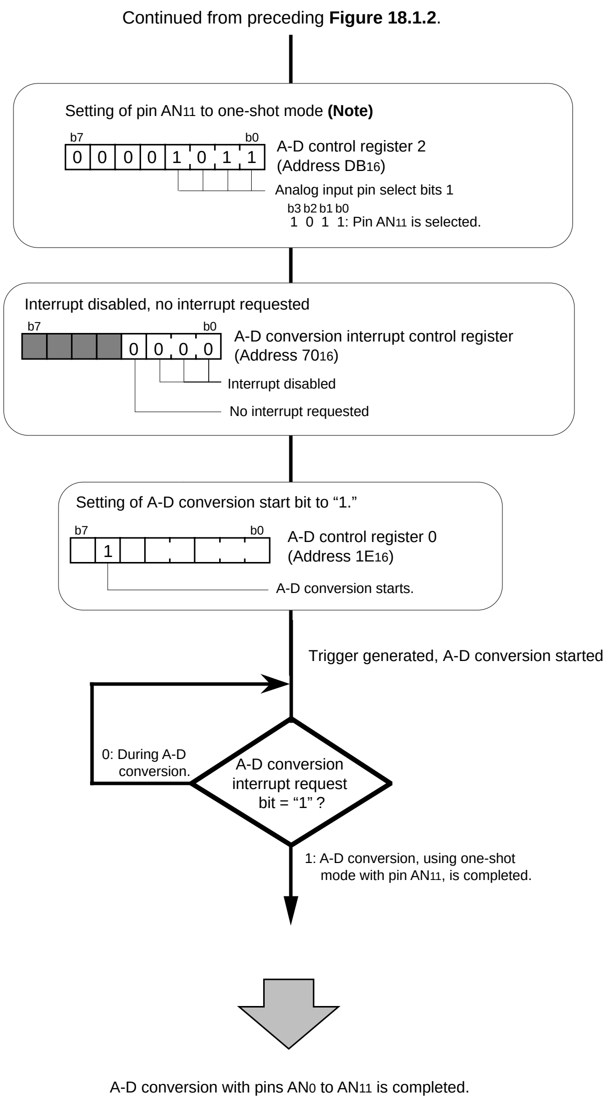


Fig. 18.1.3 Application example of A-D converter, using single sweep mode with pins AN_0 to AN_{11} (3)

MEMORANDUM

CHAPTER 19

FLASH MEMORY VERSION

19.1 Overview

The flash memory version is provided with the same function as that of the mask ROM version except that the former includes the flash memory. Note that, however, part of the SFR area of the flash memory version differs from that of the mask ROM version. (Refer to section **"19.1.1 Memory assignment."**) Also, the stop mode terminate operation of the flash memory version differs from that of the mask ROM version. (Refer to section **"19.1.2 Single-chip mode."**)
In the flash memory version, its internal flash memory can be handled in the following three reprogramming modes: flash memory CPU reprogramming mode, flash memory serial I/O mode, and flash memory parallel I/O mode.
Table 19.1.1 lists the performance overview of the flash memory version. (For the items not listed in Table 19.1.1, see Table 1.1.1.)

Table 19.1.1 Performance overview of flash memory version

Item		Performance
Power source voltage		5 V ± 0.5 V
Programming/Erase voltage		5 V ± 0.5 V
Flash memory reprogramming modes		Flash memory CPU reprogramming mode, Flash memory serial I/O mode, Flash memory parallel I/O mode
Programming	CPU reprogramming mode, Flash memory serial I/O mode	Programmed in a unit of word
	Flash memory Parallel I/O mode	Programmed in a unit of byte
Erase method		Block erase or Total erase
Maximum number of reprograms (programming and erasure)		100

For the flash memory version, in addition to the same single-chip mode as that of the mask ROM version, any of the operating modes listed in Table 19.1.2 can further be selected by the voltage levels applied to pins MD1 and MD0. Table 19.1.3 also lists the overview of flash memory reprogramming modes.

Note: Do not switch the voltages applied to pins MD0 and MD1 while the microcomputer is active.

Table 19.1.2 Operating mode selection according to voltages applied to pins MD0 and MD1

MD1	MD0	Operating modes
V_{SS}	V_{SS}	Single-chip mode
V_{SS}	V_{CC}	– **(Note 1)**
V_{CC}	V_{SS}	Boot mode **(Note 2)**
V_{CC}	V_{CC}	Flash memory parallel I/O mode **(Note 3)**

Notes 1: Do not select.
2: Refer to section **"19.1.3 Boot mode."**
3: Refer to section **"19.4 Flash memory parallel I/O mode."**

Table 19.1.3 Overview of flash memory reprogramming modes

Flash memory reprogramming mode	Flash memory CPU reprogramming mode	Flash memory serial I/O mode	Flash memory parallel I/O mode
Functional overview	User ROM area is reprogrammed by the CPU executing software commands.	User ROM area is reprogrammed by using a dedicated serial programmer.	Boot ROM area and User ROM area are reprogrammed by using a dedicated parallel programmer.
Reprogrammable area	User ROM area	User ROM area	User ROM area, Boot ROM area
Operating mode available	Single-chip mode, Boot mode	Boot mode	Flash memory parallel I/O mode
ROM programmer available	(Unnecessary)	Serial programmer **(Note)**	Parallel programmer **(Note)**

Note: For details of the serial and parallel programmers, please visit MITSUBISHI TOOL Homepage (http://www.tool-spt.maec.co.jp/index_e.htm).

FLASH MEMORY VERSION

19.1 Overview

19.1.1 Memory assignment

Figure 19.1.1 shows the memory assignment of the M37905F8.

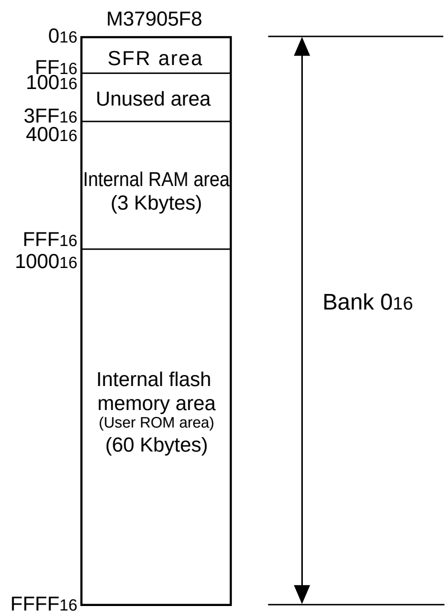


Fig. 19.1.1 Memory assignment of M37905F8

In addition to the internal flash memory area (in other words, user ROM area) shown in Figure 19.1.1, the flash memory version has the boot ROM area of 8 Kbytes.
Figure 19.1.2 shows the memory assignment of the internal flash memory.
The user ROM area is divided into several blocks. The user ROM area is reprogrammed in the flash memory CPU reprogramming mode, serial I/O mode, and parallel I/O mode.
The boot ROM area is assigned at addresses, overlapping with the user ROM area, however, the boot ROM area exists in the defferent memory; the boot ROM area can be reprogrammed only in the flash memory parallel I/O mode. (Refer to section **"19.4 Flash memory parallel I/O mode."**). When being reset with pin MD1 tied to Vcc level and pin MD0 to Vss level, the software in the boot ROM area is executed after reset. (Refer to section **"19.1.3 Boot mode."**) When pin MD1 = Vss level, however, the contents of the boot ROM area cannot be read out.

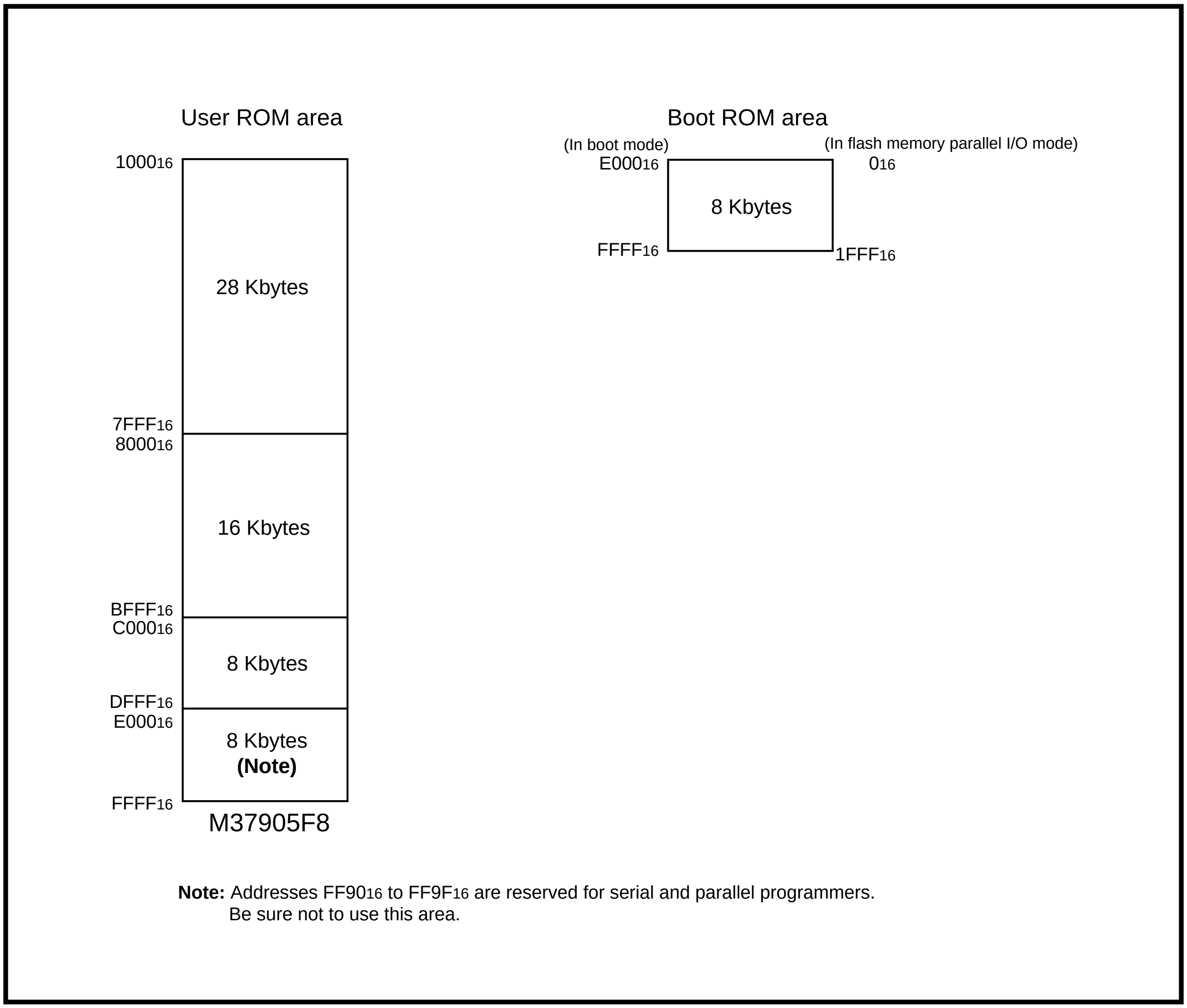


Fig. 19.1.2 Memory assignment of internal flash memory

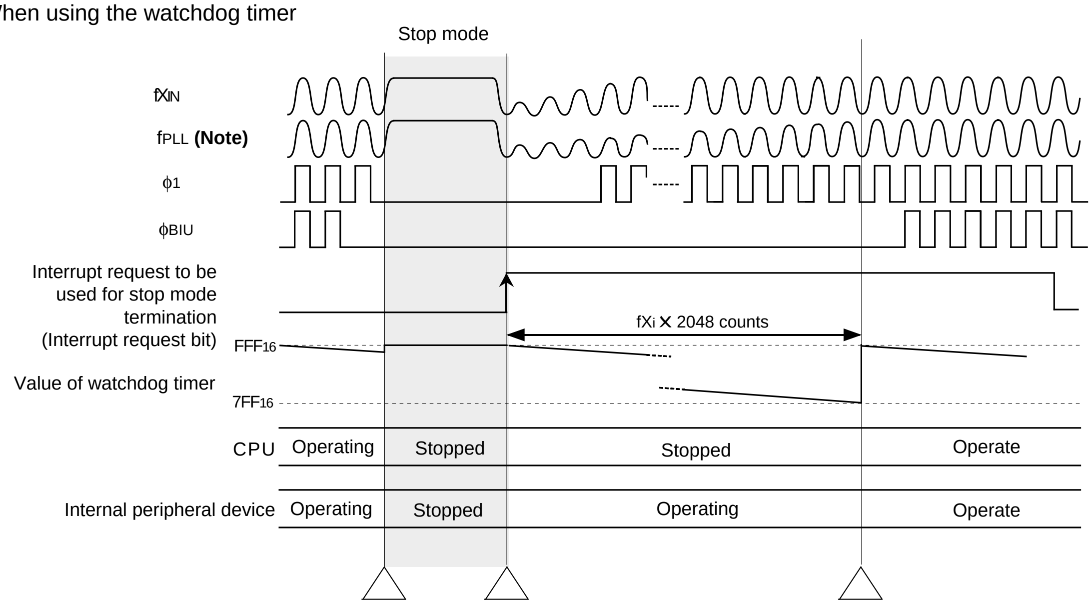
19.1.2 Single-chip mode

When being reset with both of pins MD1 and MD0 tied to Vss level, the microcomputer enters the single-chip mode. In the single-chip mode, the software in the user ROM area is executed after reset.
The difference between the flash memory version and the mask ROM version is as follows:

- Stop mode terminate operation

(1) Stop mode terminate operation
Figure 19.1.3 shows stop mode terminate sequence owing to an interrupt request occurrence in the flash memory version. (Refer from section **"Stop mode"**.)
In the flash memory version, when the watchdog timer is not used for termination of the stop mode, an interrupt request is accepted after a maximum of 10 μs has elapsed since the interrupt request occurred.

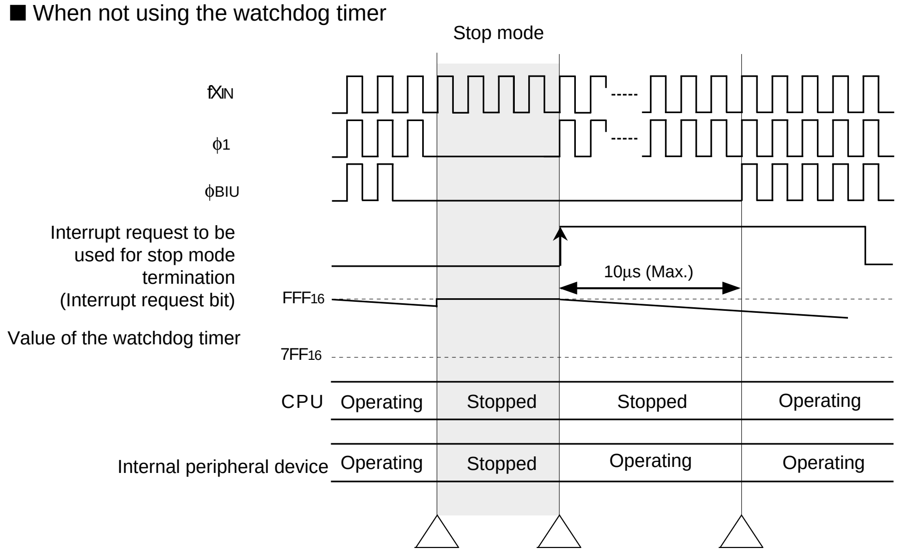
■ When using the watchdog timer



- STP instruction is executed.
- Interrupt request to be used for termination occurs.
- Oscillation stars. (When an external clock is input from pin XIN, clock input starts.)
- PLL frequency multiplier starts its operation.
- Watchdog timer starts counting.
- Watchdog timer's MSB = "0" (However, watchdog timer interrupt request dose not occur.)
- Each supply of φCPU, φBIU starts.
- Interrupt request which was used for termination is accepted.

Note: This applies when the PLL circuit operation enable bit (bit 1 at address BC_{16}) = "1."
fX_i : fX_{16}, fX_{32}, fX_{64}, fX_{128}
There are clocks selected by the watchdog timer clock source bits at STP termination (bits 6, 7 at address 61_{16})

■ When not using the watchdog timer



- STP instruction is executed.
- Interrupt request to be used for termination occurs.
- Clock input from pin XIN starts.
- Watchdog timer starts counting.
- Each supply of φCPU, φBIU starts.
- Interrupt request which was used for termination is accepted.

Fig. 19.1.3 Stop mode terminate sequence owing to interrupt request occurrence

19.1.3 Boot mode

When being reset with pin MD1 tied to Vcc level and pin MD0 to Vss level, the flash memory version enters the boot mode. In the boot mode, the software in the boot ROM area is executed after reset.
In the boot mode, either the boot ROM area or the user ROM area can be selected with the user ROM area select bit (bit 5 at address $9E_{16}$). The boot ROM area is located at addresses $E000_{16}$ to $FFFF_{16}$ in the boot mode.
A reprogramming control firmware used in the flash memory serial I/O mode has been stored in the boot ROM area on shipment. (Refer to section **"19.3 Flash memory serial I/O mode."**) Therefore, when being reset in the boot mode, the flash memory version enters the flash memory serial I/O mode, allowing the user ROM area to be reprogrammed with a dedicated serial programmer.
Also the boot ROM area can be reprogrammed in the flash memory parallel I/O mode. If an appropriate reprogramming control software using the CPU reprogramming mode has been stored in the boot ROM area, reprogramming suitable for the user's system is enabled.
Note that if the boot ROM area has been reprogrammed in the flash memory parallel I/O mode, the flash memory serial I/O mode cannot be used.

19.2 Flash memory CPU reprogramming mode

In this mode, the user ROM area can be reprogrammed by the central processing unit (CPU) executing software commands. Therefore, this mode allows the user to reprogram the contents of the user ROM area with the microcomputer mounted on the final printed circuit board, without using any ROM programmer.
Be sure to store the reprogramming control software into the user ROM area or the boot ROM area in advance. In the flash memory CPU reprogramming mode, however, an opcode cannot be fetched for the internal flash memory. Accordingly, be sure to transfer the reprogramming control software to an area other than the internal flash memory area (e.g. the internal RAM area), and then execute the software in this area.
The flash memory CPU reprogramming mode is available in any of the single-chip and boot modes.

The software commands listed in Table 19.2.1 can be used in the flash memory CPU reprogramming mode. For details of each command, refer to section **"19.2.4 Software commands."**
Note that commands and data must be read from and written into even-numbered addresses within the user ROM area, 16 bits at a time. At writing of software command codes, the high-order 8 bits (D_8 to D_{15}) are ignored. (Except for the write data at the 2nd bus cycle of the programming command code.)

Table 19.2.1 Software commands

Software commands	1st bus cycle			2nd bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data
Read Array	Write	X	FF_{16}	—	—	—
Read Status Register	Write	X	70_{16}	Read	X	SRD
Clear Status Register	Write	X	50_{16}	—	—	—
Programming	Write	X	40_{16}	Write	WA	WD
Block Erase	Write	X	20_{16}	Write	BA	$D0_{16}$
Erase All Blocks	Write	X	20_{16}	Write	X	20_{16}

SRD : Status register data (D_0 to D_7)
WA : Write address (A_7 to A_0 to be incremented by 2 from "00_{16}" to "FE_{16}")
WD : Write data (16 bits)
BA : The highest address of a block (Note that A_0 = 0.)
X : Arbitrary even-numbered address in user ROM area (A_0 = 0)

19.2 Flash memory CPU reprogramming mode

19.2.1 Flash memory control register

Figure 19.2.1 shows the structure of the flash memory control register.

Flash memory control register (Address $9E_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	RY/$\overline{BY}$ status bit	0 : BUSY (Automatic programming or erase operation is active.) 1 : READY (Automatic programming or erase operation has been completed.)	1	RO
1	CPU reprogramming mode select bit	0 : Flash memory CPU reprogramming mode is invalid. 1 : Flash memory CPU reprogramming mode is valid.	0	RW **(Notes 1, 2)**
2	The value is "0" at reading.		0	—
3	Flash memory reset bit **(Note 3)**	Writing "1" into this bit discontinues the access to the internal flash memory. This causes the built-in flash memory circuit being reset.	0	RW **(Note 4)**
4	The value is "0" at reading.		0	—
5	User ROM area select bit (Valid in boot mode) **(Note 5)**	0 : Access to boot ROM area 1 : Access to user ROM area	0	RW **(Note 2)**
7, 6	The value is "0" at reading.		0	—

Notes 1: In order to set this bit to "1," write "0" followed with "1" successively; while in order to clear this bit "0," write "0."
2: Writing to this bit must be performed in an area other than the internal flash memory.
3: This bit is valid when the CPU reprogramming mode select bit (bit 1) = "1": on the other hand, when the CPU reprogramming mode select bit = "0," be sure to fix this bit to "0." Rewriting of this bit must be performed with the CPU reprogramming mode select bit = "1."
4: After writing of "1" to this bit, be sure to confirm the RY/$\overline{BY}$ status bit (bit 0) becomes "1"; and then, write "0" to this bit.
5: When MD1 = Vss level, this bit is invalid. (It may be either "0" or "1.")

Fig. 19.2.1 Structure of flash memory control register

(1) RY/$\overline{BY}$ status bit (bit 0)

This bit is used to indicate the operating status of the sequencer. This bit is "0" during the automatic programming or erase operation is active and becomes "1" upon completion of them. This bit also changes during the execution of the programming, block erase, or erase all blocks command, but does not change owing to the execution of another command.

(2) CPU reprogramming mode select bit (bit 1)

Setting this bit to "1" allows the microcomputer to enter the flash memory CPU reprogramming mode to accept commands. In order to set this bit to "1," write "1" followed with "0" successively; while to clear this bit to "0," write "0."

Since the microcomputer enters the flash memory CPU reprogramming mode after setting this bit to "1," opcodes cannot be fetched for the internal flash memory. Accordingly, be sure to execute the instruction to be used for writing to this bit in an area other than the internal flash memory area (e.g. the internal RAM area).

When executing commands of the flash memory CPU reprogramming mode in the boot mode, be sure to set the user ROM area select bit (bit 5) to "1."

(3) Flash memory reset bit (bit 3)

Writing "1" to this bit discontinues the access to the user ROM area and causes the built-in flash memory control circuit to be reset. After this reset, the microcomputer enters the read array mode to set the RY/$\overline{BY}$ status bit (bit 0) to "1".

When this flash memory control circuit is reset with the flash memory reset bit during programming (automatic programming) or erase (automatic erase) operation, that programming or erase operation is discontinued to invalidate the data in the working block.

After writing of "1" to this bit, be sure to confirm the RY/$\overline{BY}$ status bit (bit 0) becomes "1"; and then, write "0" to this bit.

(4) User ROM area select bit (bit 5)

This bit is used to select either the boot ROM area or the user ROM area in the boot mode. In order to access the boot ROM area (read out), clear this bit to "0." On the other hand, in order to access the user ROM area (reading out, programming, or erase), set it to "1." Instructions for writing into this bit must be executed in an area other than the internal flash memory (e.g. the internal RAM area). Note that when MD1 = Vss level, the user ROM area is accessed (being read out) regardless of the contents of this bit.

19.2.2 Status register

The programming and erase operations for the internal flash memory are controlled by the sequencer in the internal flash memory. The status register indicates the completion states (normal or abnormal) of the programming and erase operations. For details of abnormal endings (errors), refer to section **"19.2.5 Full status check."**

Table 19.2.2 lists the bit definition of the status register.

The contents of the status register can be read out by the read status register command. (Refer to section **"19.2.4 Software commands."**)

Table 19.2.2 Bit definition of status register

Symbol (Data bus)	Status	Definition	
		"0"	"1"
SR.0 (D_0)	—	—	—
SR.1 (D_1)	—	—	—
SR.2 (D_2)	—	—	—
SR.3 (D_3)	—	—	—
SR.4 (D_4)	Programming Status	Terminated normally.	Error<Programming error>
SR.5 (D_5)	Erase Status	Terminated normally.	Error<Erase error>
SR.6 (D_6)	—	—	—
SR.7 (D_7)	—	—	—

Data bus: Indicates the data bus to be read out when the read status register command has been executed.

(1) Programming status bit (SR.4)

This bit is set to "1" if a programming error has occurred during the automatic programming (the programming) operation and cleared to "0" by executing the clear status register command. This bit is also cleared to "0" at reset.

(2) Erase status bit (SR.5)

This bit is set to "1" if an erase error has occurred during the automatic erase (the block erase or erase all unlocked blocks) operation and cleared to "0" by executing the clear status register command. This bit is also cleared to "0" at reset.

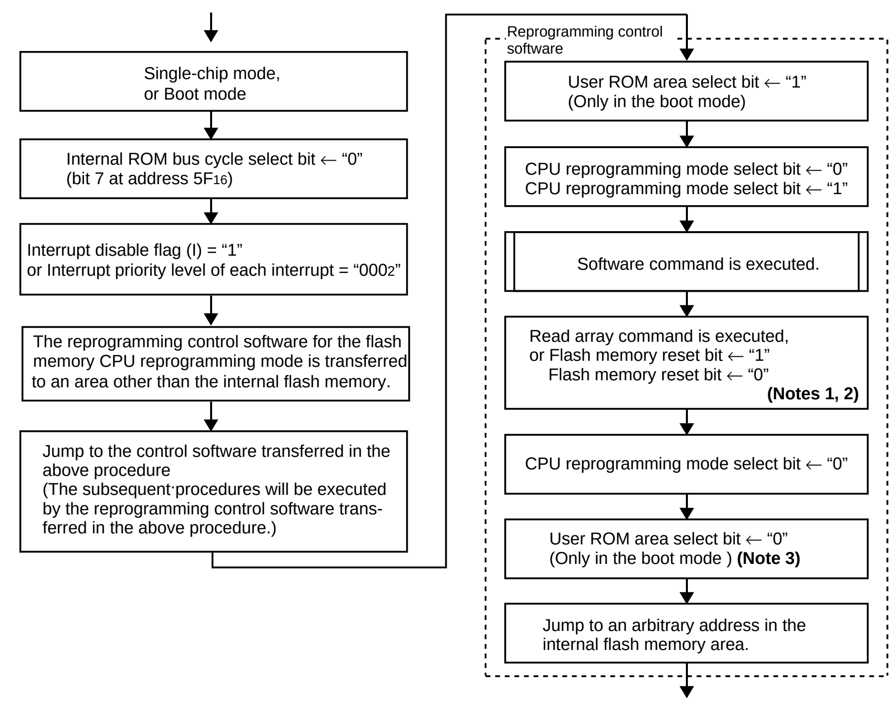
19.2.3 Setting and Terminate procedure for flash memory CPU reprogramming mode

Figure 19.2.2 shows the setting and terminate procedures for the flash memory CPU reprogramming mode. In the flash memory CPU reprogramming mode, opcodes cannot be fetched for the internal flash memory. Therefore, be sure to transfer the reprogramming control software to an area other than the internal flash memory and then execute the software in that area.
Moreover, in order to prevent any interrupt occurrence during the flash memory CPU reprogramming mode, before selecting this mode, be sure to set the interrupt disable flag (I) to “1” or set the interrupt priority level to “000_2” (interrupts disabled).
Also, we recommend to connect pins $\overline{P4OUT_{CUT}}$ and $\overline{P6OUT_{CUT}}$ with V_{CC} via resistors, respectively.

Even in the flash memory CPU reprogramming mode, periodically writing to the watchdog timer is required in order to prevent the watchdog timer interrupt occurrence.
At the same time, it is necessary to write to the watchdog timer just before executing the programming, block erase, or erase all blocks command in order to prevent the watchdog timer interrupt occurrence during the automatic programming and erase operation.

An interrupt, hardware reset, or software reset, generated in the flash memory CPU reprogramming mode, makes program runaway. If a program runaway has occurred, be sure to push the microcomputer into the power-on reset state.

<u>When an interrupt or reset is generated during the programming or erase operation, the contents of the corresponding block becomes invalidated.</u>



Notes 1: Before termination of the flash memory CPU reprogramming mode, be sure to execute the read array command or flash memory reset
2: After writing of “1” to the flash memory reset bit, be sure to confirm the RY/$\overline{BY}$ status bit (bit 0 at address $9E_{16}$) becomes “1”; and then, write “0” to this bit.
3: When the flash memory CPU reprogramming mode has been terminated with the user ROM area select bit (bit 5 at address $9E_{16}$) = “1,” the access to the user ROM area is selected.

Fig. 19.2.2 Setting and Terminate procedures for flash memory CPU reprogramming mode

19.2.4 Software commands

Software commands are described below.

Software commands and data must be read from and written into even-numbered addresses in the user ROM area, 16 bits at a time. At writing of a command code, the high-order 8 bits (D_8 to D_{15}) are ignored.

(1) Read array command

Writing command code "FF_{16}" at the 1st bus cycle pushes the microcomputer into the read array mode. When an address to be read is input at the next and the following bus cycles, the contents at the specified address are output to the data bus (D_0 to D_{15}), 16 bits at a time.

The read array mode is maintained until another software command is written.

(2) Read status register command

Writing command code "70_{16}" at the 1st bus cycle outputs the contents of the status register to the data bus (D_0 to D_7) by a read at the 2nd bus cycle. (See Table 19.2.2.)

(3) Clear status register command

Writing command code "50_{16}" at the 1st bus cycle clears two bits (SR.4 and SR.5) of the status register to "0." (See Table 19.2.2.)

(4) Programming

This command executes programming, one word at a time. Write command code "40_{16}" at the 1st bus cycle and then write data at the 2nd bus cycle, 16 bits at a time. After writing of one word has been completed, the automatic programming (programming and verification of data) operation is initiated. During the automatic programming operation, be sure not to access the flash memory or not to execute the next command. The completion of the automatic programming can be recognized by the RY/$\overline{BY}$ status bit (bit 0 at address $9E_{16}$).

After the automatic programming operation has been completed, the result of it can be recognized by reading out the status register. (Refer to section **"19.2.5 Full status check."**)

Figure 19.2.3 shows the programming operation flowchart.

<u>Note that, for the areas having already been programmed, be sure to program after an erase (block erase) operation.</u> If the programming command is executed for the areas having already been programmed, no programming error will occur, but the contents of the areas become undefined.

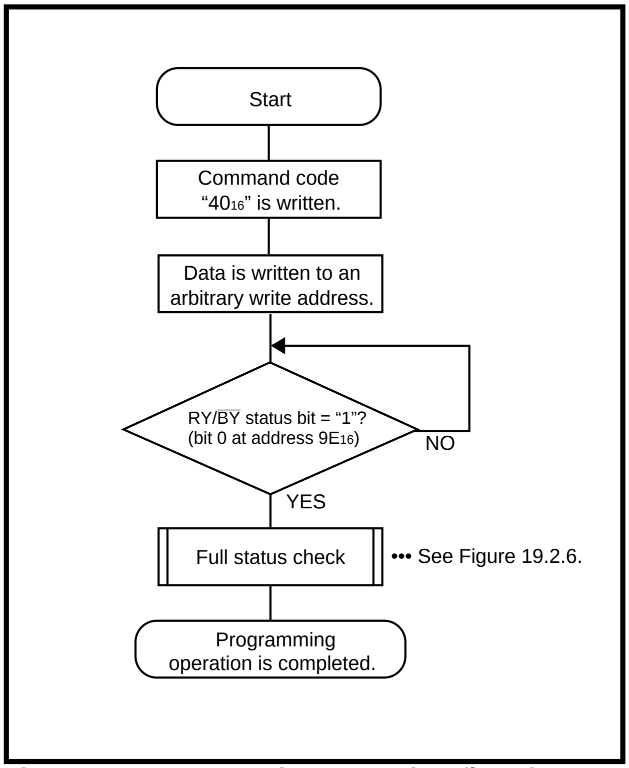


Fig. 19.2.3 Programming operation flowchart

(5) Block erase command

Writing of command code “20_{16}” at the 1st bus cycle and “$D0_{16}$” to the highest address (here, A_0 = 0) of the block to be erased at the 2nd bus cycle initiate the automatic erase (erase and erase-verify) operation for the specified block. During the automatic erase operation, be sure not to access the flash memory or not to execute the next command. The completion of the automatic erase operation can be recognized by the RY/$\overline{BY}$ status bit (bit 0 at address $9E_{16}$).

After the automatic erase operation is completed, the result of it can be recognized by reading out the status register. (Refer to section **“19.2.5 Full status check.”**)

Figure 19.2.4 shows the block erase operation flowchart.

(6) Erase-all-blocks command

Writing of command code “20_{16}” at the 1st bus cycle and “20_{16}” at the 2nd bus cycle initiate the automatic erase (erase and erase-verify) operation for all the blocks. During the automatic erase operation, be sure not to access the flash memory or not to execute the next command. The completion of the automatic erase operation can be recognized by the RY/$\overline{BY}$ status bit (bit 0 at address $9E_{16}$).

After the automatic erase operation is completed, the result of it can be recognized by reading out the status register. (Refer to section **“19.2.5 Full status check.”**)

Figure 19.2.5 shows the erase-all-blocks operation flowchart.

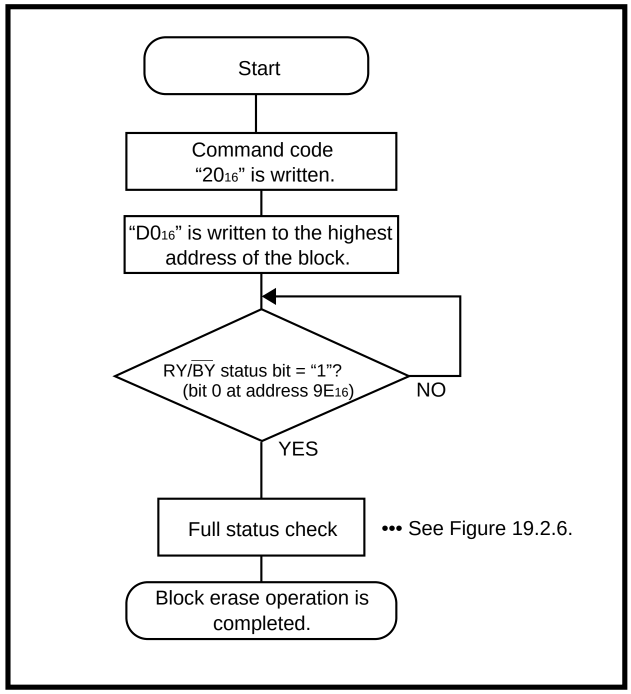


Fig. 19.2.4 Block erase operation flowchart

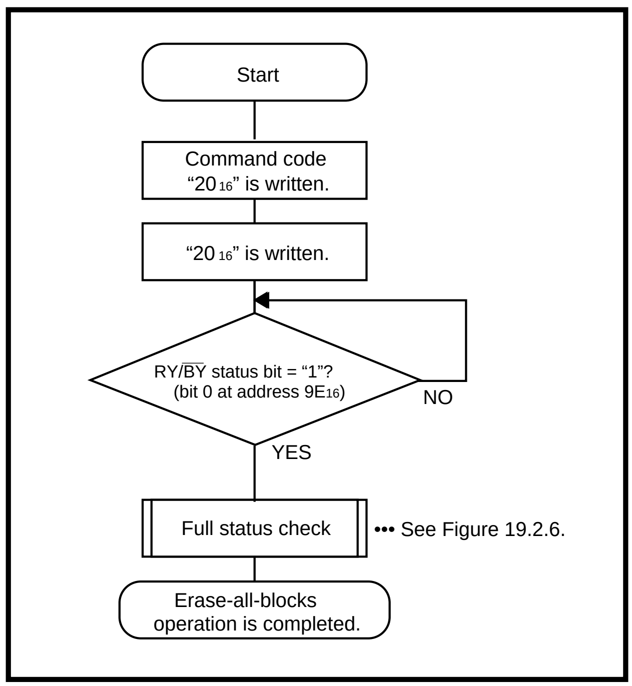


Fig. 19.2.5 Erase-all-blocks operation flowchart

19.2.5. Full status check

If an error has occurred, bits SR.4 and SR.5 of the status register are set to "1" upon completion of the programming or erase operation. Therefore, the result of the programming or erase operation can be recognized by checking these status (in other words, full status check).

Table 19.2.3 lists the errors and the states of bits SR.4 and SR.5, and Figure 19.2.6 shows the full status check flowchart and the action to be taken if any error has occurred.

Table 19.2.3 Errors and States of bits SR.3 to SR.5

Status register		Error	Error occurrence conditions
SR.5	SR.4		
1	1	Command sequence error	• Commands are not correctly written. • Data other than "D0₁₆" and "FF₁₆" is written at the 2nd bus cycle of the block erase command **(Note)**. • Data other than"20₁₆" and "FF₁₆" is written at the 2nd bus cycle of the erase-all-blocks command **(Note)**.
1	0	Erase error	• Although the block erase or erase-all-blocks command is executed, these blocks are not correctly erased.
0	1	Programming error	• Although the programming command is executed, programming is not correctly performed.

Notes: When "FF_{16}" is written at the 2nd bus cycle of any of these commands, the microcomputer enters the read array mode. Simultaneously with this, the command code written at the 1st bus cycle is cancelled.

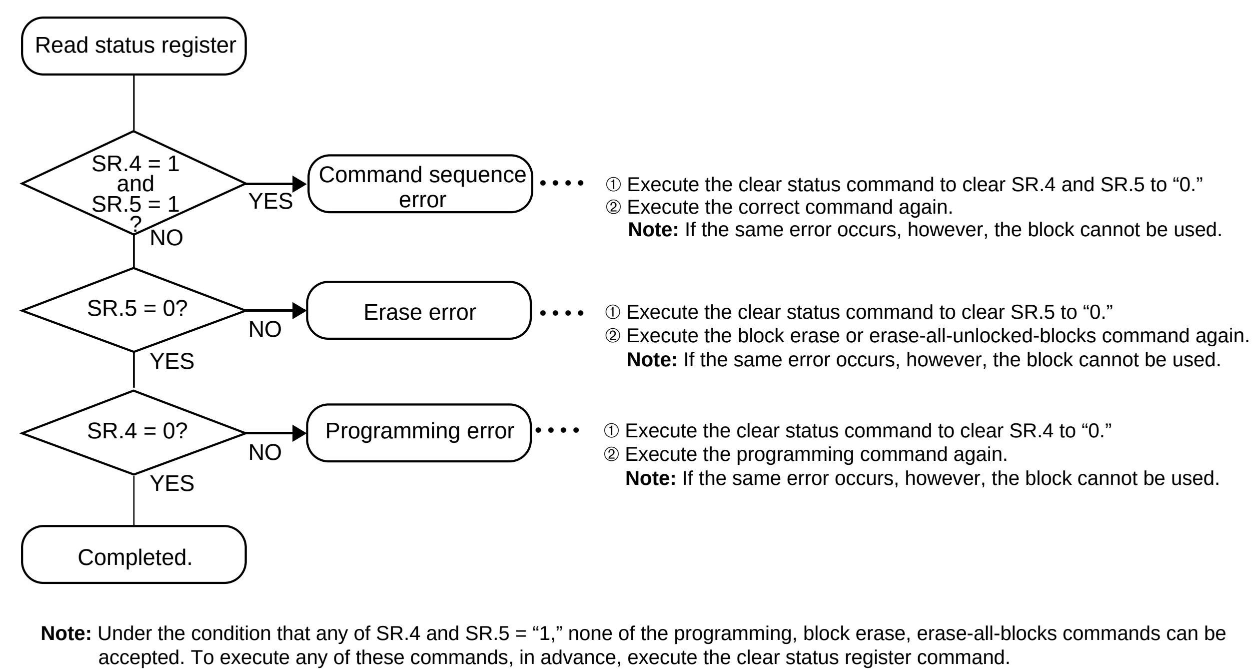


Fig. 19.2.6 Full status check flowchart and actions to be taken if any error has ocurred

19.2.6 Electrical characteristics

(1) M37905F8CFP

DC Electrical Characteristics (V_{CC} = 5 V ± 0.5 V, Ta = 0 to 60 °C, f(f_{sys}) = 20 MHz)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
I_{CC1}	V_{CC} power source current (at read)		10	30	mA
I_{CC2}	V_{CC} power source current (at write)			30	mA
I_{CC3}	V_{CC} power source current (at programming)			40	mA
I_{CC4}	V_{CC} power source current (at erasing)			40	mA

AC Electrical Characteristics (V_{CC} = 5 V ± 0.5 V, Ta = 0 to 60 °C, f(f_{sys}) = 20 MHz)

Parameter	Limits			Unit
	Min.	Typ.	Max.	
256 bytes programming time		4	40	ms
Block erase time		0.6	8	s
Erase all blocks time		0.6 × n	8 × n	s

n = Number of blocks to be erased

For the limits of parameters other than the above, refer to section **"Appendix 9. M37905M4C-XXXFP electrical characteristics."**

[Precautions for flash memory CPU reprogramming mode]

1. In the flash memory CPU reprogramming mode, an opcode cannot be fetched for the internal flash memory. Accordingly, be sure to transfer the reprogramming control software to an area other than the internal flash memory area, and then execute the software in this area. (See Figure 19.2.2.)
   Also, take consideration for instruction description (such as specified addresses, addressing modes) in the reprogramming control software since this software is to be executed in an area other than the internal flash memory area.

2. In order to prevent any interrupt occurrence during the flash memory CPU reprogramming mode, before selecting this mode, be sure to set the interrupt disable flag (I) to "1" or set the interrupt priority level to "000_2" (interrupts disabled). Also, we recommend to connect pins $\overline{P4OUT_{CUT}}$ and $\overline{P6OUT_{CUT}}$ with V_{CC} via resistors, respectively.
   Even in the flash memory CPU reprogramming mode, periodically writing to the watchdog timer is required.
   Also, an interrupt, hardware reset, or software reset, generated in the CPU reprogramming mode, makes program runaway. If a program runaway has occurred, be sure to push the microcomputer into the power-on reset state.

3. Commands and data must be read from and written into even-numbered addresses in the user ROM area, 16 bits at a time.

4. Be sure not to execute the STP instruction in the CPU reprogramming mode.

5. In order to reset the internal flash memory control circuit by using the flash memory reset bit (bit 3 at address $9E_{16}$), be sure to confirm the RY/$\overline{BY}$ status bit (bit 0 at address $9E_{16}$) becomes "1" after writing of "1" to this bit; and then, write "0" to the flash memory reset bit.

6. Addresses $FF90_{16}$ to $FF9F_{16}$ (the user ROM area) are reserved for serial and parallel programmers. Be sure not to use this area.

19.3 Flash memory serial I/O mode

In the flash memory serial I/O mode, by using a dedicated serial programmer, the contents of the user ROM area can be reprogrammed with the microcomputer mounted on the final printed circuit board. About the serial programmer concerned, consult its manufacturer, and for more information on using it, refer to the user's manual of the serial programmer.
Note that if the boot ROM area has been reprogrammed in the flash memory parallel I/O mode, the flash memory serial I/O mode cannot be used. (Refer to section **"19.4 Flash memory parallel I/O mode."**)
Addresses $FF90_{16}$ to $FF9F_{16}$ (the user ROM area) are reserved for serial or parallel programmers. Therefore, be sure not to use to this area.

19.3.1. Pin description

Table 19.3.1 lists the pin description in the flash memory serial I/O mode, and each of Figures 19.3.1 and 19.3.2 shows the pin configuration in this mode.

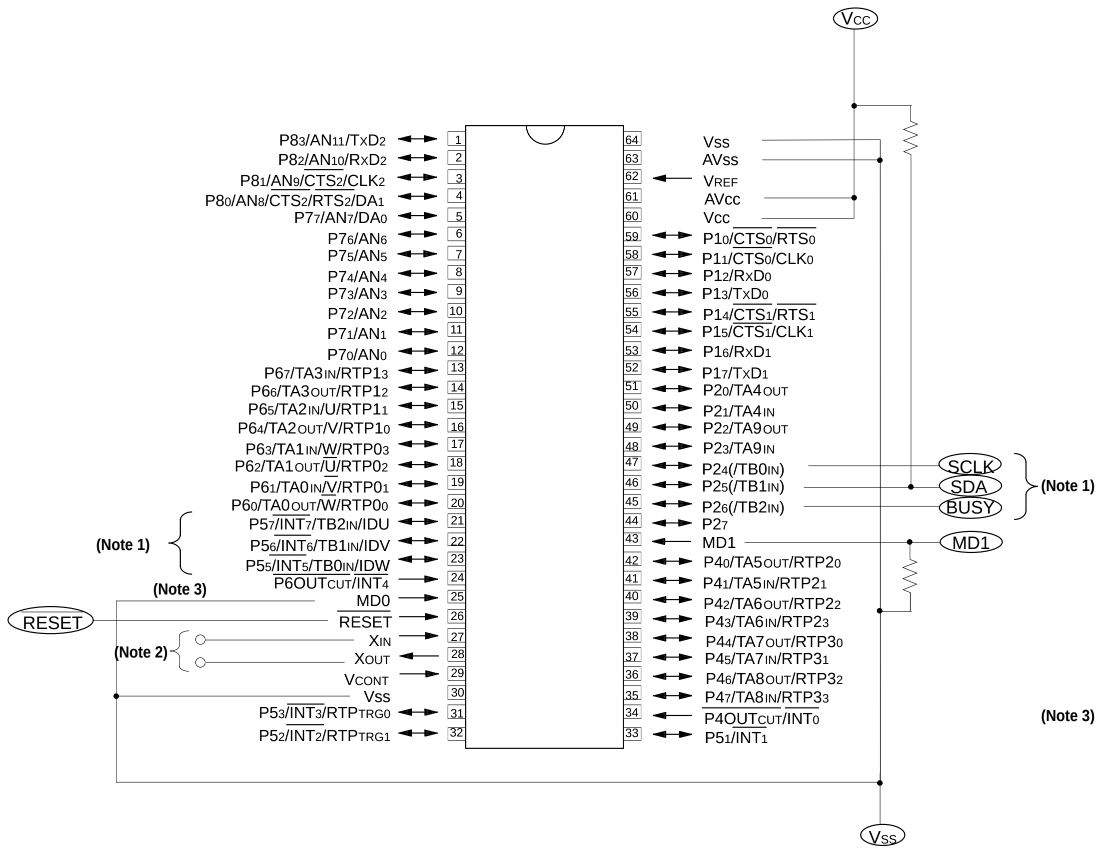
19.3 Flash memory serial I/O mode

Table 19.3.1 Pin description in flash memory serial I/O mode

Pin	Name	Input/Output	Functions
V_{CC}	Power supply input		Supply V_{CC} level voltage to pin Vcc.
V_{SS}			Supply V_{SS} level voltage to pin Vss.
MD0	MD0	Input	Connect this pin to V_{SS}.
MD1	MD1	Input	Connect this pin to V_{SS} via a resistor (about 10 kΩ to 100 kΩ).
$\overline{\text{RESET}}$	Reset input	Input	The reset input pin **(Note 1)**.
X_{IN}	Clock input	Input	Connect a ceramic resonator or quartz-crystal oscillator between X_{IN} and X_{OUT} pins. When using an external clock, the clcok source must be input to X_{IN} pin and X_{OUT} pin must be left open.
X_{OUT}	Clock output	Output	
V_{CONT}	Filter circuit connection	—	The V_{CONT} pin. (Not used in this mode.)
AV_{CC}	Analog supply input		Connect this pin to V_{CC}.
AV_{SS}			Connect this pin to V_{SS}.
V_{REF}	Reference voltage input	Input	The V_{REF} pin. (Not used in this mode.)
$P1_0$ to $P1_7$	Input port P1	Input	Input port pins. (Not used in this mode.)
$P2_0$ to $P2_3$, $P2_7$	Input port P2	Input	
$P2_4$	SCLK input	Input	The input pin for a serial clock.
$P2_5$	SDA I/O	I/O	The I/O pin for serial data. This pin must be connected with V_{CC} via a resistor (about 1 kΩ).
$P2_6$	BUSY output	Output	The BUSY signal output pin.
$P4_0$ to $P4_7$	Input port P4	Input	Input port pins. (Not used in this mode.)
$P5_1$ to $P5_3$, $P5_5$ to $P5_7$	Input port P5	Input	
$P6_0$ to $P6_7$	Input port P6	Input	
$P7_0$ to $P7_7$	Input port P7	Input	
$P8_0$ to $P8_3$	Input port P8	Input	
$\overline{\text{P4OUT}_{\text{CUT}}}$	$\overline{\text{P4OUT}_{\text{CUT}}}$ input	Input	The $\overline{\text{P4OUT}_{\text{CUT}}}$ pin. (Not used in this mode.) Recommended to be connected with V_{CC} via a resistor.
$\overline{\text{P6OUT}_{\text{CUT}}}$	$\overline{\text{P6OUT}_{\text{CUT}}}$ input	Input	The $\overline{\text{P6OUT}_{\text{CUT}}}$ pin. (Not used in this mode.) Recommended to be connected with V_{CC} via a resistor.

Notes 1: When there is a possibility that the user reset signal becomes “L” level in the flash memory serial I/O mode, be sure to cut off the current flow between the user reset signal and pin $\overline{\text{RESET}}$ by using a jumper switch, etc. (Refer to section **“19.3.2 Examples of handling control pins in flash memory serial I/O mode.”**)

2: For pins not used in the flash memory serial I/O mode, properly connect to somewhere in the user system. For pins not used in the user system, handle them with reference to section **“5.3 Examples of handling unused pins.”** For pins used in the flash memory serial I/O mode, handle them with reference to section **“19.3.2 Examples of handling control pins in flash memory serial I/O mode.”**



Notes 1: Allocation of pins TB0IN to TB2IN can be switched by software.
2: Connected to the oscillation circuit.
3: Recommended to be connected with VCC via a resistor.

(oval) : Connected to a serial programmer.

Outline 64P4B

Fig. 19.3.1 Pin connection in flash memory serial I/O mode (Outline: 64P4B)

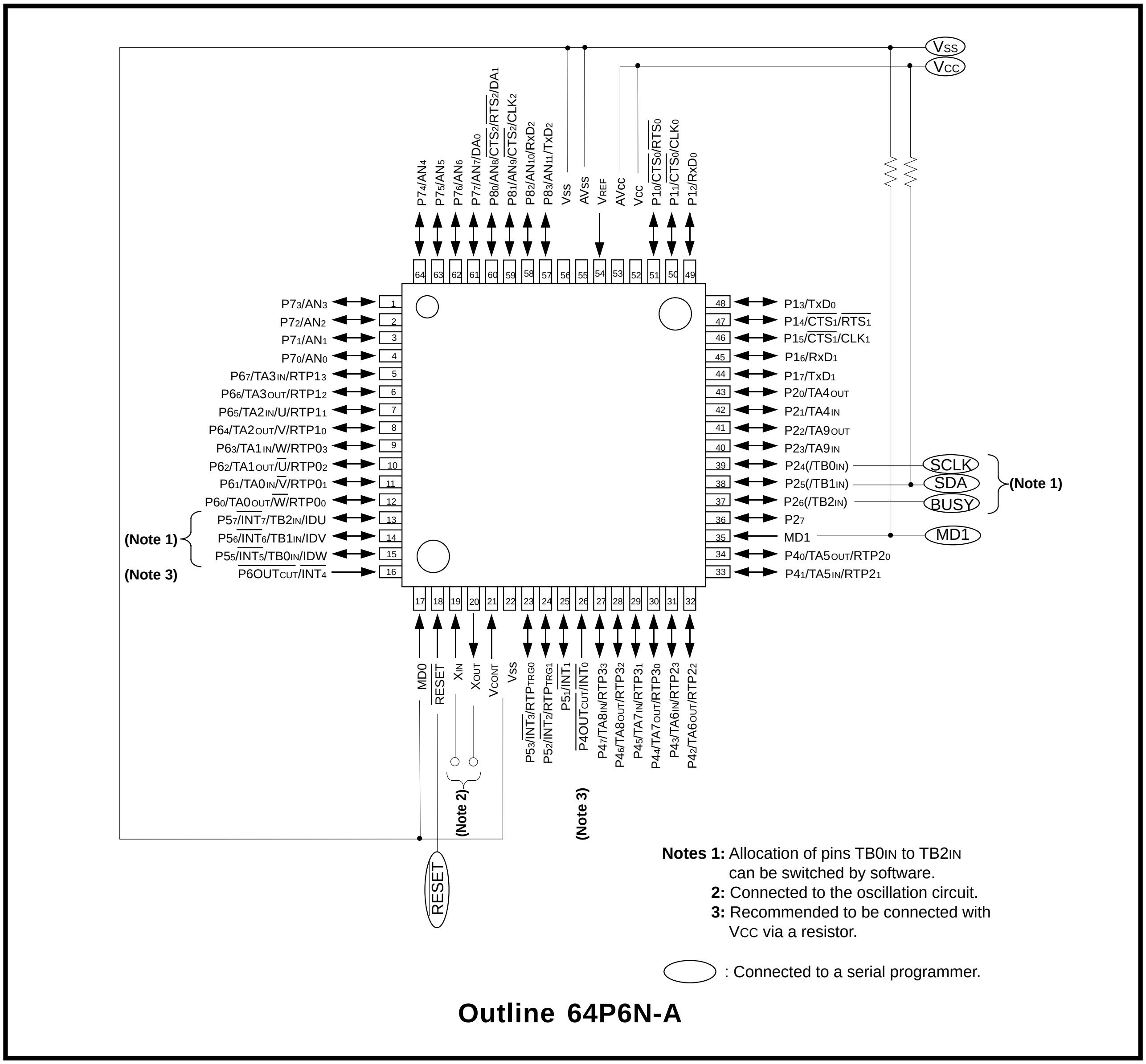


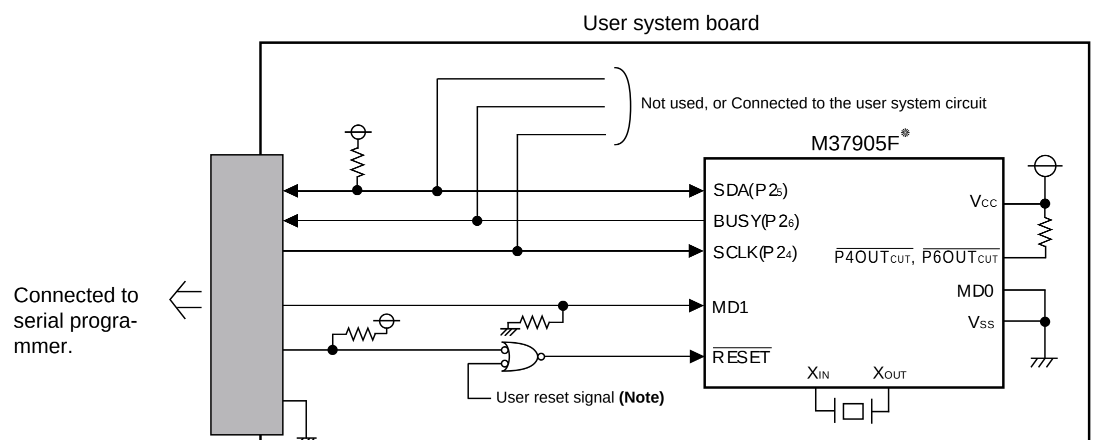
Fig. 19.3.2 Pin connection in flash memory serial I/O mode (Outline: 64P6N-A)

19.3.2. Example of handling control pins in flash memory serial I/O mode

Each of pins $P2_4$ to $P2_6$, MD0, and MD1 serves as an input/output pin for a control signal in the flash memory serial I/O mode. Examples of handling these pins and pin $\overline{RESET}$ on the board are described below.

(1) With control signals not affecting user system circuit

When control signals in the flash memory serial I/O mode are not used in the user system circuit, or when these signals do not affect that circuit, the connections shown in Figure 19.3.3 are available. When pins $\overline{P4OUT_{CUT}}$ and $\overline{P6OUT_{CUT}}$, however, are used in the user system circuit, see Figures 19.3.4 and 19.3.5.



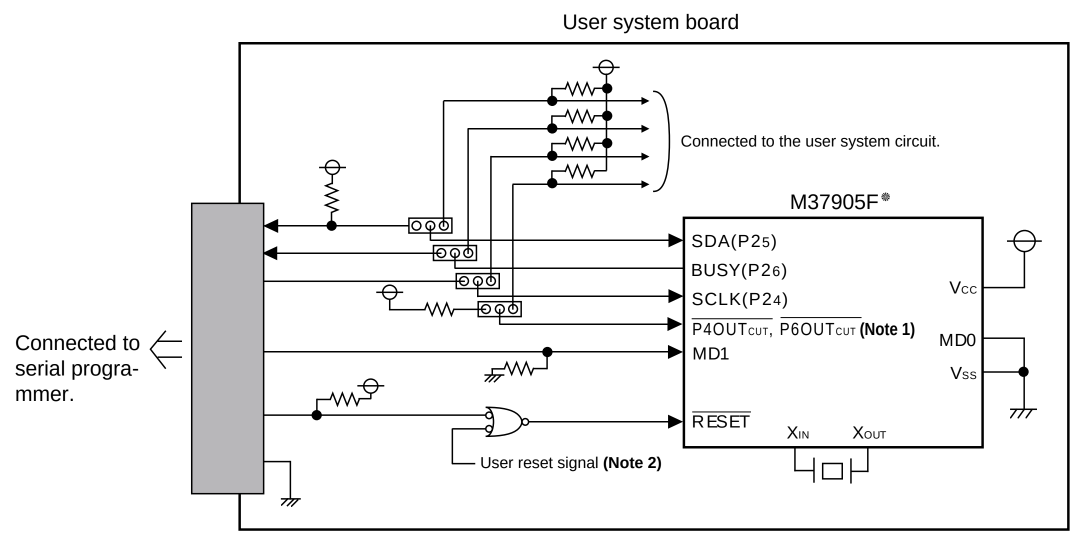
Note: When there is a possibility that the user reset signal becomes "L" level in the flash memory serial I/O mode, be sure to cut the current flow between the user reset pin and pin $\overline{RESET}$ by using a jumper switch, etc.

✻: The flash memory version of the 7905 Group

Fig. 19.3.3 Example of handing control pins when control signals do not affect user system circuit

19.3 Flash memory serial I/O mode

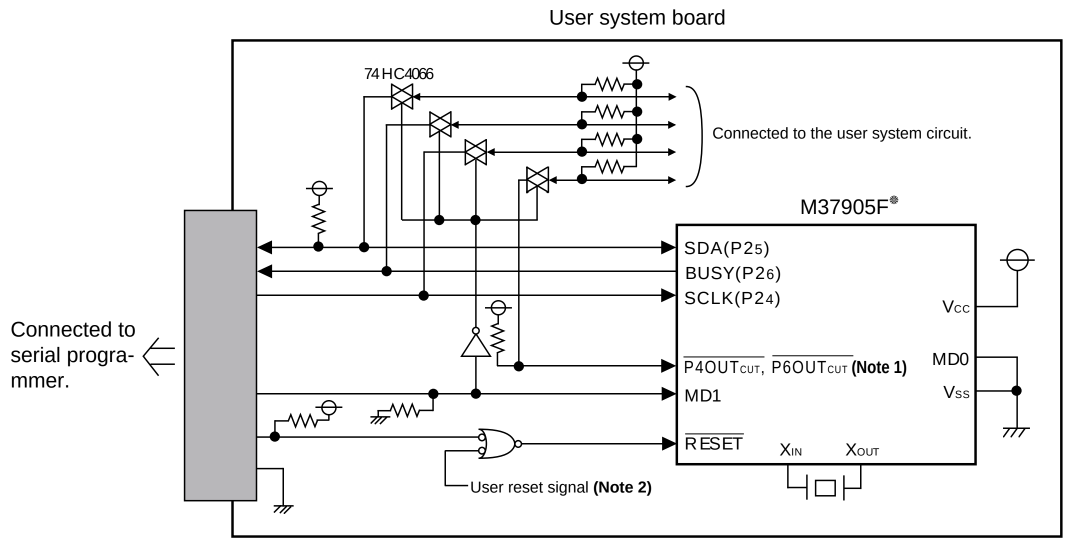
(2) With control signals affecting user system circuit
In the flash memory serial I/O mode, be sure to cut the current flow toward the user system circuit if control signals for this mode are also used in the user system circuit. Figure 19.3.4 shows an example of handling pins with jumper switches used, and Figure 19.3.5 shows an example of handling pins with analog switches used.



Note 1: Recommended to be connected with VCC via a resistor.
2: When there is a possibility that the user reset signal becomes "L" level in the flash memory serial I/O mode, be sure to cut the current flow between the user reset pin and pin RESET by using a jumper switch, etc.

✻: The flash memory version of the 7905 Group

Fig. 19.3.4 Example of handling pins with jumper switches used



Note 1: Recommended to be connected with VCC via a resistor.
2: When there is a possibility that the user reset signal becomes "L" level in the flash memory serial I/O mode, be sure to cut the current flow between the user reset pin and pin RESET by using a jumper switch, etc.

✻: The flash memory version of the 7905 Group

Fig. 19.3.5 Example of handling pins with analog switches used

[Precautions for flash memory serial I/O mode]

1. If the boot ROM area has been reprogrammed in the flash memory parallel I/O mode, the flash memory serial I/O mode cannot be used.

2. In the flash memory serial I/O mode, we recommend to connect pins $\overline{P4OUT_{CUT}}$ and $\overline{P6OUT_{CUT}}$ with V_{CC} via resistors, respectively. (Refer to section **"19.3.2 Examples of handling control pins in flash memory serial I/O mode."**)

3. When there is a possibility that the user reset signal becomes "L" level in the flash memory serial I/O mode, be sure to cut the current flow between the user reset pin and pin $\overline{RESET}$ by using a jumper switch, etc. (Refer to section **"19.3.2 Examples of handling control pins in flash memory serial I/O mode."**)

4. Addresses $FF90_{16}$ to $FF9F_{16}$ (the user ROM area) are reserved for serial and parallel programmers. Therefore, be sure not to use this area.

19.4 Flash memory parallel I/O mode

In the flash memory parallel I/O mode, the contents of the user ROM area and boot ROM area can be reprogrammed by using a dedicated parallel programmer. (See Figure 19.1.2.) About the parallel programmer concerned, consult its manufacturer, and for more information on using it, refer to the user's manual of the parallel programmer.
In the flash memory parallel I/O mode, the boot ROM area is assigned to addresses 0_{16} to $1FFFF_{16}$ (word addresses).
Note that if the boot ROM area has been reprogrammed in the flash memory parallel I/O mode, the flash memory serial I/O mode cannot be used. (Refer to section **"19.3 Flash memory serial I/O mode."**)

Also, addresses $FF90_{16}$ to $FF9F_{16}$ (the user ROM area) are reserved for serial and parallel programmers. Therefore, besure not to use this area.

[Precautions for flash memory parallel I/O mode]

1. If the boot ROM area has been reprogrammed in the flash memory parallel I/O mode, the flash memory serial I/O mode cannot be used. (Refer to section **"19.3 Flash memory serial I/O mode."**)

2. Addresses $FF90_{16}$ to $FF9F_{16}$ (the user ROM area) are reserved for serial and parallel programmers. Be sure not to use this area.

MEMORANDUM

APPENDIX

Appendix 1. Memory assigment in SFR area

■ SFR area (Addresses 0_{16} to FF_{16})

●SFR area (Addresses 0_{16} to FF_{16})

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid. It is impossible to read the bit state.
[shaded box] : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.

[0] : Always "0" at reading.
[1] : Always "1" at reading.
[?] : Always undefined at reading.
[0 hatched] : "0" immediately after reset. Fix this bit to "0."

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
0_{16}		(Note 1)	?
1_{16}		(Note 1)	?
2_{16}		(Note 2)	?
3_{16}	Port P1 register	RW	?
4_{16}		(Note 2)	?
5_{16}	Port P1 direction register	RW	00_{16}
6_{16}	Port P2 register	RW	?
7_{16}		(Note 2)	?
8_{16}	Port P2 direction register	RW	00_{16}
9_{16}		(Note 2)	?
A_{16}	Port P4 register	RW	?
B_{16}	Port P5 register	RW, [ ], RW, [ ]	? ? ? [?] ? ? ? [?]
C_{16}	Port P4 direction register	RW	00_{16}
D_{16}	Port P5 direction register	RW, [ ], RW, [ ]	0 0 0 [?] 0 0 0 [?]
E_{16}	Port P6 register	RW	?
F_{16}	Port P7 register	RW	?
10_{16}	Port P6 direction register	RW	00_{16}
11_{16}	Port P7 direction register	RW	00_{16}
12_{16}	Port P8 register	[ ], RW	[?] [?] [?] [?] 0 0 0 0
13_{16}			?
14_{16}	Port P8 direction register	[ ], RW	[?] [?] [?] [?] 0 0 0 0
15_{16}			?
16_{16}		(Note 2)	?
17_{16}		(Note 2)	?
18_{16}		(Note 2)	?
19_{16}		(Note 2)	?
$1A_{16}$			?
$1B_{16}$			?
$1C_{16}$			?
$1D_{16}$			?
$1E_{16}$	A-D control register 0	RW	0 0 [0 hatched] 0 0 ? ? ?
$1F_{16}$	A-D control register 1	[ ], RW	[0] 0 [0 hatched] 0 0 0 1 1

Notes 1: Do not read from and write to this register.
2: Do not write to this register.

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid. It is impossible to read the bit state.
(shaded box) : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.

[0] : Always "0" at reading.
[1] : Always "1" at reading.
[?] : Always undefined at reading.
[0] (hatched) : "0" immediately after reset. Fix this bit to "0."

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
20_{16}	A-D register 0	**(Note 3)**	?
21_{16}		**(Note 3)**	0 0 0 0 0 0 ?
22_{16}	A-D register 1	**(Note 3)**	?
23_{16}		**(Note 3)**	0 0 0 0 0 0 ?
24_{16}	A-D register 2	**(Note 3)**	?
25_{16}		**(Note 3)**	0 0 0 0 0 0 ?
26_{16}	A-D register 3	**(Note 3)**	?
27_{16}		**(Note 3)**	0 0 0 0 0 0 ?
28_{16}	A-D register 4	**(Note 3)**	?
29_{16}		**(Note 3)**	0 0 0 0 0 0 ?
$2A_{16}$	A-D register 5	**(Note 3)**	?
$2B_{16}$		**(Note 3)**	0 0 0 0 0 0 ?
$2C_{16}$	A-D register 6	**(Note 3)**	?
$2D_{16}$		**(Note 3)**	0 0 0 0 0 0 ?
$2E_{16}$	A-D register 7	**(Note 3)**	?
$2F_{16}$		**(Note 3)**	0 0 0 0 0 0 ?
30_{16}	UART0 transmit/receive mode register	RW	00_{16}
31_{16}	UART0 baud rate register (BRG0)	WO	?
32_{16}	UART0 transmit buffer register	WO	?
33_{16}		WO	?
34_{16}	UART0 transmit/receive control register 0	RW RO RW	0 0 0 0 1 0 0 0
35_{16}	UART0 transmit/receive control register 1	RO RW RO RW	0 0 0 0 0 0 1 0
36_{16}	UART0 receive buffer register	RO	?
37_{16}		RO	0 0 0 0 0 0 0 ?
38_{16}	UART1 transmit/receive mode register	RW	00_{16}
39_{16}	UART1 baud rate register (BRG1)	WO	?
$3A_{16}$	UART1 transmit buffer register	WO	?
$3B_{16}$		WO	?
$3C_{16}$	UART1 transmit/receive control register 0	RW RO RW	0 0 0 0 1 0 0 0
$3D_{16}$	UART1 transmit/receive control register 1	RO RW RO RW	0 0 0 0 0 0 1 0
$3E_{16}$	UART1 receive buffer register	RO	?
$3F_{16}$		RO	0 0 0 0 0 0 0 ?

Note 3: The access characteristics at addresses 20_{16} to $2F_{16}$ vary according to the contents of the comparator function select register 0 (address DC_{16}). (Refer to **"CHAPTER 12. A-D CONVERTER."**)

APPENDIX

Appendix 1. Memory assigment in SFR area

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid. It is impossible to read the bit state.
[shaded box] : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.

[0] : Always "0" at reading.
[1] : Always "1" at reading.
[?] : Always undefined at reading.
[hatched 0] : "0" immediately after reset. Fix this bit to "0."

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
40_{16}	Count start register 0	b7–b0: RW	00_{16}
41_{16}	Count start register 1	b7–b5: nothing assigned; b4–b0: RW	b7–b5: [?]; b4–b0: 0 0 0 0 0
42_{16}	One-shot start register 0	b7: RW; b6–b5: nothing assigned; b4–b0: WO	b7: [hatched 0]; b6–b5: [?]; b4–b0: [0] [0] [0] [0] [0]
43_{16}	One-shot start register 1	b7: RW; b6–b5: nothing assigned; b4–b0: WO	b7: [hatched 0]; b6–b5: [?]; b4–b0: [0] [0] [0] [0] [0]
44_{16}	Up-down register 0	b7–b5: WO; b4–b0: RW	[0] [0] [0] 0 0 0 0 0
45_{16}	Timer A clock division select register	b7–b2: nothing assigned; b1: RW; b0: RW	[0] [0] [0] [0] [0] [0] 0 0
46_{16}	Timer A0 register	(Note 4)	?
47_{16}		(Note 4)	?
48_{16}	Timer A1 register	(Note 4)	?
49_{16}		(Note 4)	?
$4A_{16}$	Timer A2 register	(Note 4)	?
$4B_{16}$		(Note 4)	?
$4C_{16}$	Timer A3 register	(Note 4)	?
$4D_{16}$		(Note 4)	?
$4E_{16}$	Timer A4 register	(Note 4)	?
$4F_{16}$		(Note 4)	?
50_{16}	Timer B0 register	(Note 5)	?
51_{16}		(Note 5)	?
52_{16}	Timer B1 register	(Note 5)	?
53_{16}		(Note 5)	?
54_{16}	Timer B2 register	(Note 5)	?
55_{16}		(Note 5)	?
56_{16}	Timer A0 mode register	RW	00_{16}
57_{16}	Timer A1 mode register	RW	00_{16}
58_{16}	Timer A2 mode register	RW	00_{16}
59_{16}	Timer A3 mode register	RW	00_{16}
$5A_{16}$	Timer A4 mode register	RW	00_{16}
$5B_{16}$	Timer B0 mode register	b7–b6: RW; b5: (Note 6); b4–b0: RW	0 0 ? 0 0 0 0 0
$5C_{16}$	Timer B1 mode register	b7–b6: RW; b5: (Note 6); b4–b0: RW	0 0 ? 0 0 0 0 0
$5D_{16}$	Timer B2 mode register	b7–b6: RW; b5: (Note 6); b4–b0: RW	0 0 ? 0 0 0 0 0
$5E_{16}$	Processor mode register 0	b7: RW; b6: WO; b5–b0: RW	[hatched 0] 0 0 0 1 0 0 0
$5F_{16}$	Processor mode register 1	RW	0 [hatched 0] [hatched 0] [hatched 0] [hatched 0] [hatched 0] 0 1

Notes 4: The access characteristics at addresses 46_{16} to $4F_{16}$ vary according to the timer A's operating mode. (Refer to **"CHAPTER 7. TIMER A."**)
5: The access characteristics at addresses 50_{16} to 55_{16} vary according to the timer B's operating mode. (Refer to **"CHAPTER 8. TIMER B."**)
6: The access characteristics for bit 5 at addresses $5B_{16}$ to $5D_{16}$ vary according to the timer B's operating mode. (Refer to **"CHAPTER 8. TIMER B."**)

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid. It is impossible to read the bit state.
[ ] : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.

[0] : Always "0" at reading.
[1] : Always "1" at reading.
[?] : Always undefined at reading.
[0 (hatched)] : "0" immediately after reset. Fix this bit to "0."

Address	Register name	Access b7	b6	b5	b4	b3	b2	b1	Access b0	Reset b7	b6	b5	b4	b3	b2	b1	Reset b0
60₁₆	Watchdog timer register	(Note 7)								? (Note 8)							
61₁₆	Watchdog timer frequency select register	RW	RW	—					RW	0	0	?					0
62₁₆	Particular function select register 0	RW						RW (Note 9)		0 (fix 0)	0 (fix 0)	0 (fix 0)	0 (fix 0)	0 (fix 0)	0 (fix 0)	0	0
63₁₆	Particular function select register 1	—	RW	—	RW	RW	RW	(Note 10)		0	0	0	0 (fix 0)	0	0 (fix 0)	(Note 11)	
64₁₆	Particular function select register 2	—								?							
65₁₆		(Note 12)								?							
66₁₆	Debug control register 0	—	RW							1	0 (fix 0)	(Note 11)	0 (fix 0)	0 (fix 0)	(Note 11)		
67₁₆	Debug control register 1	—	RO	RO	—	RW	RW	RO	RW	0	0	0	?	0	0	0	0 (fix 0)
68₁₆	Address comparison register 0	RW (Note 13)								?							
69₁₆	Address comparison register 0	RW (Note 13)								?							
6A₁₆	Address comparison register 0	RW (Note 13)								?							
6B₁₆	Address comparison register 1	RW (Note 13)								?							
6C₁₆	Address comparison register 1	RW (Note 13)								?							
6D₁₆	Address comparison register 1	RW (Note 13)								?							
6E₁₆	$\overline{INT_3}$ interrupt control register	—		RW						?		0	0	0	0	0	0
6F₁₆	$\overline{INT_4}$ interrupt control register	—		RW						?		0	0	0	0	0	0
70₁₆	A-D conversion interrupt control register	—				RW				?				?	0	0	0
71₁₆	UART0 transmit interrupt control register	—				RW				?				0	0	0	0
72₁₆	UART0 receive interrupt control register	—				RW				?				0	0	0	0
73₁₆	UART1 transmit interrupt control register	—				RW				?				0	0	0	0
74₁₆	UART1 receive interrupt control register	—				RW				?				0	0	0	0
75₁₆	Timer A0 interrupt control register	—				RW				?				0	0	0	0
76₁₆	Timer A1 interrupt control register	—				RW				?				0	0	0	0
77₁₆	Timer A2 interrupt control register	—				RW				?				0	0	0	0
78₁₆	Timer A3 interrupt control register	—				RW				?				0	0	0	0
79₁₆	Timer A4 interrupt control register	—				RW				?				0	0	0	0
7A₁₆	Timer B0 interrupt control register	—				RW				?				0	0	0	0
7B₁₆	Timer B1 interrupt control register	—				RW				?				0	0	0	0
7C₁₆	Timer B2 interrupt control register	—				RW				?				0	0	0	0
7D₁₆	$\overline{INT_0}$ interrupt control register	—		RW						?		0	0	0	0	0	0
7E₁₆	$\overline{INT_1}$ interrupt control register	—		RW						?		0	0	0	0	0	0
7F₁₆	$\overline{INT_2}$ interrupt control register	—		RW						?		0	0	0	0	0	0

Notes 7 : By writing dummy data to address 60₁₆, a value of "FFF₁₆" is set to the watchdog timer. The dummy data is not retained anywhere.
8 : A value of "FFF₁₆" is set to the watchdog timer. (Refer to **"CHAPTER 14. WATCHDOG TIMER."**)
9 : After writing "55₁₆" to address 62₁₆, each bit must be set.
10 : It is possible to read the bit state at reading. By writing "0" to this bit, this bit becomes "0." But when writing "1" to this bit, this bit will not change.
11 : This bit becomes "0" at power-on reset. This bit retains the state immediately before reset in the case of hardware reset and software reset.
12 : Do not write to this register.
13 : When these registers are accessed, set the address comparison register access enable bit (bit 2 at address 67₁₆) to "1." (Refer to **"CHAPTER 17. DEBUG FUNCTION."**)

APPENDIX

Appendix 1. Memory assigment in SFR area

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid. It is impossible to read the bit state.
[shaded box] : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.

[0] : Always "0" at reading.
[1] : Always "1" at reading.
[?] : Always undefined at reading.
[hatched 0] : "0" immediately after reset. Fix this bit to "0."

Access characteristics and state immediately after reset are given per bit, b7 (left) to b0 (right); a value in the b7 column alone applies to the whole register.

Address	Register name	Access b7	b6	b5	b4	b3	b2	b1	Access b0	Reset b7	b6	b5	b4	b3	b2	b1	Reset b0
80_{16}		**(Note 14)**								?							
81_{16}		**(Note 14)**								?							
82_{16}		**(Note 14)**								?							
83_{16}		**(Note 14)**								?							
84_{16}		**(Note 14)**								?							
85_{16}		**(Note 14)**								?							
86_{16}		**(Note 14)**								?							
87_{16}		**(Note 14)**								?							
88_{16}										?							
89_{16}										?							
$8A_{16}$		**(Note 14)**								?							
$8B_{16}$										?							
$8C_{16}$		**(Note 14)**								?							
$8D_{16}$										?							
$8E_{16}$		**(Note 14)**								?							
$8F_{16}$										?							
90_{16}		**(Note 14)**								?							
91_{16}										?							
92_{16}		**(Note 14)**								?							
93_{16}										?							
94_{16}										?							
95_{16}	External interrupt input read-out register	RO								?							
96_{16}	D-A control register							RW	RW	?						0	0
97_{16}										?							
98_{16}	D-A register 0	RW								00_{16}							
99_{16}	D-A register 1	RW								00_{16}							
$9A_{16}$										?							
$9B_{16}$										?							
$9C_{16}$		**(Note 14)**								?							
$9D_{16}$		**(Note 14)**								?							
$9E_{16}$	Flash memory control register **(Note 15)**			RW		RW		RW	RO	0	0	0	0	0	0	0	1
$9F_{16}$										?							

Notes 14 : Do not write to this register.
15 : This register is assigned only to the flash memory version. (Refer to **"CHAPTER 19. FLASH MEMORY VERSION."**) Nothing is assigned here in the mask ROM version.

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid. It is impossible to read the bit state.
[shaded box] : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.

[0] : Always "0" at reading.
[1] : Always "1" at reading.
[?] : Always undefined at reading.
[0 hatched] : "0" immediately after reset. Fix this bit to "0."

(In the table below, values are given from b7 to b0; "–" indicates a bit with nothing assigned (shaded); "(hatched)" indicates a hatched bit.)

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
A0₁₆	Pulse output control register	RW	00₁₆
A1₁₆		– (shaded)	?
A2₁₆	Pulse output data register 0	RW	00₁₆
A3₁₆		– (shaded)	?
A4₁₆	Pulse output data register 1	RW	00₁₆
A5₁₆		– (shaded)	?
A6₁₆	Waveform output mode register	RW	00₁₆
A7₁₆	Dead-time timer	WO	?
A8₁₆	Three-phase output data register 0	RW	00₁₆
A9₁₆	Three-phase output data register 1	RW	00₁₆
AA₁₆	Position-data-retain function control register	– – – – RW RO RO RO	? (b7–b4) 0 0 0 0
AB₁₆		– (shaded)	?
AC₁₆	Serial I/O pin control register	– – RW RW RW RW RW RW	0 0 0 0 0 0 0 0
AD₁₆		– (shaded)	?
AE₁₆	Port P2 pin function control register	RW – – – – RW RW RW	0 (hatched) ? ? ? ? 0 0 0
AF₁₆		– (shaded)	?
B0₁₆	UART2 transmit/receive mode register	RW	00₁₆
B1₁₆	UART2 baud rate register (BRG2)	WO	?
B2₁₆	UART2 transmit buffer register	WO	?
B3₁₆	(UART2 transmit buffer register)	– – – – – – – WO	?
B4₁₆	UART2 transmit/receive control register 0	RW RW RW RW RO RW RW RW	0 0 0 0 1 0 0 0
B5₁₆	UART2 transmit/receive control register 1	RO RO RO RO RO RW RO RW	0 0 0 0 0 0 1 0
B6₁₆	UART2 receive buffer register	RO	?
B7₁₆	(UART2 receive buffer register)	– – – – – – – RO	0 0 0 0 0 0 0 ?
B8₁₆		**(Note 16)**	?
B9₁₆		– (shaded)	?
BA₁₆		**(Note 16)**	?
BB₁₆		**(Note 16)**	?
BC₁₆	Clock control register 0	RW RW RW RW **(Note 17)** (b3–b2) RW RW	0 0 0 1 (hatched) 0 1 1 1 (hatched)
BD₁₆		**(Note 16)**	?
BE₁₆		**(Note 16)**	?
BF₁₆		**(Note 16)**	?

Notes 16 : Do not write to this register.
17 : After reset, these bits are allowed to be changed only once.

APPENDIX

Appendix 1. Memory assigment in SFR area

Access characteristics

- RW : It is possible to read the bit state at reading. The written value becomes valid.
- RO : It is possible to read the bit state at reading. The written value becomes invalid.
- WO : The written value becomes valid. It is impossible to read the bit state.
- (shaded box) : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

- 0 : "0" immediately after reset.
- 1 : "1" immediately after reset.
- ? : Undefined immediately after reset.
- [0] : Always "0" at reading.
- [1] : Always "1" at reading.
- [?] : Always undefined at reading.
- [0] (hatched) : "0" immediately after reset. Fix this bit to "0."

Address	Register name	Access characteristics (b7 … b0)	State immediately after reset (b7 … b0)
$C0_{16}$		(nothing assigned)	?
$C1_{16}$		(nothing assigned)	?
$C2_{16}$		(nothing assigned)	?
$C3_{16}$		(nothing assigned)	?
$C4_{16}$	Up-down register 1	WO | RW	0 0 0 0 0 0 0 0
$C5_{16}$		(nothing assigned)	?
$C6_{16}$	Timer A5 register	**(Note 18)**	?
$C7_{16}$		**(Note 18)**	?
$C8_{16}$	Timer A6 register	**(Note 18)**	?
$C9_{16}$		**(Note 18)**	?
CA_{16}	Timer A7 register	**(Note 18)**	?
CB_{16}		**(Note 18)**	?
CC_{16}	Timer A8 register	**(Note 18)**	?
CD_{16}		**(Note 18)**	?
CE_{16}	Timer A9 register	**(Note 18)**	?
CF_{16}		**(Note 18)**	?
$D0_{16}$	Timer $A0_1$ register	WO	?
$D1_{16}$		WO	?
$D2_{16}$	Timer $A1_1$ register	WO	?
$D3_{16}$		WO	?
$D4_{16}$	Timer $A2_1$ register	WO	?
$D5_{16}$		WO	?
$D6_{16}$	Timer A5 mode register	RW	00_{16}
$D7_{16}$	Timer A6 mode register	RW	00_{16}
$D8_{16}$	Timer A7 mode register	RW	00_{16}
$D9_{16}$	Timer A8 mode register	RW	00_{16}
DA_{16}	Timer A9 mode register	RW	00_{16}
DB_{16}	A-D control register 2	RW	0 (fixed) 0 (fixed) 0 (fixed) 0 (fixed) 0 0 0 0
DC_{16}	Comparator function select register 0	RW	0 0 0 0 0 0 0 0
DD_{16}	Comparator function select register 1	RW	0 (fixed) 0 (fixed) 0 (fixed) 0 (fixed) 0 0 0 0
DE_{16}	Comparator result register 0	RW	0 0 0 0 0 0 0 0
DF_{16}	Comparator result register 1	RW	0 (fixed) 0 (fixed) 0 (fixed) 0 (fixed) 0 0 0 0

Note 18: The access characteristics at addresses $C6_{16}$ to CF_{16} vary according to the timer A's operating mode. (Refer to **"CHAPTER 7. TIMER A."**)

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid. It is impossible to read the bit state.
[shaded] : Nothing is assigned. It is impossible to read the bit state. The written value becomes invalid.

State immediately after reset

0 : "0" immediately after reset.
1 : "1" immediately after reset.
? : Undefined immediately after reset.

[0] : Always "0" at reading.
[1] : Always "1" at reading.
[?] : Always undefined at reading.
[0, hatched] : "0" immediately after reset. Fix this bit to "0."

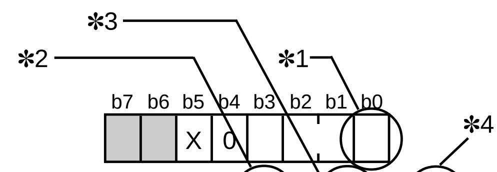
Address	Register name	Access characteristics (b7–b0)	State immediately after reset (b7–b0)
$E0_{16}$	A-D register 8	(Note 19)	?
$E1_{16}$		(Note 19)	0 0 0 0 0 0 ?
$E2_{16}$	A-D register 9	(Note 19)	?
$E3_{16}$		(Note 19)	0 0 0 0 0 0 ?
$E4_{16}$	A-D register 10	(Note 19)	?
$E5_{16}$		(Note 19)	0 0 0 0 0 0 ?
$E6_{16}$	A-D register 11	(Note 19)	?
$E7_{16}$		(Note 19)	0 0 0 0 0 0 ?
$E8_{16}$		(Note 20)	?
$E9_{16}$		(Note 20)	?
EA_{16}		(Note 20)	?
EB_{16}		(Note 20)	?
EC_{16}		(Note 20)	?
ED_{16}		(Note 20)	?
EE_{16}		(Note 20)	?
EF_{16}		(Note 20)	?
$F0_{16}$			?
$F1_{16}$	UART2 transmit interrupt control register	RW	? 0 0 0 0
$F2_{16}$	UART2 receive interrupt control register	RW	? 0 0 0 0
$F3_{16}$			?
$F4_{16}$			?
$F5_{16}$	Timer A5 interrupt control register	RW	? 0 0 0 0
$F6_{16}$	Timer A6 interrupt control register	RW	? 0 0 0 0
$F7_{16}$	Timer A7 interrupt control register	RW	? 0 0 0 0
$F8_{16}$	Timer A8 interrupt control register	RW	? 0 0 0 0
$F9_{16}$	Timer A9 interrupt control register	RW	? 0 0 0 0
FA_{16}		(Note 20)	?
FB_{16}		(Note 20)	?
FC_{16}		(Note 20)	?
FD_{16}	$\overline{INT_5}$ interrupt control register	RW	? 0 0 0 0 0 0
FE_{16}	$\overline{INT_6}$ interrupt control register	RW	? 0 0 0 0 0 0
FF_{16}	$\overline{INT_7}$ interrupt control register	RW	? 0 0 0 0 0 0

Notes 19: The access characteristics at addresses $E0_{16}$ to $E7_{16}$ vary according to the contents of the comparator function select register 1 (address DD_{16}). (Refer to **"CHAPTER 12. A-D CONVERTER."**)
20: Do not write to this register.

Appendix 2. Control registers

The control registers allocated in the SFR area are shown on the following pages.
Below is the structure diagram for all registers.

XXX register (address XX_{16})



Bit	Bit name	Function	At reset	R/W	Reference
0	• • • select bit	0 : … 1 : … The value is “0” at reading.	Undefined	WO	3-10
1	• • • select bit	b2 b1 0 0 : … 0 1 : … 1 0 : … 1 1 : …	0	RW	3-11
2			0	RW	
3	• • • flag	0 : … 1 : …	0	RO	2-6
4	Fix this bit to “0.”		0	RW	
5	This bit is invalid in … mode.		0	RW	
6	Nothing is assigned.		Undefined	–	
7	The value is “0” at reading.		0	–	

✻6

✻1

- Blank : Set to “0” or “1” according to the usage.
- 0 : Set to “0” at writing.
- 1 : Set to “1” at writing.
- × : Invalid depending on the mode or state. It may be “0” or “1.”
- (shaded box) : Nothing is assigned.

✻2

- 0 : “0” immediately after reset.
- 1 : “1” immediately after reset.
- Undefined : Undefined immediately after reset.

✻3

- RW : It is possible to read the bit state at reading. The written value becomes valid.
- RO : It is possible to read the bit state at reading. The written value becomes invalid. Accordingly, the written value may be “0” or “1.”
- WO : The written value becomes valid. It is impossible to read the bit state. The value is undefined at reading. However, when [“0” at reading”] is indicated in the “Function” or “Note” column, the bit is always “0” at reading. (See ✻5 above.)
- — : It is impossible to read the bit state. The value is undefined at reading. However, when [“0” at reading”] is indicated in the “Function” or “Note” column, the bit is always “0” at reading. (See ✻6 above.) The written value becomes invalid. Accordingly, the written value may be “0” or “1.”

✻4 Reference page for each bit.

Port Pi register (i = 1, 2, 4 to 8)
(Addresses 3_{16}, 6_{16}, A_{16}, B_{16}, E_{16}, F_{16}, 12_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Funtion	At reset	R/W	Reference
0	Port pin Pi_0	Data is input from or output to a pin by reading from or writing to the corresponding bit. 0 : "L" level 1 : "H" level	Undefined	RW	5-4
1	Port pin Pi_1		Undefined	RW	
2	Port pin Pi_2		Undefined	RW	
3	Port pin Pi_3		Undefined	RW	
4	Port pin Pi_4		Undefined	RW	
5	Port pin Pi_5		Undefined	RW	
6	Port pin Pi_6		Undefined	RW	
7	Port pin Pi_7		Undefined	RW	

Notes 1: Nothing is assigned for bits 0 and 4 of the port P5 register. These bits are undefined at reading.
2: Nothing is assigned for bits 4 to 7 of the port P8 register. These bits are undefined at reading.

Port Pi direction register (i = 1, 2, 4 to 8)
(Addresses 5_{16}, 8_{16}, C_{16}, D_{16}, 10_{16}, 11_{16}, 14_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Port Pi_0 direction bit	0 : Input mode (The port functions as an input port.) 1 : Output mode (The port functions as an output port.)	0	RW	5-3 Port P1 11-18 Port P2 7-9 8-7 Port P4 5-7 7-9 Port P5 6-17 8-7 9-11 9-22 10-13 Port P6 5-7 7-9 Port P7 12-15 Port P8 11-18 12-15
1	Port Pi_1 direction bit		0	RW	
2	Port Pi_2 direction bit		0	RW	
3	Port Pi_3 direction bit		0	RW	
4	Port Pi_4 direction bit		0	RW	
5	Port Pi_5 direction bit		0	RW	
6	Port Pi_6 direction bit		0	RW	
7	Port Pi_7 direction bit		0	RW	

Notes 1: Nothing is assigned for bits 0 and 4 of the port P5 direction register. These bits are undefined at reading.
2: Nothing is assigned for bits 4 to 7 of the port P8 direction register. These bits are undefined at reading.
3: Any of bits 0 to 7 of the port P4 direction register becomes "0" by input of a falling edge to pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$. (Refer to section **"5.2.3 Pin $\overline{P4OUT_{CUT}}/\overline{INT_0}$."**)
4: Any of bits 0 to 7 of the port P6 direction register becomes "0" by input of a falling edge to pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$. (Refer to section **"5.2.4 Pin $\overline{P6OUT_{CUT}}/\overline{INT_4}$."**)

A-D control register 0 (Address $1E_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Bit	Bit name	Function	At reset	R/W	Reference
0	Analog input pin select bits 0 (Valid in the one-shot and repeat modes.) **(Note 1)**	b2 b1 b0 0 0 0 : AN_0 is selected. 0 0 1 : AN_1 is selected. 0 1 0 : AN_2 is selected. 0 1 1 : AN_3 is selected. 1 0 0 : AN_4 is selected. 1 0 1 : AN_5 is selected. 1 1 0 : AN_6 is selected. 1 1 1 : AN_7 is selected. **(Note 2)**	Undefined	RW	12-8
1			Undefined	RW	
2			Undefined	RW	
3	A-D operation mode select bits 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or 1	0	RW	
4			0	RW	
5	Fix this bit to "0."		0	RW	
6	A-D conversion start bit	0 : A-D conversion halts. 1 : A-D conversion starts.	0	RW **(Note 3)**	
7	A-D conversion frequency (ϕ_{AD}) select bit 0	See Table 12.2.1.	0	RW	

Notes 1: When using pins AN_0 to AN_7, be sure to fix bit 3 of the analog input pin select bits 1 (bits 3 to 0 at address DB_{16}) to "0." Setting bit 3 of the analog input pin select bits 1 to "1" invalidates the analog input pin select bits 0. Also, the analog input pin select bits 0 are invalid in the single sweep mode, repeat sweep mode 0 and repeat sweet mode 1. (Each may be either "0" or "1.")

2: When using pin AN_7, be sure that the D-A_0 output enable bit (bit 0 at address 96_{16}) = "0" (output disabled).

3: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

4: Writing to each bit (except write of "0" to bit 6) of the A-D control register 0 must be performed while the A-D converter halts, regardless of the A-D operation mode.

A-D control register 1 (Address $1F_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Bit	Bit name	Function	At reset	R/W	Reference
0	A-D sweep pin select bits (Valid in the single sweep mode, repeat sweep mode 0, and repeat sweep mode 1.) **(Note 1)**	Single sweep mode/Repeat sweep mode 0 b1 b0 0 0 : Pins AN$_0$ and AN$_1$ (2 pins) 0 1 : Pins AN$_0$ to AN$_3$ (4 pins) 1 0 : Pins AN$_0$ to AN$_5$ (6 pins) 1 1 : Pins AN$_0$ to AN$_7$ (8 pins) **(Note 2)**	1	RW	12-8 12-9
1		Repeat sweep mode 1 **(Note 3)** b1 b0 0 0 : Pin AN$_0$ (1 pin) 0 1 : Pins AN$_0$ and AN$_1$ (2 pins) 1 0 : Pins AN$_0$ to AN$_2$ (3 pins) 1 1 : Pins AN$_0$ to AN$_3$ (4 pins)	1	RW	
2	A-D operation mode select bit 1 (Used in the repeat sweep mode 0 and repeat sweep mode 1.) **(Note 4)**	0 : Repeat sweep mode 0 1 : Repeat sweep mode 1	0	RW	
3	Resolution select bit	0 : 8-bit resolution mode 1 : 10-bit resolution mode	0	RW	
4	A-D conversion frequency (ϕ_{AD}) select bit 1	See Table 12.2.1.	0	RW	
5	Fix this bit to "0."		0	RW	
6	V$_{REF}$ connection select bit **(Note 5)**	0 : Pin V$_{REF}$ is connected. 1 : Pin V$_{REF}$ is disconnected.	0	RW	12-9 16-7
7	The value is "0" at reading.		0	–	

Notes 1: These bits are invalid in the one-shot and repeat modes. (They may be either "0" or "1.")
2: When using pin AN$_7$, be sure that the D-A$_0$ output enable bit (bit 0 at address 96_{16}) = "0" (output disabled).
3: Be sure to select the frequently-used analog input pins in the repeat sweep mode 1.
4: Fix this bit to "0" in the one-shot mode, repeat mode, and single sweep mode.
5: When this bit is cleared from "1" to "0," be sure to start the A-D conversion after an interval of 1 µs or more has elapsed.
6: Writing to each bit of the A-D control register 1 must be performed while the A-D converter halts, regardless of the A-D operation mode.

A-D control register 2 (Address DB_{16})

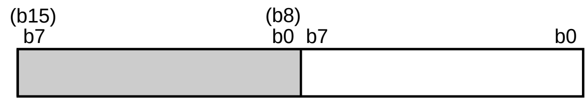
b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0				

Bit	Bit name	Function	At reset	R/W	Reference
0	Analog input pin select bits 1 **(Note 1)**	b3 b2 b1 b0 0 X X X : Pins AN$_0$ to AN$_7$ are selected. **(Note 2)** 1 0 0 0 : Pin AN$_8$ is selected. **(Note 3)** 1 0 0 1 : Pin AN$_9$ is selected. **(Note 4)** 1 0 1 0 : Pin AN$_{10}$ is selected. **(Note 5)** 1 0 1 1 : Pin AN$_{11}$ is selected. **(Note 6)** 1 1 0 0 : Do not select. ⋮ 1 1 1 1 : Do not select.	0	RW	12-8
1			0	RW	
2			0	RW	
3			0	RW	
7 to 4	Fix these bits to "0000."		0	RW	

X **:** They may be either "0" or "1."
Note 1: When using pins AN$_0$ to AN$_7$, regardless of the A-D operation mode, be sure to fix bit 3 to "0." Also, pins AN$_8$ to AN$_{11}$ are used only in the one-shot mode and repeat mode.
2: Use bits 2 to 0 of A-D control register 0 (address $1E_{16}$) for selection of pins AN$_0$ to AN$_7$.
3: When using pin AN$_8$, be sure that the D-A$_1$ output enable bit (bit 1 at address 96_{16}) = "0" (output disabled). Also, be sure not to use pin $\overline{CTS_2}/\overline{RTS_2}$.
4: When using pin AN$_9$, be sure not to use pin $\overline{CTS_2}$/CLK$_2$.
5: When using pin AN$_{10}$, be sure not to use pin R$_X$D$_2$.
6: When using pin AN$_{11}$, be sure not to use pin T$_X$D$_2$.
7: Writing to each bit of A-D control register 2 must be performed while the A-D conversion halts, regardless of the A-D operation mode.

■ When 8-bit resolution mode is selected

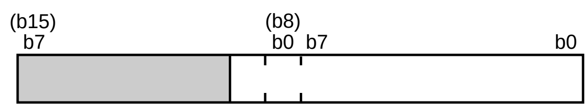
A-D register 0 (Addresses 21_{16}, 20_{16})
A-D register 1 (Addresses 23_{16}, 22_{16})
A-D register 2 (Addresses 25_{16}, 24_{16})
A-D register 3 (Addresses 27_{16}, 26_{16})
A-D register 4 (Addresses 29_{16}, 28_{16})
A-D register 5 (Addresses $2B_{16}$, $2A_{16}$)
A-D register 6 (Addresses $2D_{16}$, $2C_{16}$)
A-D register 7 (Addresses $2F_{16}$, $2E_{16}$)
A-D register 8 (Addresses $E1_{16}$, $E0_{16}$)
A-D register 9 (Addresses $E3_{16}$, $E2_{16}$)
A-D register 10 (Addresses $E5_{16}$, $E4_{16}$)
A-D register 11 (Addresses $E7_{16}$, $E6_{16}$)



Bit	Function	At reset	R/W	Reference
7 to 0	Reads an A-D conversion result.	Undefined	RO	12-10
15 to 8	The value is “0” at reading.	0	–	

■ When 10-bit resolution mode is selected

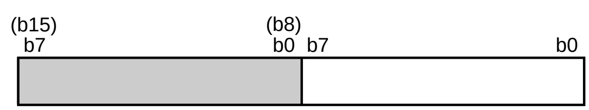
A-D register 0 (Addresses 21_{16}, 20_{16})
A-D register 1 (Addresses 23_{16}, 22_{16})
A-D register 2 (Addresses 25_{16}, 24_{16})
A-D register 3 (Addresses 27_{16}, 26_{16})
A-D register 4 (Addresses 29_{16}, 28_{16})
A-D register 5 (Addresses $2B_{16}$, $2A_{16}$)
A-D register 6 (Addresses $2D_{16}$, $2C_{16}$)
A-D register 7 (Addresses $2F_{16}$, $2E_{16}$)
A-D register 8 (Addresses $E1_{16}$, $E0_{16}$)
A-D register 9 (Addresses $E3_{16}$, $E2_{16}$)
A-D register 10 (Addresses $E5_{16}$, $E4_{16}$)
A-D register 11 (Addresses $E7_{16}$, $E6_{16}$)



Bit	Function	At reset	R/W	Reference
9 to 0	Reads an A-D conversion result.	Undefined	RO	12-10
15 to 10	The value is “0” at reading.	0	–	

■ When comparator function is selected

A-D register 0 (Addresses 21_{16}, 20_{16})
A-D register 1 (Addresses 23_{16}, 22_{16})
A-D register 2 (Addresses 25_{16}, 24_{16})
A-D register 3 (Addresses 27_{16}, 26_{16})
A-D register 4 (Addresses 29_{16}, 28_{16})
A-D register 5 (Addresses $2B_{16}$, $2A_{16}$)
A-D register 6 (Addresses $2D_{16}$, $2C_{16}$)
A-D register 7 (Addresses $2F_{16}$, $2E_{16}$)
A-D register 8 (Addresses $E1_{16}$, $E0_{16}$)
A-D register 9 (Addresses $E3_{16}$, $E2_{16}$)
A-D register 10 (Addresses $E5_{16}$, $E4_{16}$)
A-D register 11 (Addresses $E7_{16}$, $E6_{16}$)



Bit	Function	At reset	R/W	Reference
7 to 0	Any value in the range from “00_{16}” to “FF_{16}” can be set. The set value is compared with the input voltage. The value is undefined at reading.	Undefined	WO	12-10
15 to 8	The value is “0” at reading.	0	–	

Note: When the comparator function is selected, writing to and reading from the A-D register i must be performed while the A-D converter halts.

UART0 transmit/receive mode register (Address 30_{16})
UART1 transmit/receive mode register (Address 38_{16})
UART2 transmit/receive mode register (Address $B0_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Serial I/O mode select bits	b2 b1 b0 0 0 0 : Serial I/O is invalid. (P1 and P8 function as programmable I/O ports.) 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : } Do not select. 0 1 1 : } 1 0 0 : UART mode (Transfer data length = 7 bits) 1 0 1 : UART mode (Transfer data length = 8 bits) 1 1 0 : UART mode (Transfer data length = 9 bits) 1 1 1 : Do not select.	0	RW	11-5
1			0	RW	
2			0	RW	
3	Internal/External clock select bit	0 : Internal clock 1 : External clock	0	RW	
4	Stop bit length select bit (Valid in UART mode) **(Note)**	0 : One stop bit 1 : Two stop bits	0	RW	
5	Odd/Even parity select bit (Valid in UART mode when parity enable bit = "1.") **(Note)**	0 : Odd parity 1 : Even parity	0	RW	
6	Parity enable bit (Valid in UART mode) **(Note)**	0 : Parity disabled 1 : Parity enabled	0	RW	
7	Sleep select bit (Valid in UART mode) **(Note)**	0 : Sleep mode terminated (Invalid) 1 : Sleep mode selected	0	RW	

Note: Bits 4 to 6 are invalid in the clock synchronous serial I/O mode. (Each may be either "0" or "1.") Additionally, fix bit 7 to "0."

UART0 baud rate register (BRG0) (Address 31_{16})
UART1 baud rate register (BRG1) (Address 39_{16})
UART2 baud rate register (BRG2) (Address $B1_{16}$)

b7 b0

Bit	Function	At reset	R/W	Reference
7 to 0	Any value in the range from "00_{16}" to "FF_{16}" can be set. Assuming that the set value = n, BRGi divides the count source frequency by (n + 1).	Undefined	WO	11-14

Note: Writing to this register must be performed while the transmission/reception halts.
Use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction for writing to this register.

UART0 transmit buffer register (Addresses 33_{16}, 32_{16})
UART1 transmit buffer register (Addresses $3B_{16}$, $3A_{16}$)
UART2 transmit buffer register (Addresses $B3_{16}$, $B2_{16}$)

(b15) b7 (b8) b0 b7 b0

Bit	Function	At reset	R/W	Reference
8 to 0	Transmit data is set.	Undefined	WO	11-11
15 to 9	Nothing is assigned.	Undefined	–	

Note: Use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction for writing to this register.

UART0 transmit/receive control register 0 (Address 34_{16})
UART1 transmit/receive control register 0 (Address $3C_{16}$)
UART2 transmit/receive control register 0 (Address $B4_{16}$)

Bit	Bit name	Function	At reset	R/W	Reference
0	BRG count source select bits	b1 b0 0 0 : Clock f_2 0 1 : Clock f_{16} 1 0 : Clock f_{64} 1 1 : Clock f_{512}	0	RW	11-7
1			0	RW	
2	$\overline{CTS}/\overline{RTS}$ function select bit **(Note 1)**	0 : The $\overline{CTS}$ function is selected. 1 : The $\overline{RTS}$ function is selected.	0	RW	
3	Transmit register empty flag	0 : Data is present in the transmit register. (Transmission is in progress.) 1 : No data is present in the transmit register. (Transmission is completed.)	1	RO	
4	$\overline{CTS}/\overline{RTS}$ enable bit	0 : The $\overline{CTS}/\overline{RTS}$ function is enabled. 1 : The $\overline{CTS}/\overline{RTS}$ function is disabled.	0	RW	
5	UARTi receive interrupt mode select bit	0 : Reception interrupt 1 : Reception error interrupt	0	RW	
6	CLK polarity select bit (This bit is used in the clock synchronous serial I/O mode.) **(Note 2)**	0 : At the falling edge of the transfer clock, transmit data is output; at the rising edge of the transfer clock, receive data is input. When not in transferring, pin CLKi's level is "H." 1 : At the falling edge of the transfer clock, transmit data is output; at the falling edge of the transfer clock, receive data is input. When not in transferring, pin CLKi's level is "L."	0	RW	
7	Transfer format select bit (This bit is used in the clock synchronous serial I/O mode.) **(Note 2)**	0 : LSB (Least Significant Bit) first 1 : MSB (Most Significant Bit) first	0	RW	

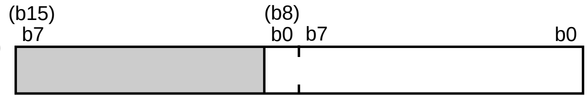
Notes 1: Valid when the $\overline{CTS}/\overline{RTS}$ enable bit (bit 4) is "0" and $\overline{CTS_i}/\overline{RTS_i}$ separate select bit (bit 0, 1 or 4 at address AC_{16}) is "0."
2: Fix these bits to "0" in the UART mode or when serial I/O is disabled.

UART0 transmit/receive control register 1 (Address 35_{16})
UART1 transmit/receive control register 1 (Address $3D_{16}$)
UART2 transmit/receive control register 1 (Address $B5_{16}$)

Bit	Bit name	Function	At reset	R/W	Reference
0	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0	RW	11-9
1	Transmit buffer empty flag	0 : Data is present in the transmit buffer register. 1 : No data is present in the transmit buffer register.	1	RO	
2	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0	RW	
3	Receive complete flag	0 : No data is present in the receive buffer register. 1 : Data is present in the receive buffer register.	0	RO	
4	Overrun error flag	0 : No overrun error 1 : Overrun error detected	0	RO	
5	Framing error flag **(Note)** (Valid in UART mode)	0 : No framing error 1 : Framing error detected	0	RO	
6	Parity error flag **(Note)** (Valid in UART mode)	0 : No parity error 1 : Parity error detected	0	RO	
7	Error sum flag **(Note)** (Valid in UART mode)	0 : No error 1 : Error detected	0	RO	

Note: Bits 5 to 7 are invalid in the clock synchronous serial I/O mode.

UART0 receive buffer register (Addresses 37_{16}, 36_{16})
UART1 receive buffer register (Addresses $3F_{16}$, $3E_{16}$)
UART2 receive buffer register (Addresses $B7_{16}$, $B6_{16}$)



Bit	Function	At reset	R/W	Reference
8 to 0	Receive data is read out from here.	Undefined	RO	11-13
15 to 9	The value is “0” at reading.	0	–	

Count start register 0 (Address 40_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Timer A0 count start bit	0 : Stop counting 1 : Start counting	0	RW	7-6
1	Timer A1 count start bit		0	RW	
2	Timer A2 count start bit		0	RW	
3	Timer A3 count start bit		0	RW	
4	Timer A4 count start bit		0	RW	
5	Timer B0 count start bit		0	RW	8-4
6	Timer B1 count start bit		0	RW	
7	Timer B2 count start bit		0	RW	

Count start register 1 (Address 41_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Timer A5 count start bit	0 : Stop counting 1 : Start counting	0	RW	7-6
1	Timer A6 count start bit		0	RW	
2	Timer A7 count start bit		0	RW	
3	Timer A8 count start bit		0	RW	
4	Timer A9 count start bit		0	RW	
7 to 5	Nothing is assigned.		Undefined	–	

One-shot start register 0 (Address 42_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0							

Bit	Bit name	Function	At reset	R/W	Reference
0	Timer A0 one-shot start bit	1 : Start outputting one-shot pulse. (Valid when an internal trigger is selected.) The value is “0” at reading.	0	WO	7-33
1	Timer A1 one-shot start bit		0	WO	
2	Timer A2 one-shot start bit		0	WO	
3	Timer A3 one-shot start bit		0	WO	
4	Timer A4 one-shot start bit		0	WO	
6, 5	Nothing is assigned.		Undefined	–	
7	Fix this bit to “0.”		0	RW	

One-shot start register 1 (Address 43_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0							

Bit	Bit name	Function	At reset	R/W	Reference
0	Timer A5 one-shot start bit	1 : Start outputting one-shot pulse. (Valid when an internal trigger is selected.) The value is “0” at reading.	0	WO	7-33
1	Timer A6 one-shot start bit		0	WO	
2	Timer A7 one-shot start bit		0	WO	
3	Timer A8 one-shot start bit		0	WO	
4	Timer A9 one-shot start bit		0	WO	
6, 5	Nothing is assigned.		Undefined	–	
7	Fix this bit to “0.”		0	RW	

Up-down register 0 (Address 44_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Timer A0 up-down bit	0 : Countdown 1 : Countup This function is valid when the contents of the up-down register is selected as the up-down switching factor.	0	RW	7-24
1	Timer A1 up-down bit		0	RW	
2	Timer A2 up-down bit		0	RW	
3	Timer A3 up-down bit		0	RW	
4	Timer A4 up-down bit		0	RW	
5	Timer A2 two-phase pulse signal processing select bit	0 : Two-phase pulse signal processing function disabled 1 : Two-phase pulse signal processing function enabled When not using the two-phase pulse signal processing function, clear the bit to "0." The value is "0" at reading.	0	WO **(Note)**	7-26
6	Timer A3 two-phase pulse signal processing select bit		0	WO **(Note)**	
7	Timer A4 two-phase pulse signal processing select bit		0	WO **(Note)**	

Note: Use the **MOVM (MOVMB)** or **STA(STAB, STAD)** instruction for writing to bits 5 to 7.

Up-down register 1 (Address $C4_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Timer A5 up-down bit	0 : Countdown 1 : Countup This function is valid when the contents of the up-down register is selected as the up-down switching factor.	0	RW	7-24
1	Timer A6 up-down bit		0	RW	
2	Timer A7 up-down bit		0	RW	
3	Timer A8 up-down bit		0	RW	
4	Timer A9 up-down bit		0	RW	
5	Timer A7 two-phase pulse signal processing select bit	0 : Two-phase pulse signal processing function disabled 1 : Two-phase pulse signal processing function enabled When not using the two-phase pulse signal processing function, clear the bit to "0." The value is "0" at reading.	0	WO **(Note)**	7-26
6	Timer A8 two-phase pulse signal processing select bit		0	WO **(Note)**	
7	Timer A9 two-phase pulse signal processing select bit		0	WO **(Note)**	

Note: Use the **MOVM (MOVMB)** or **STA(STAB, STAD)** instruction for writing to bits 5 to 7.

Timer A clock division select register (Address 45_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Timer A clock division select bits	See Table 7.2.3.	0	RW	7-5
1			0	RW	
7 to 2	The value is “0” at reading.		0	–	

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 (b8) b0 b7 b0

Bit	Function	At reset	R/W	Reference
15 to 0	These bits have different functions according to the operating mode.	Undefined	RW	7-4

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot pulse mode 1 1 : Pulse width modulation (PWM) mode.	0	RW	7-7
1			0	RW	
2	These bits have different functions according to the operating mode.		0	RW	
3			0	RW	
4			0	RW	
5			0	RW	
6			0	RW	
7			0	RW	

■ Timer mode

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 | (b8) b0 | b7 | b0

Bit	Function	At reset	R/W	Reference
15 to 0	Any value in the range from “0000_{16}” to “$FFFF_{16}$” can be set. Assuming that the set value = n, the counter divides the count source frequency by (n + 1). When reading, the register indicates the counter value.	Undefined	RW	7-12

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 0 0 : Timer mode	0	RW	7-12 9-12 9-23 10-15
1			0	RW	
2	Pulse output function select bit	0 : No pulse output (TAi$_{OUT}$ pin functions as a programmable I/O port pin.) 1 : Pulse output (TAi$_{OUT}$ pin functions as a pulse output pin.)	0	RW	7-16
3	Gate function select bits	b4 b3 0 0 : } No gate function 0 1 : } (TAi$_{IN}$ pin functions as a programmable I/O port pin.) 1 0 : Gate function (Counter is active only while TAi$_{IN}$ pin's input signal is at “L” level.) 1 1 : Gate function (Counter is active only while TAi$_{IN}$ pin's input signal is at “H” level.)	0	RW	7-15
4			0	RW	
5	Fix this bit to “0” in timer mode.		0	RW	
6	Count source select bits	See Table 7.2.3.	0	RW	7-5
7			0	RW	

■ Event counter mode

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7	(b8) b0	b7	b0

Bit	Function	At reset	R/W	Reference
15 to 0	Any value in the range from "0000_{16}" to "$FFFF_{16}$" can be set. Assuming that the set value = n, the counter divides the count source frequency by (n + 1) during countdown, or by ($FFFF_{16}$ – n + 1) during countup. When reading, the register indicates the counter value.	Undefined	RW	7-20

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0
X	X	0				0	1

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 0 1 : Event counter mode	0	RW	7-20
1			0	RW	
2	Pulse output function select bit	0 : No pulse output (TAiOUT pin functions as a programmable I/O port pin.) 1 : Pulse output (TAiOUT pin functions as a pulse output pin.)	0	RW	7-26
3	Count polarity select bit	0 : Counts at falling edge of external signal 1 : Counts at rising edge of external signal	0	RW	7-20
4	Up-down switching factor select bit	0 : Contents of up-down register 1 : Input signal to TAiOUT pin	0	RW	7-24
5	Fix this bit to "0" in event counter mode.		0	RW	
6	These bits are invalid in event counter mode.		0	RW	
7			0	RW	

X : It may be either "0" or "1."

■ One-shot pulse mode

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 (b8) b0 b7 b0

Bit	Function	At reset	R/W	Reference
15 to 0	Any value in the range from "0000_{16}" to "$FFFF_{16}$" can be set. Assuming that the set value = n, the "H" level width of the one-shot pulse which is output from the TAi_{OUT} pin is expressed as follows : $\frac{n}{f_i}$.	Undefined	WO	7-30 10-13

f_i: Frequency of count source
Note: Use the **MOVM** or **STA(STAD)** instruction for writing to this register.
Writing to this register must be performed in a unit of 16 bits.

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0
		0			1	1	0

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 1 0 : One-shot pulse mode	0	RW	7-30 10-13
1			0	RW	
2	Fix this bit to "1" in one-shot pulse mode.		0	RW	
3	Trigger select bits	b4 b3 0 0 : } Writing "1" to one-shot start bit 0 1 : } (TAi_{IN} pin functions as a programmable I/O port pin.) 1 0 : Falling edge of TAi_{IN} pin's input signal 1 1 : Rising edge of TAi_{IN} pin's input signal	0	RW	7-33
4			0	RW	
5	Fix this bit to "0" in one-shot pulse mode.		0	RW	
6	Count source select bits	See Table 7.2.3.	0	RW	7-5
7			0	RW	

■ Pulse width modulation (PWM) mode

<When operating as a 16-bit pulse width modulator>

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 | (b8) b0 | b7 | b0

Bit	Function	At reset	R/W	Reference
15 to 0	Any value in the range from "0000_{16}" to "$FFFE_{16}$" can be set. Assuming that the set value = n, the "H" level width of the PWM pulse which is output from the TAi_{OUT} pin is expressed as follows : $\frac{n}{f_i}$ (PWM pulse period = $\frac{2^{16}-1}{f_i}$)	Undefined	WO	7-39

f_i: Frequency of count source
Note: Use the **MOVM** or **STA(STAD)** instruction for writing to this register.
Writing to this register must be performed in a unit of 16 bits.

<When operating as an 8-bit pulse width modulator>

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)
Timer A5 register (Addresses $C7_{16}$, $C6_{16}$)
Timer A6 register (Addresses $C9_{16}$, $C8_{16}$)
Timer A7 register (Addresses CB_{16}, CA_{16})
Timer A8 register (Addresses CD_{16}, CC_{16})
Timer A9 register (Addresses CF_{16}, CE_{16})

(b15) b7 | (b8) b0 | b7 | b0

Bit	Function	At reset	R/W	Reference
7 to 0	Any value in the range from "00_{16}" to "FF_{16}" can be set. Assuming that the set value = m, the period of the PWM pulse which is output from the TAi_{OUT} pin is expressed as follows: $\frac{(m+1)(2^8-1)}{f_i}$	Undefined	WO	7-39
15 to 8	Any value in the range from "00_{16}" to "FF_{16}" can be set. Assuming that the set value = n, the "H" level width of the PWM pulse which is output from the TAi_{OUT} pin is expressed as follows: $\frac{n(m+1)}{f_i}$	Undefined	WO	

f_i: Frequency of count source
Note: Use the **MOVM** or **STA(STAD)** instruction for writing to this register.
Writing to this register must be performed in a unit of 16 bits.

■ Pulse width modulation (PWM) mode

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)
Timer Ai mode register (i = 5 to 9) (Addresses $D6_{16}$ to DA_{16})

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 1 1 : PWM mode	0	RW	7-40 9-12 9-23
1			0	RW	
2	Fix this bit to “1” in PWM mode.		0	RW	
3	Trigger select bits	b4 b3 0 0 : } Writing “1” to count start bit 0 1 : } (TAi$_{IN}$ pin functions as a programmable I/O port pin.) 1 0 : Falling edge of TAi$_{IN}$ pin’s input signal 1 1 : Rising edge of TAi$_{IN}$ pin’s input signal	0	RW	7-43
4			0	RW	
5	16/8-bit PWM mode select bit	0 : 16-bit pulse width modulator 1 : 8-bit pulse width modulator	0	RW	7-44
6	Count source select bits	See Table 7.2.3.	0	RW	7-5
7			0	RW	

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

(b15) b7 (b8) b0 b7 b0

Bit	Function	At reset	R/W	Reference
15 to 0	These bits have different functions according to the operating mode.	Undefined	RW	8-3

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/Pulse width measurement mode 1 1 : Do not select.	0	RW	8-4
1			0	RW	
2	These bits have different functions according to the operating mode.		0	RW	
3			0	RW	
4			0	RW	
5			Undefined	RO **(Note)**	
6			0	RW	
7			0	RW	

Note: Bit 5 is invalid in the timer and event counter modes; its value is undefined at reading.

■ Timer mode

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

(b15) b7 ... (b8) b0 | b7 ... b0

Bit	Function	At reset	R/W	Reference
15 to 0	Any value in the range from "0000_{16}" to "$FFFF_{16}$" can be set. Assuming that the set value = n, the counter divides the count source frequency by (n + 1). When reading, the register indicates the counter value.	Undefined	RW	8-9

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		X	X	X	X	0	0

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 0 0 : Timer mode	0	RW	8-9
1			0	RW	
2	These bits are invalid in timer mode.		0	RW	
3			0	RW	
4			0	RW	
5	This bit is invalid in timer mode; its value is undefined at reading.		Undefined	RO	
6	Count source select bits	b7 b6 0 0 : f_2 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW	8-7
7			0	RW	

X : It may be either "0" or "1."

■ Event counter mode

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

(b15) b7 … (b8) b0 b7 … b0

Bit	Function	At reset	R/W	Reference
15 to 0	Any value in the range from “0000_{16}” to “$FFFF_{16}$” can be set. Assuming that the set value = n, the counter divides the count source frequency by (n + 1). When reading, the register indicates the counter value.	Undefined	RW	8-14

Note: Reading from or writing to this register must be performed in a unit of 16 bits.

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X			0	1

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 0 1 : Event counter mode	0	RW	8-14
1			0	RW	
2	Count polarity select bits	b3 b2 0 0 : Count at falling edge of external signal 0 1 : Count at rising edge of external signal 1 0 : Count at both falling and rising edges of external signal 1 1 : Do not select. **(Note)**	0	RW	
3			0	RW	
4	This bit is invalid in event counter mode.		0	RW	
5	This bit is invalid in event counter mode; its value is undefined at reading.		Undefined	RO	
6	These bits are invalid in event counter mode.		0	RW	
7			0	RW	

X : It may be either “0” or “1.”

Note: When the timer B2 clock source select bit (bit 6 at address 63_{16}) = “1,” be sure to fix these bits to “01_2” (count at the rising edge of the external signal).

■ Pulse period/Pulse width measurement mode

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

(b15) b7 (b8) b0 b7 b0

Bit	Function	At reset	R/W	Reference
15 to 0	The measurement result of pulse period or pulse width is read out.	Undefined	RO	8-21

Note: Reading from this register must be performed in a unit of 16 bits.

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Bit	Bit name	Function	At reset	R/W	Reference
0	Operating mode select bits	b1 b0 1 0 : Pulse period/Pulse width measurement mode	0	RW	8-21
1			0	RW	
2	Measurement mode select bits	b3 b2 0 0 : Pulse period measurement (Interval between falling edges of measurement pulse) 0 1 : Pulse period measurement (Interval between rising edges of measurement pulse) 1 0 : Pulse width measurement (Interval from a falling edge to a rising edge, and from a rising edge to a falling edge of measurement pulse) 1 1 : Do not select.	0	RW	8-23
3			0	RW	
4	Count-type select bit	0 : Counter clear type 1 : Free-run type	0	RW	
5	Timer Bi overflow flag **(Note)**	0 : No overflow 1 : Overflowed	Undefined	RO	8-24
6	Count source select bits	b7 b6 0 0 : f_2 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW	8-7
7			0	RW	

Note: The timer Bi overflow flag is cleared to "0" when a value is written to the timer Bi mode register with the count start bit = "1."
This flag cannot be set to "1" by software.

Processor mode register 0 (Address $5E_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
0				X	X	0	0

Bit	Bit name	Function	At reset	R/W	Reference
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Do not select. 1 0 : Do not select. 1 1 : Do not select.	0	RW	2-20
1			0	RW	
2	Any of these bits may be either “0” or “1.”		0	RW	
3			1	RW	
4	Interrupt priority detection time select bits	b5 b4 0 0 : 7 cycles of f_{sys} 0 1 : 4 cycles of f_{sys} 1 0 : 2 cycles of f_{sys} 1 1 : Do not select.	0	RW	6-11
5			0	RW	
6	Software reset bit	The microcomputer is reset by writing “1” to this bit. The value is “0” at reading.	0	WO	3-3
7	Fix this bit to “0.”		0	RW	

X : It may be either “0” or “1.”

Processor mode register 1 (Address $5F_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	0		X

Bit	Bit name	Function	At reset	R/W	Reference
0	This bit may be either “0” or “1.”		1	RW	
1	Direct page register switch bit	0 : Only DPR0 is used. 1 : DPR0 through DPR3 are used.	0	RW **(Note 1)**	2-6
6 to 2	Fix these bits to “00000.”		0	RW	
7	Internal ROM bus cycle select bit **(Note 2)**	0 : 3ϕ 1 : 2ϕ	0	RW	2-12

X : It may be either “0” or “1.”

Notes 1: After reset, this bit is allowed to be changed only once. (During the software execution, be sure not to change this bit’s content.)

2: To reprogram the internal flash memory by using the CPU reprogramming mode, clear this bit to “0.” (Refer to section **“19.2 Flash memory CPU reprogramming mode.”**)

Watchdog timer register (Address 60_{16})

b7 b0

Bit	Function	At reset	R/W	Reference
7 to 0	Initializes the watchdog timer. When dummy data has been written to this register, the watchdog timer's value is initialized to "FFF_{16}" (dummy data: 00_{16} to FF_{16}).	Undefined	—	14-3

Watchdog timer frequency select register (Address 61_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Watchdog timer frequency select bit	0 : Wf_{512} 1 : Wf_{32}	0	RW	14-3
5 to 1	Nothing is assigned.		Undefined	—	
6	Watchdog timer clock source select bits at STP termination	b7 b6 0 0 : fX_{32} 0 1 : fX_{16} 1 0 : fX_{128} 1 1 : fX_{64}	0	RW	14-3 15-7
7			0	RW	

Particular function select register 0 (Address 62_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0		

Bit	Bit name	Function	At reset	R/W	Reference
0	STP instruction invalidity select bit	0 : STP instruction is valid. 1 : STP instruction is invalid.	0	RW **(Note)**	15-4
1	External clcok input select bit	0 : Oscillation circuit is active. (Oscillator is connected.) Watchdog timer is used at stop mode termination. 1 : Oscillation circuit is inactive. (External clock is input.) When the system clock select bit (bit 5 at address BC_{16}) = "0," watchdog timer is not used at stop mode termination. When the system clock select bit = "1," watchdog timer is used at stop mode termination.	0	RW **(Note)**	4-10 15-5 16-4
7 to 2	Fix these bits to "000000."		0	RW	

Note: Writing to these bits requires the following procedure:
- Write "55_{16}" to this register. (The bit status does not change only by this writing.)
- Succeedingly, write "0" or "1" to each bit.

Also, use the **MOVMB** (**MOVM** when m = 1) instruction or **STAB** (**STA** when m = 1) instruction.
If an interrupt occurs between writing of "55_{16}" and next writing of "0" or "1," latter writing may be ignored. When there is a possibility that an interrupt occurs at the above timing, be sure to read this bit's contents after writing of "0" or "1," and verify whether "0" or "1" has correctly been written or not.

Particular function select register 1 (Address 63_{16})

b7	b6	b5	b4	b3	b2	b1	b0
			0		0		

Bit	Bit name	Function	At reset	R/W	Reference
0	STP-instruction-execution status bit	0 : Normal operation. 1 : During execution of STP instruction	**(Note 1)**	RW **(Note 2)**	15-6
1	WIT-instruction-execution status bit	0 : Normal operation. 1 : During execution of WIT instruction	**(Note 1)**	RW **(Note 2)**	
2	Fix this bit to "0."		0	RW	
3	System clock stop select bit at WIT **(Note 3)**	0 : In the wait mode, system clock f_{sys} is active. 1 : In the wait mode, system clock f_{sys} is inactive.	0	RW	16-5
4	Fix this bit to "0."		0	RW	
5	The value is "0" at reading.		0	—	
6	Timer B2 clock source select bit (Valid in event counter mode.) **(Note 4)**	0 : External signal input to the $TB2_{IN}$ pin is counted. 1 : fX_{32} is counted.	0	RW	8-15
7	The value is "0" at reading.		0	—	

Notes 1: At power-on reset, this bit becomes "0." At hardware reset or software reset, this bit retains the value just before reset.
2: Even when "1" is written, the bit status will not change.
3: Setting this bit to "1" must be performed just before execution of the WIT instruction. Also, after the wait state is terminated, this bit must be cleared to "0" immediately.
4: When using timer B2 in the pulse period/pulse width measurement mode, be sure to clear this bit to "0."

Particular function select register 2 (Address 64_{16})

b7 b0

Bit	Function	At reset	R/W	Reference
7 to 0	Disables the watchdog timer. When values of "79_{16}" and "50_{16}" succeedingly in this order, the watchdog timer will stop its operation.	Undefined	—	14-4

Note: After reset, this register can be set only once. Writing to this register requires the following procedure:

- Write values of "79_{16}" and "50_{16}" to this register succeedingly in this order.
- For the above writing, be sure to use the **MOVMB** (**MOVM** when m = 1) instruction or the **STAB** (**STA** when m = 1).

Note that the following: if an interrupt occurs between writing of "79_{16}" and next writing of "50_{16}," the watchdog timer does not stop its operation.

If any of the following has been performed after reset, writing to this register will be disabled from that time:

- If this register is read out.
- If writing to this register is performed by the procedure other than the above procedure.

Debug control register 0 (Address 66_{16})

b7	b6	b5	b4	b3	b2	b1	b0
	0		0	0			

Bit	Bit name	Function	At reset	R/W	Reference
0	Detect condition select bits **(Note 1)**	b2 b1 b0 0 0 0 : Do not select. 0 0 1 : Address matching detection 0 0 1 0 : Address matching detection 1 0 1 1 : Address matching detection 2 1 0 0 : Do not select. 1 0 1 : Out-of-address-area detection 1 1 0 : } Do not select. 1 1 1 :	**(Note 2)**	RW	17-3
1			**(Note 2)**	RW	
2			**(Note 2)**	RW	
3	Fix these bits to “00.”		**(Note 2)**	RW	
4			**(Note 2)**	RW	
5	Detect enable bit	0 : Detection disabled. 1 : Detection enabled.	**(Note 2)**	RW	
6	Fix this bit to “0.”		**(Note 2)**	RW	
7	The value is “1” at reading.		1	—	

Notes 1: These bits are valid when the detect enable bit (bit 5) = “1.” Therefore, these bits must be set before or simultaneously with setting of the detect enable bit to “1.”

2: At power-on reset, each bit becomes “0”; at hardware reset or software reset, each bit retains the value immediately before reset.

Debug control register 1 (Address 67_{16})

b7	b6	b5	b4	b3	b2	b1	b0
				1			0

Bit	Bit name	Function	At reset	R/W	Reference
0	Fix this bit to “0.”		**(Note 1)**	RW	17-4
1	The value is “0” at reading.		**(Note 1)**	RO	
2	Address compare register access enable bit **(Note 2)**	0 : Disabled. 1 : Enabled.	0	RW	
3	Fix this bit to “1” when using the debug function.		0	RW	
4	Nothing is assigned.		Undefined	—	
5	While a debugger is not used, the value is “0” at reading. While a debugger is used, the value is “1” at reading.		0	RO	
6	Address-matching-detection 2 decision bit (Valid when the address matching detection 2 is selected.)	0 : Matches with the contents of the address compare register 0. 1 : Matches with the contents of the address compare register 1.	0	RO	
7	The value is “0” at reading.		0	—	

Notes 1: At power-on reset, each bit becomes “0”; at hardware reset or software reset, each bit retains the value immediately before reset.

2: Be sure to set this bit to “1” immediately before the access to the address compare registers 0 and 1 (addresses 68_{16} to $6D_{16}$). Then, be sure to clear this bit to “0” immediately after this access.

Address compare register 0 (Addresses $6A_{16}$ to 68_{16})
Address compare register 1 (Addresses $6D_{16}$ to $6B_{16}$)

(b23) b7	(b16) b0	(b15) b7	(b8) b0	b7	b0

Bit	Function	At reset	R/W	Reference
23 to 0	The address to be detected (in other words, the start address of instructions) is set here.	Undefined	RW	17-5

Note: When accessing these registers, be sure to set the address compare register access enable bit (bit 2 at address 67_{16}) to “1” immediately before this access. Then, be sure to clear this bit to “0” immediately after this access.

$\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_2}$ interrupt control registers (Addresses $7D_{16}$, $7E_{16}$, $7F_{16}$)
$\overline{INT_3}$, $\overline{INT_4}$ interrupt control registers (Addresses $6E_{16}$, $6F_{16}$)
$\overline{INT_5}$, $\overline{INT_6}$, $\overline{INT_7}$ interrupt control registers (Addresses FD_{16}, FE_{16}, FF_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	RW	6-7
1			0	RW	
2			0	RW	
3	Interrupt request bit **(Note 1)**	0 : No interrupt requested 1 : Interrupt requested	0	RW **(Note 2)**	
4	Polarity select bit	0 : The interrupt request bit is set to "1" at "H" level when level sense is selected; this bit is set to "1" at falling edge when edge sense is selected. 1 : The interrupt request bit is set to "1" at "L" level when level sense is selected; this bit is set to "1" at rising edge when edge sense is selected.	0	RW	6-17
5	Level sense/Edge sense select bit	0 : Edge sense 1 : Level sense	0	RW	
7, 6	Nothing is assigned.		Undefined	—	

Notes 1: The interrupt request bits of $\overline{INT_0}$ to $\overline{INT_7}$ interrupts are invalid when the level sense is selected.
2: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

A-D conversion, UART0 and 1 transmit, UART0 and 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers (Addresses 70_{16} to $7C_{16}$)
UART2 transmit, UART2 receive interrupt control registers (Addresses $F1_{16}$, $F2_{16}$)
Timers A5 to A9 interrupt control registers (Addresses $F5_{16}$ to $F9_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	RW	6-7 Timer Ai 7-8 Timer Bi 8-5 UART0 UART1 UART2 11-16 A-D 12-14
1			0	RW	
2			0	RW	
3	Interrupt request bit	0 : No interrupt requested 1 : Interrupt requested	0 **(Note 1)**	RW **(Note 2)**	
7 to 4	Nothing is assigned.		Undefined	—	

Notes 1: The A-D conversion interrupt request bit is undefined after reset.
2: When writing to this bit, use the **MOVM (MOVMB)** or **STA (STAB, STAD)** instruction.

External interrupt input read register (Address 95_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	$\overline{INT_0}$ read out bit	The input level at the corresponding pin is read out. 0 : "L" level 1 : "H" level	Undefined	RO
1	$\overline{INT_1}$ read out bit		Undefined	RO
2	$\overline{INT_2}$ read out bit		Undefined	RO
3	$\overline{INT_3}$ read out bit		Undefined	RO
4	$\overline{INT_4}$ read out bit		Undefined	RO
5	$\overline{INT_5}$ read out bit		Undefined	RO
6	$\overline{INT_6}$ read out bit		Undefined	RO
7	$\overline{INT_7}$ read out bit		Undefined	RO

Reference 6-17

D-A control register (Address 96_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	D-A_0 output enable bit	0: Output is disabled. 1: Output is enabled. **(Notes 1, 2)**	0	RW	13-3
1	D-A_1 output enable bit	0: Output is disabled. 1: Output is enabled. **(Notes 1, 2)**	0	RW	
7 to 2	Nothing is assigned.		Undefined	—	

Notes 1: Pin DA_i is multiplexed with an analog input pin or serial input/output pin. When a D-A_i output enable bit = "1" (in other words, output is enabled.), however, the corresponding pin cannot function as any other multiplexed input/output pin (including a programmable I/O port pin).
2: When not using the D-A converter, be sure to clear this bit to "0."

D-A register i (i = 0 and 1) (Addresses 98_{16} and 99_{16})

b7 … b0

Bit	Function	At reset	R/W	Reference
7 to 0	Any value in the range from 00_{16} through FF_{16} can be set **(Note)**, and this value will be D-A converted and will be output.	0	RW	13-3

Note: When not using the D-A converter, be sure to clear the contents of these bits to "00_{16}."

Flash memory control register (Address $9E_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	RY/$\overline{BY}$ status bit	0 : BUSY (Automatic programming or erase operation is active.) 1 : READY (Automatic programming or erase operation has been completed.)	1	RO	19-10 19-11
1	CPU reprogramming mode select bit	0 : Flash memory CPU reprogramming mode is invalid. 1 : Flash memory CPU reprogramming mode is valid.	0	RW **(Notes 1, 2)**	
2	The value is "0" at reading.		0	—	
3	Flash memory reset bit **(Note 3)**	Writing "1" into this bit discontinues the access to the internal flash memory. This causes the built-in flash memory circuit being reset.	0	RW **(Note 4)**	
4	The value is "0" at reading.		0	—	
5	User ROM area select bit (Valid in boot mode) **(Note 5)**	0 : Access to boot ROM area 1 : Access to user ROM area	0	RW **(Note 2)**	
7, 6	The value is "0" at reading.		0	—	

Notes 1: In order to set this bit to "1," write "0" followed with "1" successively; while in order to clear this bit "0," write "0."
2: Writing to this bit must be performed in an area other than the internal flash memory.
3: This bit is valid when the CPU reprogramming mode select bit (bit 1) = "1": on the other hand, when the CPU reprogramming mode select bit = "0," be sure to fix this bit to "0." Rewriting of this bit must be performed with the CPU reprogramming mode select bit = "1."
4: After writing of "1" to this bit, be sure to confirm the RY/$\overline{BY}$ status bit (bit 0) becomes "1"; and then, write "0" to this bit.
5: When MD1 = Vss level, this bit is invalid. (It may be either "0" or "1.")

Pulse output control register (Address $A0_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W
0	Waveform output select bits **(Note)**	See Table 9.3.1.	0	RW
1			0	RW
2			0	RW
3	Pulse output mode select bit	0 : Pulse mode 0 1 : Pulse mode 1	0	RW
4	Pulse width modulation timer select bits	See Table 9.3.2.	0	RW
5			0	RW
6	Waveform output control bit 0	When pulse mode 0 is selected, 0: $RTP3_0$ to $RTP3_3$: pulse outputs are disabled. 1: $RTP3_0$ to $RTP3_3$: pulse outputs are enabled. When pulse mode 1 is selected, 0: $RTP3_2$, $RTP3_3$: pulse outputs are disabled. 1: $RTP3_2$, $RTP3_3$: pulse outputs are enabled.	0	RW
7	Waveform output control bit 1	When pulse mode 0 is selected, 0 : $RTP2_0$ to $RTP2_3$: pulse outputs are disabled. 1 : $RTP2_0$ to $RTP2_3$: pulse outputs are enabled. When pulse mode 1 is selected, 0 : $RTP2_0$ to $RTP2_3$, $RTP3_0$, $RTP3_1$: pulse outputs are disabled. 1 : $RTP2_0$ to $RTP2_3$, $RTP3_0$, $RTP3_1$: pulse outputs are enabled.	0	RW

Reference: 9-17, 9-18

Note: When not using pulse output port 1, be sure to fix these bits to "000_2."

Pulse output data register 0 (Address $A2_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	$RTP2_0$ pulse output data bit	0 : "L" level output 1 : "H" level output	0	RW	9-20
1	$RTP2_1$ pulse output data bit		0	RW	
2	$RTP2_2$ pulse output data bit		0	RW	
3	$RTP2_3$ pulse output data bit		0	RW	
4	$RTP3_0$ pulse output data bit (Valid in pulse mode 1) **(Note)**		0	RW	
5	$RTP3_1$ pulse output data bit (Valid in pulse mode 1) **(Note)**		0	RW	
7, 6	Pulse output trigger select bits	b7 b6 0 0 : Underflow of timer A5 0 1 : Falling edge of input signal to pin RTP_{TRG1} 1 0 : Rising edge of input signal to pin RTP_{TRG1} 1 1 : Both falling and rising edges of input signal to pin RTP_{TRG1}	0	RW	

Note: This bit is invalid in pulse mode 0.

Pulse output data register 1 (Address $A4_{16}$)

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Pulse width modulation enable bit 0	0 : No pulse width modulation by timer A6 1 : Pulse width modulation by timer A6	0	RW	9-20
1	Pulse width modulation enable bit 1	0 : No pulse width modulation by timer A7 1 : Pulse width modulation by timer A7	0	RW	
2	Pulse width modulation enable bit 2	0 : No pulse width modulation by timer A9 1 : Pulse width modulation by timer A9	0	RW	
3	Pulse output polarity select bit	0 : Positive 1 : Negative	0	RW	
4	$RTP3_0$ pulse output data bit (Valid in pulse mode 0) **(Note)**	0 : "L" level output 1 : "H" level output	0	RW	
5	$RTP3_1$ pulse output data bit (Valid in pulse mode 0) **(Note)**		0	RW	
6	$RTP3_2$ pulse output data bit		0	RW	
7	$RTP3_3$ pulse output data bit		0	RW	

Note: This bit is invalid in pulse mode 1.

Waveform output mode register (Address $A6_{16}$)

■ Three-phase waveform mode

Waveform output mode register (Address $A6_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		X			1	0	0

Bit	Bit name	Function	At reset	R/W	Reference
0	Waveform output select bits **(Note 1)**	b2 b1 b0 1 0 0 : Three-phase waveform mode	0	RW	10-6
1			0	RW	
2			0	RW	
3	Three-phase output polarity set buffer (Valid in three-phase mode 1) **(Note 2)**	0 : "H" output 1 : "L" output	0	RW	
4	Three-phase mode select bit	0 : Three-phase mode 0 1 : Three-phase mode 1	0	RW	
5	Invalid in the three-phase waveform mode.		0	RW	
6	Dead-time timer trigger select bit **(Note 3)**	0: Both falling and rising edges of one-shot pulse for timers A0 to A2 1: Only the falling edge of one-shot pulse for timers A0 to A2	0	RW	
7	Waveform output control bit	0 : Waveform output disabled 1 : Waveform output enabled	0	RW	

X: It may be either "0" or "1."

Notes 1: When not using pulse output port 0 and three-phase waveform mode, be sure to fix these bits to "000_2."
2: This bit is invalid in three-phase mode 0.
3: When the saw-tooth-wave modulation output is performed, be sure to fix this bit to "0."
4: Writing to any of bits 0 to 6 must be performed while counting for timers A0 to A3 halts.

■ Pulse output mode (Pulse output port 0)

Waveform output mode register (Address $A6_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Waveform output select bits **(Note)**	See Table 9.2.1.	0	RW	9-6 9-7
1			0	RW	
2			0	RW	
3	Pulse output mode select bit	0 : Pulse mode 0 1 : Pulse mode 1	0	RW	
4	Pulse width modulation timer select bits	See Table 9.2.2.	0	RW	
5			0	RW	
6	Waveform output control bit 0	When pulse mode 0 is selected, 0: $RTP1_0$ to $RTP1_3$: pulse outputs are disabled. 1: $RTP1_0$ to $RTP1_3$: pulse outputs are enabled. When pulse mode 1 is selected, 0: $RTP1_2$, $RTP1_3$: pulse outputs are disabled. 1: $RTP1_2$, $RTP1_3$: pulse outputs are enabled.	0	RW	
7	Waveform output control bit 1	When pulse mode 0 is selected, 0 : $RTP0_0$ to $RTP0_3$: pulse outputs are disabled. 1 : $RTP0_0$ to $RTP0_3$: pulse outputs are enabled. When pulse mode 1 is selected, 0 : $RTP0_0$ to $RTP0_3$, $RTP1_0$, $RTP1_1$: pulse outputs are disabled. 1 : $RTP0_0$ to $RTP0_3$, $RTP1_0$, $RTP1_1$: pulse outputs are enabled.	0	RW	

Note: When not using pulse output port 0 and three-phase waveform mode, be sure to fix these bits to "000_2."

Dead-time timer (Address $A7_{16}$)

b7 b0

Bit	Function	At reset	R/W	Reference
7 to 0	A value in the range from "00_{16}" to "FF_{16}" can be set.	Undefined	WO	10-7

Note: Use the **MOVMB** (**MOVM** when m = 1) or **STAB** (**STA** when m = 1) instruction for writing to this register.
Additionally, make sure writing to this register does not overlap with a trigger-occurrence timing of the dead-time timer.

Three-phase output data register 0 (Address $A8_{16}$)

■ Three-phase waveform mode

Three-phase output data register 0 (Address $A8_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
		X	X				

Bit	Bit name	Function	At reset	R/W
0	W-phase output fix bit	0 : Released from output fixation 1 : Output fixed	0	RW
1	V-phase output fix bit	0 : Released from output fixation 1 : Output fixed	0	RW
2	U-phase output fix bit	0 : Released from output fixation 1 : Output fixed	0	RW
3	W-phase output polarity set buffer (Valid in three-phase mode 0.) **(Note)**	0 : "H" output 1 : "L" output	0	RW
5, 4	Invalid in the three-phase waveform mode.		0	RW
6	Clock-source-of-dead-time-timer select bits	b7 b6 0 0 : f_2 0 1 : $f_2/2$ 1 0 : $f_2/4$ 1 1 : Do not select.	0	RW
7			0	RW

Reference: 10-9

X: It may be either "0" or "1."

Note: This bit is invalid in three-phase mode 1.

■ Pulse output port mode (Pluse output port 0)

Three-phase output data register 0 (Address $A8_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W
0	$RTP0_0$ pulse output data bit	0 : "L" level output 1 : "H" level output	0	RW
1	$RTP0_1$ pulse output data bit		0	RW
2	$RTP0_2$ pulse output data bit		0	RW
3	$RTP0_3$ pulse output data bit		0	RW
4	$RTP1_0$ pulse output data bit (Valid in pulse mode 1.) **(Note)**		0	RW
5	$RTP1_1$ pulse output data bit (Valid in pulse mode 1.) **(Note)**		0	RW
7, 6	Pulse output trigger select bits	b7 b6 0 0 : Underflow of timer A0 0 1 : Falling edge of input signal to pin RTP_{TRG0} 1 0 : Rising edge of input signal to pin RTP_{TRG0} 1 1 : Both falling and rising edges of input signal to pin RTP_{TRG0}	0	RW

Reference: 9-9

Note: This bit is invalid in pulse mode 0.

Three-phase output data register 1 (Address $A9_{16}$)

■ Three-phase waveform mode

Three-phase output data register 1 (Address $A9_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0
X	X			X			

Bit	Bit name	Function	At reset	R/W	Reference
0	W-phase fixed output's polarity set bit **(Note 1)**	0 : "H" output fixed 1 : "L" output fixed	0	RW	10-11
1	V-phase fixed output's polarity set bit **(Note 2)**	0 : "H" output fixed 1 : "L" output fixed	0	RW	
2	U-phase fixed output's polarity set bit **(Note 3)**	0 : "H" output fixed 1 : "L" output fixed	0	RW	
3	Invalid in the three-phase waveform mode.		0	RW	
4	V-phase output polarity set buffer (in three-phase mode 0)	0 : "H" output 1 : "L" output	0	RW	
	Interrupt request interval set bit (in three-phase mode 1)	0 : Every second time 1 : Every forth time			
5	U-phase output polarity set buffer (in three-phase mode 0)	0 : "H" output 1 : "L" output	0	RW	
	Interrupt validity output select bit (in three-phase mode 1)	0 : An interrupt request occurs at each even-numbered underflow of timer A3 1 : An interrupt request occurs at each odd-numbered underflow of timer A3			
7, 6	Invalid in the three-phase waveform mode.		0	RW	

X: It may be either "0" or "1."

Notes 1: Valid when the W-phase output fix bit (bit 0 at address $A8_{16}$) = "1." Be sure not to change the value during output of a fixed value.
2: Valid when the V-phase output fix bit (bit 1 at address $A8_{16}$) = "1." Be sure not to change the value during output of a fixed value.
3: Valid when the U-phase output fix bit (bit 2 at address $A8_{16}$) = "1." Be sure not to change the value during output of a fixed value.

■ Pulse output port mode (Pulse output port 0)

Three-phase output data register 1 (Address $A9_{16}$)

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	Pulse width modulation enable bit 0	0 : No pulse width modulation by timer A1 1 : Pulse width modulation by timer A1	0	RW	9-9
1	Pulse width modulation enable bit 1	0 : No pulse width modulation by timer A2 1 : Pulse width modulation by timer A2	0	RW	
2	Pulse width modulation enable bit 2	0 : No pulse width modulation by timer A4 1 : Pulse width modulation by timer A4	0	RW	
3	Pulse output polarity select bit	0 : Positive 1 : Negative	0	RW	
4	$RTP1_0$ pulse output data bit (Valid in pulse mode 0) **(Note)**	0 : "L" level output 1 : "H" level output	0	RW	
5	$RTP1_1$ pulse output data bit (Valid in pulse mode 0) **(Note)**		0	RW	
6	$RTP1_2$ pulse output data bit		0	RW	
7	$RTP1_3$ pulse output data bit		0	RW	

Note: This bit is invalid in pulse mode 1.

Position-data-retain function control register (Address AA_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	W-phase position data retain bit	Input level at pin IDW is read out. 0 : "L" level 1 : "H" level	0	RO	10-12
1	V-phase position data retain bit	Input level at pin IDV is read out. 0 : "L" level 1 : "H" level	0	RO	
2	U-phase position data retain bit	Input level at pin IDU is read out. 0 : "L" level 1 : "H" level	0	RO	
3	Retain-trigger polarity select bit	0 : Falling edge of positive phase 1 : Rising edge of positive phase	0	RW	
7 to 4	Nothing is assigned.		Undefined	—	

Note: This register is valid only in the three-phase mode.

Serial I/O pin control register (Address AC_{16})

b7 b6 b5 b4 b3 b2 b1 b0

Bit	Bit name	Function	At reset	R/W	Reference
0	$\overline{CTS_0}/\overline{RTS_0}$ separate select bit **(Note)**	0 : $\overline{CTS_0}/\overline{RTS_0}$ are used together. 1 : $\overline{CTS_0}/\overline{RTS_0}$ are separated.	0	RW	11-17 11-18
1	$\overline{CTS_1}/\overline{RTS_1}$ separate select bit **(Note)**	0 : $\overline{CTS_1}/\overline{RTS_1}$ are used together. 1 : $\overline{CTS_1}/\overline{RTS_1}$ are separated.	0	RW	
2	$TxD_0/P1_3$ switch bit	0 : Functions as TxD_0. 1 : Functions as $P1_3$.	0	RW	
3	$TxD_1/P1_7$ switch bit	0 : Functions as TxD_1. 1 : Functions as $P1_7$.	0	RW	
4	$\overline{CTS_2}/\overline{RTS_2}$ separate select bit **(Note)**	0 : $\overline{CTS_2}/\overline{RTS_2}$ are used together. 1 : $\overline{CTS_2}/\overline{RTS_2}$ are separated.	0	RW	
5	$TxD_2/P8_3$ switch bit	0 : Functions as TxD_2. 1 : Functions as $P8_3$.	0	RW	
7, 6	The value is "00" at reading.		0	—	

Note: Valid when the $\overline{CTS}/\overline{RTS}$ enable bit (bit 4 at addresses 34_{16}, $3C_{16}$, and $B4_{16}$) is "0."

Port P2 pin function control register (Address AE_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0							

Bit	Bit name	Function	At reset	R/W	Reference
0	Pin $TB0_{IN}$ select bit	0 : Allocate pin $TB0_{IN}$ to $P5_5$. 1 : Allocate pin $TB0_{IN}$ to $P2_4$.	0	RW	8-6
1	Pin $TB1_{IN}$ select bit	0 : Allocate pin $TB1_{IN}$ to $P5_6$. 1 : Allocate pin $TB1_{IN}$ to $P2_5$.	0	RW	
2	Pin $TB2_{IN}$ select bit	0 : Allocate pin $TB2_{IN}$ to $P5_7$. 1 : Allocate pin $TB2_{IN}$ to $P2_6$.	0	RW	
6 to 3	Nothing is assigned.		Undefined	—	
7	Fix this bit to “0.”		0	RW	

Clock control register 0 (Address BC_{16})

b7	b6	b5	b4	b3	b2	b1	b0
			1				1

Bit	Bit name	Function	At reset	R/W
0	Fix this bit to “1.”		1	RW
1	PLL circuit operation enable bit **(Note 1)**	0 : PLL frequency multiplier is inactive, and pin V_{CONT} is invalid. (Floating) 1 : PLL frequency multiplier is active, and pin V_{CONT} is valid.	1	RW
2	PLL multiplication ratio select bits **(Note 2)**	b3 b2 0 0 : Do not select. 0 1 : × 2 1 0 : × 3 1 1 : × 4	1	RW
3			0	RW
4	Fix this bit to “1.”		1	RW
5	System clock select bit **(Note 3)**	0 : fX_{IN} 1 : f_{PLL}	0	RW
6	Peripheral device’s clock select bit 0	See Table 4.2.2.	0	RW
7	Peripheral device’s clock select bit 1		0	RW

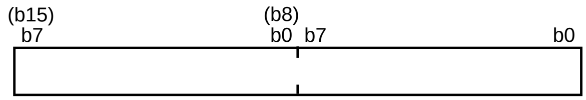
Reference: 4-6, 4-7

Notes 1: Clear this bit to “0” if the PLL frequency multiplier needs not to be active.
In the stop and flash memory parallel I/O modes, the PLL frequency multiplier is inactive and pin V_{CONT} is invalid regardless of the contents of this bit.

2: Rewriting of these bits must be performed simultaneously with clearance of the system clock select bit (bit 5) to “0.” Then, set bit 5 to “1” 2 ms after the rewriting of these bits. (After reset, these bits are allowed to be changed only once.)

3: Clearance of the PLL circuit operation enable bit (bit 1) to “0” clears the system clock select bit to “0.” Also, while the PLL circuit operation enable bit = “0,” nothing can be written to the system clock select bit. (Fixed to be “0.”)
Before setting of the system clock select bit to “1” after reset, it is necessary to insert an interval of 2 ms after the stabilization of $f(X_{IN})$.

Timer $A0_1$ register (Addresses $D1_{16}$, $D0_{16}$)
Timer $A1_1$ register (Addresses $D3_{16}$, $D2_{16}$)
Timer $A2_1$ register (Addresses $D5_{16}$, $D4_{16}$)



Bit	Function	At reset	R/W	Reference
15 to 0	Any value in the range from 0000_{16} to $FFFF_{16}$ can be set. Assuming that the set value = n, the “H” level width of the one-shot pulse is expressed as follows: n/f_i.	Undefined	WO	10-13

f_i: Frequency of a count source

Notes 1: Use the **MOVM** or **STA (STAD)** instruction for writing to this register. Additionally, make sure writing to this register must be performed in a unit of 16 bits.

2: This register is valid only in three-phase mode 1 of the three-phase waveform mode.

Comparator function select register 0 (Address DC_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	AN_0 pin comparator function select bit	0 : The comparator function is not selected. 1 : The comparator function is selected.	0	RW	12-12
1	AN_1 pin comparator function select bit		0	RW	
2	AN_2 pin comparator function select bit		0	RW	
3	AN_3 pin comparator function select bit		0	RW	
4	AN_4 pin comparator function select bit		0	RW	
5	AN_5 pin comparator function select bit		0	RW	
6	AN_6 pin comparator function select bit		0	RW	
7	AN_7 pin comparator function select bit		0	RW	

Note: Writing to comparator function select register 0 must be performed while the A-D converter halts.

Comparator function select register 1 (Address DD_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0				

Bit	Bit name	Function	At reset	R/W	Reference
0	AN_8 pin comparator function select bit	0 : The comparator function is not selected. 1 : The comparator function is selected.	0	RW	12-12
1	AN_9 pin comparator function select bit		0	RW	
2	AN_{10} pin comparator function select bit		0	RW	
3	AN_{11} pin comparator function select bit		0	RW	
7 to 4	Fix these bits to “0000.”		0	RW	

Note: Writing to comparator function select register 1 must be performed while the A-D converter halts.

Comparator result register 0 (Address DE_{16})

b7	b6	b5	b4	b3	b2	b1	b0

Bit	Bit name	Function	At reset	R/W	Reference
0	AN_0 pin comparator result bit	0 : The set value > The input level at pin AN_i 1 : The set value < The input level at pin AN_i	0	RW	12-12
1	AN_1 pin comparator result bit		0	RW	
2	AN_2 pin comparator result bit		0	RW	
3	AN_3 pin comparator result bit		0	RW	
4	AN_4 pin comparator result bit		0	RW	
5	AN_5 pin comparator result bit		0	RW	
6	AN_6 pin comparator result bit		0	RW	
7	AN_7 pin comparator result bit		0	RW	

Note: Writing to comparator result register 0 must be performed while the A-D converter halts.

Comparator result register 1 (Address DF_{16})

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0				

Bit	Bit name	Function	At reset	R/W	Reference
0	AN_8 pin comparator result bit	0 : The set value > The input level at pin AN_i 1 : The set value < The input level at pin AN_i	0	RW	12-12
1	AN_9 pin comparator result bit		0	RW	
2	AN_{10} pin comparator result bit		0	RW	
3	AN_{11} pin comparator result bit		0	RW	
7 to 4	Fix these bits to “0000.”		0	RW	

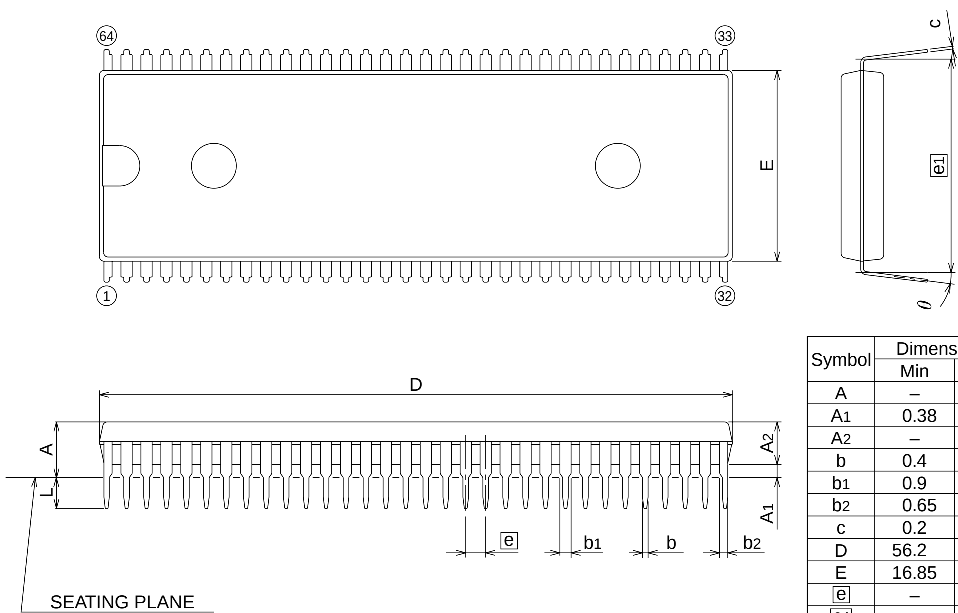
Note: Writing to comparator result register 1 must be performed while the A-D converter halts.

Appendix 3. Package outline

64P4B

Plastic 64pin 750mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP64-P-750-1.78	–	7.9	Alloy 42

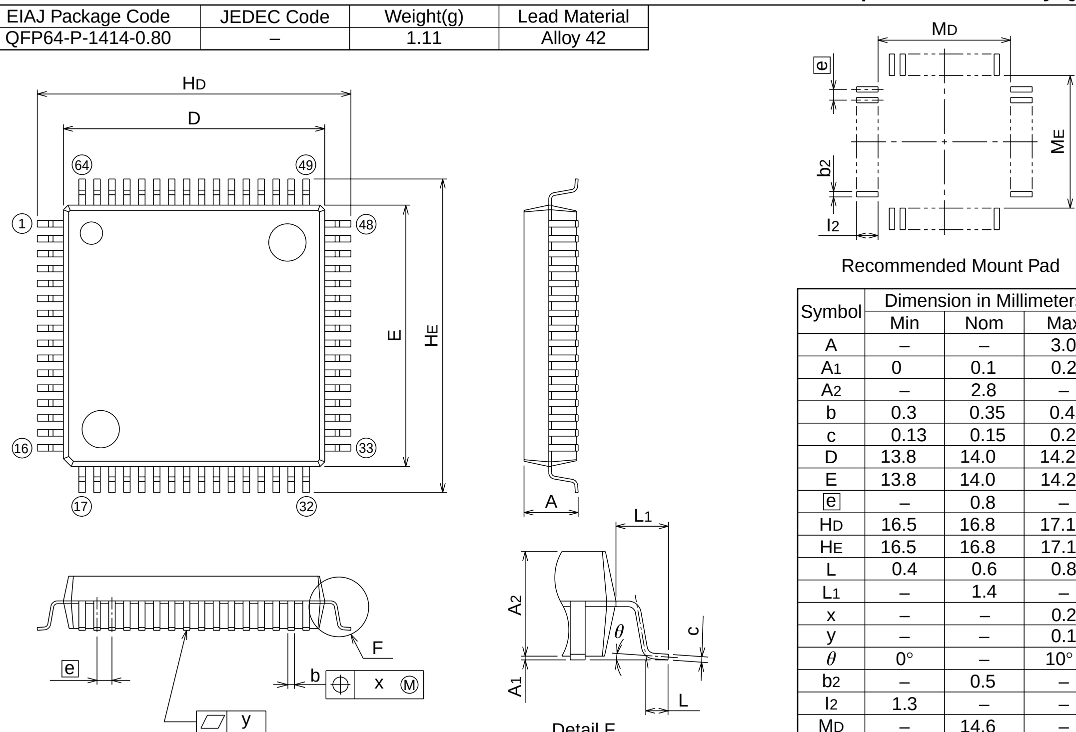


Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.08
A_1	0.38	–	–
A_2	–	3.8	–
b	0.4	0.5	0.6
b_1	0.9	1.0	1.3
b_2	0.65	0.75	1.05
c	0.2	0.25	0.32
D	56.2	56.4	56.6
E	16.85	17.0	17.15
e	–	1.778	–
e_1	–	19.05	–
L	2.8	–	–
θ	0°	–	15°

64P6N-A

Plastic 64pin 14X14mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP64-P-1414-0.80	–	1.11	Alloy 42



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A_1	0	0.1	0.2
A_2	–	2.8	–
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	13.8	14.0	14.2
e	–	0.8	–
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	–
x	–	–	0.2
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.5	–
l_2	1.3	–	–
M_D	–	14.6	–
M_E	–	14.6	–

Appendix 4. Examples of handling unused pins

When unusing an I/O pin, some handling is necessary for this pin. Examples of handling unused pins are described below.

The following are just examples. In actual use, the user shall modify them according to the user's application and properly evaluate their performance.

Table 1 Example of handling unused pins

Pin name	Handling example
P1, P2, P5 to P8	Set these pins to the input mode and connect each pin to Vcc or Vss via a resistor; or set these pins to the output mode and leave them open **(Note 1).**
$\overline{P4OUT_{CUT}}/\overline{INT_0}$, $\overline{P6OUT_{CUT}}/\overline{INT_4}$	Connect this pin to Vcc via a resistor. Select a falling edge for pins $\overline{INT_0}$ and $\overline{INT_4}$.
X_{OUT} **(Note 2)**, V_{CONT} **(Note 3)**	Leave these pins open.
AV_{CC}	Connect this pin to Vcc.
AV_{SS}, V_{REF}	Connect these pins to Vss.

Notes 1: When leaving these pins open after they have been set to the output mode, note the following: these port pins are placed in the input mode from reset until they are switched to the output mode by software. Therefore, voltage levels of these pins are undefined and the power source current may increase while these port pins are placed in the input mode.
Software reliability can be enhanced by setting the contents of the above ports' direction registers periodically. This is because these contents may be changed by noise, a program runaway which occurs owing to noise, etc.
For unused pins, use the shortest possible wiring (within 20 mm from the microcomputer's pins).

2: This applies when a clock externally generated is input to pin X_{IN}.

3: Be sure that the PLL circuit operation enable bit (bit 1 at address BC_{16}) = "0."

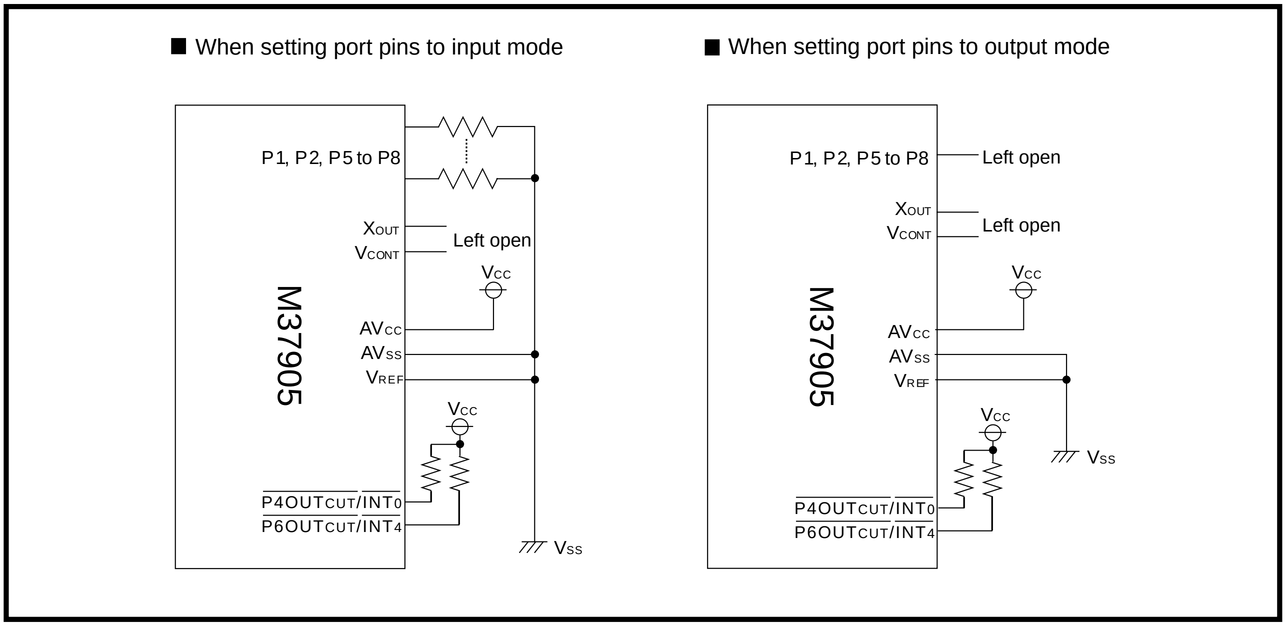


Fig. 1 Example of handling unused pins

Appendix 5. Hexadecimal instruction code table

INSTRUCTION CODE TABLE 0

D7–D4 \ D3–D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK IMP	Table 1	LDX DIR	ASL A	SEC IMP	SEI IMP		LDX ABS	LDAB A,(DIR),Y	LDAB A,L(DIR),Y	LDAB A,DIR	LDAB A,DIR,X	LDAB A,ABL	LDAB A,ABL,X	LDAB A,ABS	LDAB A,ABS,X
0001	1	BPL REL	Table 2	LDY DIR	ROL A	CLC IMP	CLI IMP	LDA A,IMM	LDY ABS	LDA A,(DIR),Y	LDA A,L(DIR),Y	LDA A,DIR	LDA A,DIR,X	LDA A,ABL	LDA A,ABL,X	LDA A,ABS	LDA A,ABS,X
0010	2	BRA REL	Table 3	CPX DIR	ANDB A,IMM	NEG A	SEM IMP	ADD A,IMM	LDXB IMM	LDAB A,IMM	ADDB A,IMM	ADD A,DIR	ADD A,DIR,X	LDAD E,IMM	ADDD E,IMM	ADD A,ABS	ADD A,ABS,X
0011	3	BMI REL	Table 4	CPY DIR	EORB A,IMM	EXTZ A	EXTS A	SUB A,IMM	LDYB IMM	CMPB A,IMM	SUBB A,IMM	SUB A,DIR	SUB A,DIR,X	CMPD E,IMM	SUBD E,IMM	SUB A,ABS	SUB A,ABS,X
0100	4	BGTU REL	Table 5	BBSB DIR,b,REL	LSR A	CLRB A	CLM IMP	CMP A,IMM	BBSB ABS,b,REL	MOVMB DIR/DIR		CMP A,DIR	CMP A,DIR,X	MOVMB DIR/ABS	MOVMB DIR/ABS,X	CMP A,ABS	CMP A,ABS,X
0101	5	BVC REL	Table 6	BBCB DIR,b,REL	ROR A	CLR A	XAB IMP	ORA A,IMM	BBCB ABS,b,REL	MOVM DIR/DIR		ORA A,DIR	ORA A,DIR,X	MOVM DIR/ABS	MOVM DIR/ABS,X	ORA A,ABS	ORA A,ABS,X
0110	6	BLEU REL	Table 7	CBEQB DIR/IMM,REL	ORAB A,IMM	ASR A	CLV IMP	AND A,IMM	PUL STK	MOVMB ABS/DIR	MOVMB ABS/DIR,X	AND A,DIR	AND A,DIR,X	MOVMB ABS/ABS		AND A,ABS	AND A,ABS,X
0111	7	BVS REL	Table 8	CBNEB DIR/IMM,REL		NOP IMP		EOR A,IMM	PLD n /RTLD n /RTSD n STK	MOVM ABS/DIR	MOVM ABS/DIR,X	EOR A,DIR	EOR A,DIR,X	MOVM ABS/ABS		EOR A,ABS	EOR A,ABS,X
1000	8	BGT REL	Table 9	INC DIR	PHD STK	RTS IMP	PHA STK	MOVM DIR/IMM	INC ABS	LDAD E,(DIR),Y	LDAD E,L(DIR),Y	LDAD E,DIR	LDAD E,DIR,X	LDAD E,ABL	LDAD E,ABL,X	LDAD E,ABS	LDAD E,ABS,X
1001	9	BCC REL	Table 10	DEC DIR	PLD STK	RTL IMP	PLA STK	MOVM ABS/IMM	DEC ABS	CLP IMM	SEP IMM	ADDD E,DIR	ADDD E,DIR,X	JMP ABS	JSR ABS	ADDD E,ABS	ADDD E,ABS,X
1010	A	BLE REL	Table 11	CBEQB A/IMM,REL	INC A	TXA IMP	PHP STK	CBEQ A/IMM,REL	BRAL REL	PSH STK	MOVMB DIR/IMM	SUBD E,DIR	SUBD E,DIR,X	JMPL ABL	JSRL ABL	SUBD E,ABS	SUBD E,ABS,X
1011	B	BCS REL	Table 12	CBNEB A/IMM,REL	DEC A	TYA IMP	PLP STK	CBNE A/IMM,REL		LDD n /PHD n /PHLD n STK/IMM	MOVMB ABS/IMM	CMPD E,DIR	CMPD E,DIR,X	JMP (ABS,X)	JSR (ABS,X)	CMPD E,ABS	CMPD E,ABS,X
1100	C	BGE REL	Table 13	CLRMB DIR	INX IMP	TAX IMP	PHX STK	LDX IMM	CLRMB ABS	STAB A,(DIR),Y	STAB A,L(DIR),Y	STAB A,DIR	STAB A,DIR,X	STAB A,ABL	STAB A,ABL,X	STAB A,ABS	STAB A,ABS,X
1101	D	BNE REL	Table 14	CLRM DIR	INY IMP	TAY IMP	PLX STK	LDY IMM	CLRM ABS	STA A,(DIR),Y	STA A,L(DIR),Y	STA A,DIR	STA A,DIR,X	STA A,ABL	STA A,ABL,X	STA A,ABS	STA A,ABS,X
1110	E	BLT REL	ABS A	STX DIR	DEX IMP	CLRX IMP	PHY STK	CPX IMM	STX ABS	STAD E,(DIR),Y	STAD E,L(DIR),Y	STAD E,DIR	STAD E,DIR,X	STAD E,ABL	STAD E,ABL,X	STAD E,ABS	STAD E,ABS,X
1111	F	BEQ REL	RTI IMP	STY DIR	DEY IMP	CLRY IMP	PLY STK	CPY IMM	STY ABS	← BSR REL →							

Note: Tables 1 through 14 specifies the contents of the INSTRUCTION CODE TABLE 1 through 14.
About the second word's codes, refer to the INSTRUCTION CODE TABLE 1 through 14.

Appendix 5. Hexadecimal instruction code table

INSTRUCTION CODE TABLE 1 (The first word's code of each instruction is 01_{16})

D7–D4	Hexadecimal notation	D3–D0: 0000–1111 (0–F)
0000	0	ADDX IMM
0001	1	ADDX IMM
0010	2	ADDY IMM
0011	3	ADDY IMM
0100	4	SUBX IMM
0101	5	SUBX IMM
0110	6	SUBY IMM
0111	7	SUBY IMM
1000	8	BSS A,b,REL
1001	9	
1010	A	BSC A,b,REL
1011	B	
1100	C	DXBNE IMM,REL
1101	D	DXBNE IMM,REL
1110	E	DYBNE IMM,REL
1111	F	DYBNE IMM,REL

INSTRUCTION CODE TABLE 2 (The first word's code of each instruction is 11_{16})

D7–D4 \ D3–D0	Hexadecimal notation	0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8	1001 9	1010 A	1011 B	1100 C	1101 D	1110 E	1111 F
0000	0	LDAB A,(DIR)	LDAB A,(DIR,X)	LDAB A,L(DIR)	LDAB A,SR	LDAB A,(SR),Y		LDAB A,ABS,Y									
0001	1	LDA A,(DIR)	LDA A,(DIR,X)	LDA A,L(DIR)	LDA A,SR	LDA A,(SR),Y		LDA A,ABS,Y									
0010	2	ADD A,(DIR)	ADD A,(DIR,X)	ADD A,L(DIR)	ADD A,SR	ADD A,(SR),Y		ADD A,ABS,Y		ADD A,(DIR),Y	ADD A,L(DIR),Y			ADD A,ABL	ADD A,ABL,X		
0011	3	SUB A,(DIR)	SUB A,(DIR,X)	SUB A,L(DIR)	SUB A,SR	SUB A,(SR),Y		SUB A,ABS,Y		SUB A,(DIR),Y	SUB A,L(DIR),Y			SUB A,ABL	SUB A,ABL,X		
0100	4	CMP A,(DIR)	CMP A,(DIR,X)	CMP A,L(DIR)	CMP A,SR	CMP A,(SR),Y		CMP A,ABS,Y		CMP A,(DIR),Y	CMP A,L(DIR),Y			CMP A,ABL	CMP A,ABL,X		
0101	5	ORA A,(DIR)	ORA A,(DIR,X)	ORA A,L(DIR)	ORA A,SR	ORA A,(SR),Y		ORA A,ABS,Y		ORA A,(DIR),Y	ORA A,L(DIR),Y			ORA A,ABL	ORA A,ABL,X		
0110	6	AND A,(DIR)	AND A,(DIR,X)	AND A,L(DIR)	AND A,SR	AND A,(SR),Y		AND A,ABS,Y		AND A,(DIR),Y	AND A,L(DIR),Y			AND A,ABL	AND A,ABL,X		
0111	7	EOR A,(DIR)	EOR A,(DIR,X)	EOR A,L(DIR)	EOR A,SR	EOR A,(SR),Y		EOR A,ABS,Y		EOR A,(DIR),Y	EOR A,L(DIR),Y			EOR A,ABL	EOR A,ABL,X		
1000	8	LDAD E,(DIR)	LDAD E,(DIR,X)	LDAD E,L(DIR)	LDAD E,SR	LDAD E,(SR),Y		LDAD E,ABS,Y									
1001	9	ADDD E,(DIR)	ADDD E,(DIR,X)	ADDD E,L(DIR)	ADDD E,SR	ADDD E,(SR),Y		ADDD E,ABS,Y		ADDD E,(DIR),Y	ADDD E,L(DIR),Y			ADDD E,ABL	ADDD E,ABL,X		
1010	A	SUBD E,(DIR)	SUBD E,(DIR,X)	SUBD E,L(DIR)	SUBD E,SR	SUBD E,(SR),Y		SUBD E,ABS,Y		SUBD E,(DIR),Y	SUBD E,L(DIR),Y			SUBD E,ABL	SUBD E,ABL,X		
1011	B	CMPD E,(DIR)	CMPD E,(DIR,X)	CMPD E,L(DIR)	CMPD E,SR	CMPD E,(SR),Y		CMPD E,ABS,Y		CMPD E,(DIR),Y	CMPD E,L(DIR),Y			CMPD E,ABL	CMPD E,ABL,X		
1100	C	STAB A,(DIR)	STAB A,(DIR,X)	STAB A,L(DIR)	STAB A,SR	STAB A,(SR),Y		STAB A,ABS,Y									
1101	D	STA A,(DIR)	STA A,(DIR,X)	STA A,L(DIR)	STA A,SR	STA A,(SR),Y		STA A,ABS,Y									
1110	E	STAD E,(DIR)	STAD E,(DIR,X)	STAD E,L(DIR)	STAD E,SR	STAD E,(SR),Y		STAD E,ABS,Y									
1111	F																

APPENDIX

Appendix 5. Hexadecimal instruction code table

INSTRUCTION CODE TABLE 3 (The first word's code of each instruction is 21_{16})

D7–D4 \ D3–D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0											ASL DIR	ASL DIR,X			ASL ABS	ASL ABS,X
0001	1											ROL DIR	ROL DIR,X			ROL ABS	ROL ABS,X
0010	2											LSR DIR	LSR DIR,X			LSR ABS	LSR ABS,X
0011	3											ROR DIR	ROR DIR,X			ROR ABS	ROR ABS,X
0100	4											ASR DIR	ASR DIR,X			ASR ABS	ASR ABS,X
0101	5																
0110	6																
0111	7																
1000	8	ADC A,(DIR)	ADC A,(DIR,X)	ADC A,L(DIR)	ADC A,SR	ADC A,(SR),Y		ADC A,ABS,Y		ADC A,(DIR),Y	ADC A,L(DIR),Y	ADC A,DIR	ADC A,DIR,X	ADC A,ABL	ADC A,ABL,X	ADC A,ABS	ADC A,ABS,X
1001	9	ADCD E,(DIR)	ADCD E,(DIR,X)	ADCD E,L(DIR)	ADCD E,SR	ADCD E,(SR),Y		ADCD E,ABS,Y		ADCD E,(DIR),Y	ADCD E,L(DIR),Y	ADCD E,DIR	ADCD E,DIR,X	ADCD E,ABL	ADCD E,ABL,X	ADCD E,ABS	ADCD E,ABS,X
1010	A	SBC A,(DIR)	SBC A,(DIR,X)	SBC A,L(DIR)	SBC A,SR	SBC A,(SR),Y		SBC A,ABS,Y		SBC A,(DIR),Y	SBC A,L(DIR),Y	SBC A,DIR	SBC A,DIR,X	SBC A,ABL	SBC A,ABL,X	SBC A,ABS	SBC A,ABS,X
1011	B	SBCD E,(DIR)	SBCD E,(DIR,X)	SBCD E,L(DIR)	SBCD E,SR	SBCD E,(SR),Y		SBCD E,ABS,Y		SBCD E,(DIR),Y	SBCD E,L(DIR),Y	SBCD E,DIR	SBCD E,DIR,X	SBCD E,ABL	SBCD E,ABL,X	SBCD E,ABS	SBCD E,ABS,X
1100	C	MPY (DIR)	MPY (DIR,X)	MPY L(DIR)	MPY SR	MPY (SR),Y		MPY ABS,Y		MPY (DIR),Y	MPY L(DIR),Y	MPY DIR	MPY DIR,X	MPY ABL	MPY ABL,X	MPY ABS	MPY ABS,X
1101	D	MPYS (DIR)	MPYS (DIR,X)	MPYS L(DIR)	MPYS SR	MPYS (SR),Y		MPYS ABS,Y		MPYS (DIR),Y	MPYS L(DIR),Y	MPYS DIR	MPYS DIR,X	MPYS ABL	MPYS ABL,X	MPYS ABS	MPYS ABS,X
1110	E	DIV (DIR)	DIV (DIR,X)	DIV L(DIR)	DIV SR	DIV (SR),Y		DIV ABS,Y		DIV (DIR),Y	DIV L(DIR),Y	DIV DIR	DIV DIR,X	DIV ABL	DIV ABL,X	DIV ABS	DIV ABS,X
1111	F	DIVS (DIR)	DIVS (DIR,X)	DIVS L(DIR)	DIVS SR	DIVS (SR),Y		DIVS ABS,Y		DIVS (DIR),Y	DIVS L(DIR),Y	DIVS DIR	DIVS DIR,X	DIVS ABL	DIVS ABL,X	DIVS ABS	DIVS ABS,X

INSTRUCTION CODE TABLE 4 (The first word's code of each instruction is 31_{16})

D7–D4 \ D3–D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			TAD,0 IMP					RLA A			ADDS IMM	SUBS IMM				
0001	1	WIT IMP		TAD,1 IMP								ADCB A,IMM	SBCB A,IMM	ADCD E,IMM	SBCD E,IMM		
0010	2			TAD,2 IMP								MVP BLK	MVN BLK				
0011	3	STP IMP		TAD,3 IMP								MOVMB DIR,X/IMM	MOVMB ABS,X/IMM				
0100	4	PHT STK		TDA,0 IMP					MOVM DIR,X/IMM			LDT IMM	PEI STK	PEA STK	PER STK		
0101	5	PLT STK		TDA,1 IMP					MOVM ABS,X/IMM			RMPA Multiplied accumulation		JMP (ABS)	JMPL L(ABS)		
0110	6	PHG STK		TDA,2 IMP													
0111	7	TSD IMP		TDA,3 IMP	TDS IMP												
1000	8	NEGD E		TAS IMP					ADC A,IMM								
1001	9	ABSD E		TSA IMP													
1010	A	EXTZD E							SBC A,IMM								
1011	B	EXTSD E															
1100	C			TXY IMP					MPY IMM								
1101	D			TYX IMP					MPYS IMM								
1110	E			TXS IMP					DIV IMM								
1111	F			TSX IMP					DIVS IMM								

INSTRUCTION CODE TABLE 5 (The first word's code of each instruction is 41_{16})

D_3–D_0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
D_7–D_4	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0						LDX DIR,Y	LDX ABS,Y									
0001	1												LDY DIR,X				LDY ABS,X
0010	2															CPX ABS	
0011	3															CPY ABS	
0100	4											BBS DIR,b,REL				BBS ABS,b,REL	
0101	5											BBC DIR,b,REL				BBC ABS,b,REL	
0110	6											CBEQ DIR/IMM,REL					
0111	7											CBNE DIR/IMM,REL					
1000	8												INC DIR,X				INC ABS,X
1001	9												DEC DIR,X				DEC ABS,X
1010	A																
1011	B																
1100	C																
1101	D																
1110	E						STX DIR,Y										
1111	F												STY DIR,X				

INSTRUCTION CODE TABLE 6 (The first word's code of each instruction is 51_{16})

D_3–D_0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
D_7–D_4	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			ADDMB DIR/IMM	ADDM DIR/IMM			ADDMB ABS/IMM	ADDM ABS/IMM								
0001	1			SUBMB DIR/IMM	SUBM DIR/IMM			SUBMB ABS/IMM	SUBM ABS/IMM								
0010	2			CMPMB DIR/IMM	CMPM DIR/IMM			CMPMB ABS/IMM	CMPM ABS/IMM								
0011	3			ORAMB DIR/IMM	ORAM DIR/IMM			ORAMB ABS/IMM	ORAM ABS/IMM								
0100	4																
0101	5																
0110	6			ANDMB DIR/IMM	ANDM DIR/IMM			ANDMB ABS/IMM	ANDM ABS/IMM								
0111	7			EORMB DIR/IMM	EORM DIR/IMM			EORMB ABS/IMM	EORM ABS/IMM								
1000	8				ADDMD DIR/IMM				ADDMD ABS/IMM								
1001	9				SUBMD DIR/IMM				SUBMD ABS/IMM								
1010	A				CMPMD DIR/IMM				CMPMD ABS/IMM								
1011	B				ORAMD DIR/IMM				ORAMD ABS/IMM								
1100	C																
1101	D																
1110	E				ANDMD DIR/IMM				ANDMD ABS/IMM								
1111	F				EORMD DIR/IMM				EORMD ABS/IMM								

Appendix 5. Hexadecimal instruction code table

INSTRUCTION CODE TABLE 7 (The first word's code of each instruction is 61_{16})

D7–D4 \ D3–D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	MOVRB DIR/IMM															→
0001	1	MOVR DIR/IMM															→
0010	2	MOVRB ABS/IMM															→
0011	3	MOVR ABS/IMM															→
0100	4	MOVRB DIR/DIR															→
0101	5	MOVR DIR/DIR															→
0110	6	MOVRB ABS/DIR															→
0111	7	MOVR ABS/DIR															→
1000	8	MOVRB DIR/ABS															→
1001	9	MOVR DIR/ABS															→
1010	A	MOVRB ABS/ABS															→
1011	B	MOVR ABS/ABS															→
1100	C																
1101	D																
1110	E																
1111	F																

INSTRUCTION CODE TABLE 8 (The first word's code of each instruction is 71_{16})

D7–D4 \ D3–D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	MOVRB DIR/ABS,X															→
0001	1	MOVR DIR/ABS,X															→
0010	2																
0011	3																
0100	4																
0101	5																
0110	6	MOVRB ABS/DIR,X															→
0111	7	MOVR ABS/DIR,X															→
1000	8								BSS DIR,b,REL								
1001	9																
1010	A								BSC DIR,b,REL								
1011	B																
1100	C								BSS ABS,b,REL								
1101	D																
1110	E								BSC ABS,b,REL								
1111	F																

INSTRUCTION CODE TABLE 9 (The first word's code of each instruction is 81_{16})

D7–D4 \ D3–D0	Hexadecimal notation	0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8	1001 9	1010 A	1011 B	1100 C	1101 D	1110 E	1111 F
0000	0				ASL B					LDAB B,(DIR),Y	LDAB B,L(DIR),Y	LDAB B,DIR	LDAB B,DIR,X	LDAB B,ABL	LDAB B,ABL,X	LDAB B,ABS	LDAB B,ABS,X
0001	1				ROL B			LDA B,IMM		LDA B,(DIR),Y	LDA B,L(DIR),Y	LDA B,DIR	LDA B,DIR,X	LDA B,ABL	LDA B,ABL,X	LDA B,ABS	LDA B,ABS,X
0010	2				ANDB B,IMM	NEG B		ADD B,IMM		LDAB B,IMM	ADDB B,IMM	ADD B,DIR	ADD B,DIR,X			ADD B,ABS	ADD B,ABS,X
0011	3				EORB B,IMM	EXTZ B	EXTS B	SUB B,IMM		CMPB B,IMM	SUBB B,IMM	SUB B,DIR	SUB B,DIR,X			SUB B,ABS	SUB B,ABS,X
0100	4				LSR B	CLRB B		CMP B,IMM				CMP B,DIR	CMP B,DIR,X			CMP B,ABS	CMP B,ABS,X
0101	5				ROR B	CLR B		ORA B,IMM				ORA B,DIR	ORA B,DIR,X			ORA B,ABS	ORA B,ABS,X
0110	6				ORAB B,IMM	ASR B		AND B,IMM				AND B,DIR	AND B,DIR,X			AND B,ABS	AND B,ABS,X
0111	7							EOR B,IMM				EOR B,DIR	EOR B,DIR,X			EOR B,ABS	EOR B,ABS,X
1000	8						PHB STK										
1001	9						PLB STK										
1010	A			CBEQB B/IMM,REL	INC B	TXB IMP		CBEQ B/IMM,REL									
1011	B			CBNEB B/IMM,REL	DEC B	TYB IMP		CBNE B/IMM,REL									
1100	C					TBX IMP				STAB B,(DIR),Y	STAB B,L(DIR),Y	STAB B,DIR	STAB B,DIR,X	STAB B,ABL	STAB B,ABL,X	STAB B,ABS	STAB B,ABS,X
1101	D					TBY IMP				STA B,(DIR),Y	STA B,L(DIR),Y	STA B,DIR	STA B,DIR,X	STA B,ABL	STA B,ABL,X	STA B,ABS	STA B,ABS,X
1110	E		ABS B														
1111	F																

INSTRUCTION CODE TABLE 10 (The first word's code of each instruction is 91_{16})

D7–D4 \ D3–D0	Hexadecimal notation	0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8	1001 9	1010 A	1011 B	1100 C	1101 D	1110 E	1111 F
0000	0	LDAB B,(DIR)	LDAB B,(DIR,X)	LDAB B,L(DIR)	LDAB B,SR	LDAB B,(SR),Y		LDAB B,ABS,Y									
0001	1	LDA B,(DIR)	LDA B,(DIR,X)	LDA B,L(DIR)	LDA B,SR	LDA B,(SR),Y		LDA B,ABS,Y									
0010	2	ADD B,(DIR)	ADD B,(DIR,X)	ADD B,L(DIR)	ADD B,SR	ADD B,(SR),Y		ADD B,ABS,Y		ADD B,(DIR),Y	ADD B,L(DIR),Y			ADD B,ABL	ADD B,ABL,X		
0011	3	SUB B,(DIR)	SUB B,(DIR,X)	SUB B,L(DIR)	SUB B,SR	SUB B,(SR),Y		SUB B,ABS,Y		SUB B,(DIR),Y	SUB B,L(DIR),Y			SUB B,ABL	SUB B,ABL,X		
0100	4	CMP B,(DIR)	CMP B,(DIR,X)	CMP B,L(DIR)	CMP B,SR	CMP B,(SR),Y		CMP B,ABS,Y		CMP B,(DIR),Y	CMP B,L(DIR),Y			CMP B,ABL	CMP B,ABL,X		
0101	5	ORA B,(DIR)	ORA B,(DIR,X)	ORA B,L(DIR)	ORA B,SR	ORA B,(SR),Y		ORA B,ABS,Y		ORA B,(DIR),Y	ORA B,L(DIR),Y			ORA B,ABL	ORA B,ABL,X		
0110	6	AND B,(DIR)	AND B,(DIR,X)	AND B,L(DIR)	AND B,SR	AND B,(SR),Y		AND B,ABS,Y		AND B,(DIR),Y	AND B,L(DIR),Y			AND B,ABL	AND B,ABL,X		
0111	7	EOR B,(DIR)	EOR B,(DIR,X)	EOR B,L(DIR)	EOR B,SR	EOR B,(SR),Y		EOR B,ABS,Y		EOR B,(DIR),Y	EOR B,L(DIR),Y			EOR B,ABL	EOR B,ABL,X		
1000	8																
1001	9																
1010	A																
1011	B																
1100	C	STAB B,(DIR)	STAB B,(DIR,X)	STAB B,L(DIR)	STAB B,SR	STAB B,(SR),Y		STAB B,ABS,Y									
1101	D	STA B,(DIR)	STA B,(DIR,X)	STA B,L(DIR)	STA B,SR	STA B,(SR),Y		STA B,ABS,Y									
1110	E																
1111	F																

Appendix 5. Hexadecimal instruction code table

INSTRUCTION CODE TABLE 11 (The first word's code of each instruction is $A1_{16}$)

D3–D0 / D7–D4	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0																
0001	1																
0010	2																
0011	3																
0100	4																
0101	5																
0110	6																
0111	7																
1000	8	ADC B,(DIR)	ADC B,(DIR,X)	ADC B,L(DIR)	ADC B,SR	ADC B,(SR),Y		ADC B,ABS,Y		ADC B,(DIR),Y	ADC B,L(DIR),Y	ADC B,DIR	ADC B,DIR,X	ADC B,ABL	ADC B,ABL,X	ADC B,ABS	ADC B,ABS,X
1001	9																
1010	A	SBC B,(DIR)	SBC B,(DIR,X)	SBC B,L(DIR)	SBC B,SR	SBC B,(SR),Y		SBC B,ABS,Y		SBC B,(DIR),Y	SBC B,L(DIR),Y	SBC B,DIR	SBC B,DIR,X	SBC B,ABL	SBC B,ABL,X	SBC B,ABS	SBC B,ABS,X
1011	B																
1100	C																
1101	D																
1110	E																
1111	F																

INSTRUCTION CODE TABLE 12 (The first word's code of each instruction is $B1_{16}$)

D3–D0 / D7–D4	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			TBD,0 IMP													
0001	1			TBD,1 IMP								ADCB B,IMM	SBCB B,IMM				
0010	2			TBD,2 IMP													
0011	3			TBD,3 IMP													
0100	4			TDB,0 IMP													
0101	5			TDB,1 IMP													
0110	6			TDB,2 IMP													
0111	7			TDB,3 IMP													
1000	8			TBS IMP					ADC B,IMM								
1001	9			TSB IMP													
1010	A								SBC B,IMM								
1011	B																
1100	C																
1101	D																
1110	E																
1111	F																

INSTRUCTION CODE TABLE 13 (The first word's code of each instruction is $C1_{16}$)

D7–D4 \ D3–D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0									LSR,#n A							
0001	1																
0010	2									ROR,#n A							
0011	3																
0100	4									ASL,#n A							
0101	5																
0110	6									ROL,#n A							
0111	7																
1000	8									ASR,#n A							
1001	9																
1010	A									DEBNE DIR/IMM,REL							
1011	B																
1100	C																
1101	D																
1110	E																
1111	F																

INSTRUCTION CODE TABLE 14 (The first word's code of each instruction is $D1_{16}$)

D7–D4 \ D3–D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0									LSRD,#n E							
0001	1																
0010	2									RORD,#n E							
0011	3																
0100	4									ASLD,#n E							
0101	5																
0110	6									ROLD,#n E							
0111	7																
1000	8									ASRD,#n E							
1001	9																
1010	A																
1011	B																
1100	C																
1101	D																
1110	E									DEBNE ABS/IMM,REL							
1111	F																

Appendix 6. Machine instructions

Note: For an instruction of which “Operation length (Bit)” = 16/8 is executed in the bit length described below.

- 16-bit length when m = 0 or x = 0.
- 8-bit length when m = 1 or x = 1.

For an instruction of which “Operation length (Bit)” = 8 or 32 is executed in 8-bit or 32-bit length regardless of the contents of flags m and x.

Symbol	Description	Symbol	Description
IMP	Implied addressing mode	E	Accumulator E
IMM	Immediate addressing mode	E_H	Accumulator E's high-order 16 bits (Accumulator B)
A	Accumulator addressing mode	E_L	Accumulator E's low-order 16 bits (Accumulator A)
DIR	Direct addressing mode	X	Index register X
DIR, X	Direct indexed X addressing mode	X_H	Index register X's high-order 8 bits
DIR, Y	Direct indexed Y addressing mode	X_L	Index register X's low-order 8 bits
(DIR)	Direct indirect addressing mode	Y	Index register Y
(DIR, X)	Direct indexed X indirect addressing mode	Y_H	Index register Y's high-order 8 bits
(DIR), Y	Direct indirect indexed Y addressing mode	Y_L	Index register Y's low-order 8 bits
L(DIR)	Direct indirect long addressing mode	S	Stack pointer
L(DIR), Y	Direct indirect long indexed Y addressing mode	REL	Relative address
ABS	Absolute addressing mode	PC	Program counter
ABS, X	Absolute indexed X addressing mode	PC_H	Program counter's high-order 8 bits
ABS, Y	Absolute indexed Y addressing mode	PC_L	Program counter's low-order 8 bits
ABL	Absolute long addressing mode	PG	Program bank register
ABL, X	Absolute long indexed X addressing mode	DT	Data back register
(ABS)	Absolute indirect addressing mode	DPR0	Direct page register 0
L(ABS)	Absolute indirect long addressing mode	$DPR0_H$	Direct page register 0's high-order 8 bits
(ABS, X)	Absolute indexed X indirect addressing mode	$DPR0_L$	Direct page register 0's low-order 8 bits
STK	Stack addressing mode	DPRn	Direct page register n
REL	Relative addressing mode	$DPRn_H$	Direct page register n's high-order 8 bits
DIR, b, R	Direct bit relative addressing mode	$DPRn_L$	Direct page register n's low-order 8 bits
ABS, b, R	Absolute bit relative addressing mode	PS	Processor status register
SR	Stack pointer relative addressing mode	PS_H	Processor status register's high-order 8 bits
(SR), Y	Stack pointer relative indirect indexed Y addressing mode	PS_L	Processor status register's low-order 8 bits
BLK	Block transfer addressing mode	PS_L(bit n)	nth bit in processor status register
Multiplied accumulation	Multiplied accumulation addressing mode	M	Contents of memory
op	Instruction code (Op code)	M(S)	Contents of memory at address indicated by stack pointer
n	Number of cycles	M(bit n)	nth bit of memory
#	Number of bytes	Mn	n-bit memory's address or contents
C	Carry flag	IMM	Immediate value (8 bits or 16 bits)
Z	Zero flag	IMMn	n-bit immediate value
I	Interrupt disable flag	IMM_H	16-bit immediate value's high-order 8 bits
D	Decimal operation mode flag	IMM_L	16-bit immediate value's low-order 8 bits
x	Index register length selection flag	AD_H	Value of 24-bit address's high-order 8 bits (A_{23}–A_{16})
m	Data length selection flag	AD_M	Value of 24-bit address's middle-order 8 bits (A_{15}–A_8)
V	Overflow flag	AD_L	Value of 24-bit address's low-order 8 bits (A_7–A_0)
N	Negative flag	EAR	Effective address (16 bits)
IPL	Processor interrupt priority level	EAR_H	Effective address's high-order 8 bits
+	Addition	EAR_L	Effective address's low-order 8 bits
–	Subtraction	imm	8-bit immediate value
×	Multiplication	imm_n	n-bit immediate value
÷	Division	dd	Displacement for DPR (8 bits or 16 bits)
∧	Logical AND	i	Number of transfer bytes, rotation or repeated operations
∨	Logical OR	i_1, i_2	Number of registers pushed or pulled
∀	Logical exclusive OR	source	Operand to specify transfer source
| |	Absolute value	dest	Operand to specify transfer destination
—	Negation		
→	Movement to the arrow direction		
←	Movement to the arrow direction		
⇔	Exchange		
Acc	Accumulator		
Acc_H	Accumulator's high-order 8 bits		
Acc_L	Accumulator's low-order 8 bits		
A	Accumulator A		
A_H	Accumulator A's high-order 8 bits		
A_L	Accumulator A's low-order 8 bits		
B	Accumulator B		
B_H	Accumulator B's high-order 8 bits		
B_L	Accumulator B's low-order 8 bits		

7900 Series Machine Instructions

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
ABS **(Note 1)**	Acc← | Acc |	16/8							E1	3	1																									
									81 E1	4	2																									
ABSD	E← | E |	32							31 90	5	2																									
ADC **(Notes 1 and 2)**	Acc←Acc + M + C	16/8				31 87	3	3				21 8A	5	3	21 8B	6	3				21 80	7	3	21 81	8	3	21 88	8	3	21 82	9	3	21 89	10	3
						B1 87	3	3				A1 8A	7	3	A1 8B	8	3				A1 80	9	3	A1 81	10	3	A1 88	10	3	A1 82	11	3	A1 89	12	3
ADCB **(Note 1)**	AccL←AccL + IMM8 + C	8				31 1A	3	3																												
						B1 1A	3	3																												
ADCD	E←E + M32 + C	32				31 1C	4	6				21 9A	7	3	21 9B	8	3				21 90	9	3	21 91	10	3	21 98	10	3	21 92	11	3	21 99	12	3
ADD **(Notes 1 and 2)**	Acc←Acc + M	16/8				26	1	2				2A	3	2	2B	4	2				11 20	6	3	11 21	7	3	11 28	7	3	11 22	8	3	11 29	9	3
						81 26	2	3				81 2A	4	3	81 2B	5	3				91 20	6	3	91 21	7	3	91 28	7	3	91 22	8	3	91 29	9	3
ADDB **(Note 1)**	AccL←AccL + IMM8	8				29	1	2																												
						81 29	2	3																												
ADDD	E←E + M32	32				2D	3	5				9A	6	2	9B	7	2				11 90	9	3	11 91	10	3	11 98	10	3	11 92	11	3	11 99	12	3
ADDM **(Note 3)**	M←M + IMM	16/8										51 03	7	4																						
ADDMB	M8←M8 + IMM8	8										51 02	7	4																						
ADDMD	M32←M32 + IMM32	32										51 83	10	7																						
ADDS	S←S + IMM8	16				31 0A	2	3																												
ADDX	X←X + IMM (IMM = 0 to 31)	16/8				01	2	2																												
ADDY **(Note 4)**	Y←Y + IMM (IMM = 0 to 31)	16/8				01 20 + imm	2	2																												

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																																•	•	•	0	V	•	•	•	•	Z	0
																																																•	•	•	0	V	•	•	•	•	Z	0
21 8E	5	4	21 8F	6	4	21 86	6	4	21 8C	6	5	21 8D	7	5																						21 83	6	3	21 84	9	3							•	•	•	N	V	•	•	•	•	Z	C
A1 8E	7	4	A1 8F	8	4	A1 86	8	4	A1 8C	8	5	A1 8D	9	5																						A1 83	8	3	A1 84	11	3																	
																																																•	•	•	N	V	•	•	•	•	Z	C
21 9E	7	4	21 9F	8	4	21 96	8	4	21 9C	8	5	21 9D	9	5																						21 93	8	3	21 94	11	3							•	•	•	N	V	•	•	•	•	Z	C
2E	3	3	2F	4	3	11 26	5	4	11 2C	5	5	11 2D	6	5																						11 93	5	3	11 24	8	3							•	•	•	N	V	•	•	•	•	Z	C
81 2E	4	4	81 2F	5	4	91 26	5	4	91 2C	5	5	91 2D	6	5																						91 23	5	3	91 24	8	3																	
																																																•	•	•	N	V	•	•	•	•	Z	C
9E	6	3	9F	7	3	11 96	8	4	11 9C	8	5	11 9D	9	5																						11 93	8	3	11 94	11	3							•	•	•	N	V	•	•	•	•	Z	C
51 07	7	5																																														•	•	•	N	V	•	•	•	•	Z	C
51 06	7	5																																														•	•	•	N	V	•	•	•	•	Z	C
51 87	10	8																																														•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	N	V	•	•	•	•	Z	C

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
AND **(Notes 1 and 2)**	Acc←Acc∧M	16/8				66	1	2				6A	3	2	6B	4	2				11 60	6	3	11 61	7	3	11 68	7	3	11 62	8	3	11 69	9	3
						81 66	2	3				81 6A	4	3	81 6B	5	3				91 60	6	3	91 61	7	3	91 68	7	3	91 62	8	3	91 69	9	3
ANDB **(Note 1)**	AccL←AccL∧IMM8	8				23	1	2																												
						81 23	2	3																												
ANDM **(Note 3)**	M←M∧IMM	16/8										51 63	7	4																						
ANDMB	M8←M8∧IMM8	8										51 62	7	4																						
ANDMD	M32←M32∧IMM32	32										51 E3	10	7																						
ASL **(Note 1)**	Arithmetic shift to the left by 1 bit m = 0 Acc or M16 C ← b15 … b0 ← 0 m = 1 AccL or M8 C ← b7 … b0 ← 0	16/8							03	1	1	21 0A	7	3	21 0B	8	3																			
									81 03	2	2																									
ASL #n **(Note 4)**	Arithmetic shift to the left by n bits (n = 0 to 15) m = 0 A C ← b15 … b0 ← 0 m = 1 AL C ← b7 … b0 ← 0	16/8							C1 40 + imm	6 + imm	2																									
ASLD #n **(Note 4)**	Arithmetic shift to the left by n bits (n = 0 to 31) E C ← b31 … b0 ← 0	32							D1 40 + imm	8 + imm	2																									
ASR **(Note 1)**	Arithmetic shift to the right by 1 bit m = 0 Acc or M16 → b15 … b0 → C m = 1 AccL or M8 → b7 … b0 → C	16/8							64	1	1	21 4A	7	3	21 4B	8	3																			
									81 64	2	2																									
ASR #n **(Note 4)**	Arithmetic shift to the right by n bits (n = 0 to 15) m = 0 A → b15 … b0 → C m = 1 AL → b7 … b0 → C	16/8							C1 80 + imm	6 + imm	2																									

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
6E	3	3	6F	4	3	11 66	5	4	11 6C	5	5	11 6D	6	5																						11 63	5	3	11 64	8	3							•	•	•	N	•	•	•	•	•	Z	•
81 6E	4	4	81 6F	5	4	91 66	5	4	91 6C	5	5	91 6D	6	5																						91 63	5	3	91 64	8	3																	
																																																•	•	•	N	•	•	•	•	•	Z	•
51 67	7	5																																														•	•	•	N	•	•	•	•	•	Z	•
51 66	7	5																																														•	•	•	N	•	•	•	•	•	Z	•
51 E7	10	8																																														•	•	•	N	•	•	•	•	•	Z	•
21 0E	7	4	21 0F	8	4																																											•	•	•	N	•	•	•	•	•	Z	C
																																																•	•	•	N	•	•	•	•	•	Z	C
																																																•	•	•	N	•	•	•	•	•	Z	C
21 4E	7	4	21 4F	8	4																																											•	•	•	N	•	•	•	•	•	Z	C
																																																•	•	•	N	•	•	•	•	•	Z	C

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
ASRD #n **(Note 4)**	Arithmetic shift to the right by n bits (n = 0 to 31) E → b31 … b0 ← C	32							D1 80 + imm	8 + imm	2																								
BBC **(Note 3)**	if M(bit n) = 0 then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	16/8																																	
BBCB	if M8(bit n) = 0 then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	8																																	
BBS **(Note 3)**	if M(bit n) = 1 then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	16/8																																	
BBSB	if M8(bit n) = 1 then PC←PC+cnt+REL (–128 to +127) (cnt: Number of bytes of instruction)	8																																	
BCC	if C = 0 then PC←PC + 2 + REL (–128 to +127)	–																																	
BCS	if C = 1 then PC←PC + 2 + REL (–128 to +127)	–																																	
BEQ	if Z = 1 then PC←PC + 2 + REL (–128 to +127)	–																																	
BGE	if N∀V = 0 then PC←PC + 2 + REL (–128 to +127)	–																																	
BGT	if Z = 0 and N∀V = 0 then PC←PC + 2 + REL (–128 to +127)	–																																	
BGTU	if C = 1 and Z = 0 then PC←PC + 2 + REL (–128 to +127)	–																																	
BLE	if Z = 1 or N∀V = 1 then PC←PC + 2 + REL (–128 to +127)	–																																	
BLEU	if C = 0 or Z = 1 then PC←PC + 2 + REL(–128 to +127)	–																																	
BLT	if N∀V = 1 then PC←PC + 2 + REL (–128 to +127)	–																																	

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																																•	•	•	N	•	•	•	•	•	Z	C
																														41 5A	9	5	41 5E	9	6													•	•	•	•	•	•	•	•	•	•	•
																														52	8	4	57	8	5													•	•	•	•	•	•	•	•	•	•	•
																														41 4A	9	5	41 4E	9	6													•	•	•	•	•	•	•	•	•	•	•
																														42	8	4	47	8	5													•	•	•	•	•	•	•	•	•	•	•
																											90	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											B0	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											F0	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											C0	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											80	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											40	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											A0	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											60	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											E0	6	2																			•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
BMI	if N = 1 then PC←PC + 2 + REL (–128 to +127)	–																																	
BNE	if Z = 0 then PC←PC + 2 + REL (–128 to +127)	–																																	
BPL	if N = 0 then PC←PC + 2 + REL (–128 to +127)	–																																	
BRA/BRAL **(Note 5)**	PC←PC + cn t + REL (BRA:–128 to +127, BRAL: –32768 to +32767) (cnt: Number of bytes of instruction) PG←PG + 1 (When carry occurs) PG←PG – 1 (When borrow occurs)	–																																	
BRK **(Note 6)**	PC←PC + 2 M(S)←PG S←S – 1 M(S)←PC_H S←S – 1 M(S)←PC_L S←S – 1 M(S)←PS_H S←S – 1 M(S)←PS_L S←S – 1 I←1 PC_L←AD_L PC_H←AD_M PG←00_{16} or FF_{16}	–	00 74	15	2																														
BSC **(Note 7)**	if A(bit n) or M(bit n) = 0 (n = 0 to 15), then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	16/8							01 A0 + n	7	3	71 A0 + n	11	4																					
BSR	(S)←PC + 2 PC←PC + 2 + REL (–1024 to +1023)	–																																	
BSS **(Note 7)**	if A(bit n) or M(bit n) = 1 (n = 0 to 15), then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	16/8							01 80 + n	7	3	71 80 + n	11	4																					
BVC	if V = 0 then PC←PC + 2 + REL (–128 to +127)	–																																	
BVS	if V = 1 then PC←PC + 2 + REL (–128 to +127)	–																																	

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																											30	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											D0	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											10	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											20	5	2																			•	•	•	•	•	•	•	•	•	•	•
																											A7	5	3																													
																																																•	•	•	•	•	•	•	•	1	•	•
71 E + n	10	5																																														•	•	•	•	•	•	•	•	•	•	•
																											F8 – FF	7	2																			•	•	•	•	•	•	•	•	•	•	•
71 C0 + n	10	5																																														•	•	•	•	•	•	•	•	•	•	•
																											50	6	2																			•	•	•	•	•	•	•	•	•	•	•
																											70	6	2																			•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
CBEQ **(Notes 1 and 3)**	if Acc = IMM or M = IMM then PC←PC + cnt + REL(–128 to +127) (cnt: Number of bytes of instruction)	16/8							A6 81 A6	6 7	3 4	41 6A	9	5																						
CBEQB **(Note 1)**	if Acc_L = IMM8 or M8 = IMM8 then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	8							A2 81 A2	6 7	3 4	62	8	4																						
CBNE **(Notes 1 and 3)**	if Acc ≠ IMM or M ≠ IMM then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	16/8							B6 81 B6	6 7	3 4	41 7A	9	5																						
CBNEB **(Note 1)**	if Acc_L ≠ IMM8 or M8 ≠ IMM8 then PC←PC+cnt+REL(–128 to +127) (cnt: Number of bytes of instruction)	8							B2 81 B2	6 7	3 4	72	8	4																						
CLC	C←0	–	14	1	1																														
CLI	I←0	–	15	3	1																														
CLM	m←0	–	45	3	1																														
CLP	PS_L(bit n)←0 (n = 0 to 7. Multiple bits can be specified.)	–				98	4	2																											
CLR **(Note 1)**	Acc←0	16/8							54 81 54	1 2	1 2																								
CLRB **(Note 1)**	Acc_L←00_{16}	8							44 81 44	1 2	1 2																								
CLRM	M←0	16/8										D2	5	2																						
CLRMB	M8←00_{16}	8										C2	5	2																						
CLRX	X←0	16/8	E4	1	1																														
CLRY	Y←0	16/8	F4	1	1																														

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																																•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	•	•	•	•	•	•	•	0
																																																•	•	•	•	•	•	•	•	0	•	•
																																																•	•	•	•	•	0	•	•	•	•	•
																																																•	•	•	Specified flag becomes “0.”							
																																																•	•	•	0	•	•	•	•	•	1	•
																																																•	•	•	0	•	•	•	•	•	1	•
D7	5	3																																														•	•	•	•	•	•	•	•	•	•	•
C7	5	3																																														•	•	•	•	•	•	•	•	•	•	•
																																																•	•	•	0	•	•	•	•	•	1	•
																																																•	•	•	0	•	•	•	•	•	1	•

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
CLV	V←0	–	65	1	1																														
CMP **(Notes 1 and 2)**	Acc – M	16/8				46	1	2				4A	3	2	4B	4	2				11 40	6	3	11 41	7	3	11 48	7	3	11 42	8	3	11 49	9	3
						81 46	2	3				81 4A	4	3	81 4B	5	3				91 40	6	3	91 41	7	3	91 48	7	3	91 42	8	3	91 49	9	3
CMPB **(Note 1)**	Acc_L – IMM8	8				38	1	2																											
						81 38	2	3																											
CMPD	E – M32	32				3C	3	5				BA	6	2	BB	7	2				11 B0	9	3	11 B1	10	3	11 B8	10	3	11 B2	11	3	11 B9	12	3
CMPM **(Note 3)**	M – IMM	16/8										51 23	5	4																					
CMPMB	M8 – IMM8	8										51 22	5	4																					
CMPMD	M32 – IMM32	32										51 A3	7	7																					
CPX **(Note 8)**	X – M	16/8				E6	1	2				22	3	2																					
CPY **(Note 8)**	Y – M	16/8				F6	1	2				32	3	2																					
DEBNE **(Note 4)**	M←M – IMM(IMM = 0 to 31) if M ≠ 0, then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	16/8										C1 A0 + imm	12	4																					
DEC **(Note 1)**	Acc←Acc – 1 or M←M – 1	16/8							B3	1	1	92	6	2	41 9B	8	3																		
									81 B3	2	2																								
DEX	X←X – 1	16/8	E3	1	1																														
DEY	Y←Y – 1	16/8	F3	1	1																														
DIV **(Notes 2, 9, and 10)**	A (quotient) ← (B, A) ÷ M B (remainder)	16/8				31 E7	15	3				21 EA	16	3	21 EB	17	3				21 E0	18	3	21 E1	19	3	21 E8	19	3	21 E2	20	3	21 E9	21	3

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																																•	•	•	•	0	•	•	•	•	•	•
4E	3	3	4F	4	3	11 46	5	4	11 4C	5	5	11 4D	6	5																						11 43	5	3	11 44	8	3							•	•	•	N	V	•	•	•	•	Z	C
81 4E	4	4	81 4F	5	4	91 46	5	4	91 4C	5	5	91 4D	6	5																						91 43	5	3	91 44	8	3																	
																																																•	•	•	N	V	•	•	•	•	Z	C
BE	6	3	BF	7	3	11 B6	8	4	11 BC	8	5	11 BD	9	5																						11 B3	8	3	11 B4	11	3							•	•	•	N	V	•	•	•	•	Z	C
51 27	5	5																																														•	•	•	N	V	•	•	•	•	Z	C
51 26	5	5																																														•	•	•	N	V	•	•	•	•	Z	C
51 A7	7	8																																														•	•	•	N	V	•	•	•	•	Z	C
41 2E	4	4																																														•	•	•	N	V	•	•	•	•	Z	C
41 3E	4	4																																														•	•	•	N	V	•	•	•	•	Z	C
D1 E0 + imm	11	5																																														•	•	•	•	•	•	•	•	•	•	•
97	6	3	41 9F	8	4																																											•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	N	•	•	•	•	•	Z	•
21 EE	16	4	21 EF	17	4	21 E6	17	4	21 EC	17	5	21 ED	18	5																						21 E3	17	3	21 E4	20	3							•	•	•	N	V	•	•	•	I	Z	C

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
DIVS **(Notes 2, 9, and 10)**	A (quotient) ←(B, A) ÷ M B (remainder) (Signed)	16/8				31 F7	22	3				21 FA	23	3	21 FB	24	3				21 F0	25	3	21 F1	26	3	21 F8	26	3	21 F2	27	3	21 F9	28	3
DXBNE **(Note 4)**	X←X – IMM (IMM = 0 to 31) if X ≠ 0, then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	16/8				01 C0 + imm	7	3																												
DYBNE **(Note 4)**	Y←Y – IMM (IMM = 0 to 31) if Y≠0, then PC←PC + cnt + REL (–128 to +127) (cnt: Number of bytes of instruction)	16/8				01 E0 + imm	7	3																												
EOR **(Notes 1 and 2)**	Acc←Acc∀M	16/8				76 81 76	1 2	2 3				7A 81 7A	3 4	2 3	7B 81 7B	4 5	2 3				11 70 91 70	6 6	3 3	11 71 91 71	7 7	3 3	11 78 91 78	7 7	3 3	11 72 91 72	8 8	3 3	11 79 91 79	9 9	3 3
EORB **(Note 1)**	Acc_L←Acc_L∀IMMB	8				33 81 33	1 2	2 3																												
EORM **(Note 3)**	M←M∀IMM	16/8										51 73	7	4																						
EORMB	M8←M8∀IMM8	8										51 72	7	4																						
EORMD	M32←M32∀IMM32	32										51 F3	10	7																						
EXTS **(Note 1)**	Acc ←Acc_L (Extension sign) (Bit 7 of Acc_L = 0) b_{15} b_7 b_0 00000000 0 Acc_H Acc_L (Bit 7 of Acc_L = 1) b_{15} b_7 b_0 11111111 1 Acc_H Acc_L	16							35 81 35	1 2	1 2																									
EXTSD	E←E_L(= A) (Extension sign) (Bit 15 of A = 0) b_{15} b_0 b_{15} b_0 0000_{16} 0 E_H(B) E_L(A) (Bit 15 of A = 1) b_{15} b_0 b_{15} b_0 $FFFF_{16}$ 1 E_H(B) E_L(A)	32							31 B0	5	2																									
EXTZ **(Note 1)**	Acc ← Acc_L (Extension zero) b_{15} b_8 b_7 b_0 00000000 Acc_H Acc_L	16							34 81 34	1 2	1 2																									
EXTZD	E←E_L(= A) (Extension zero) b_{15} b_0 b_{15} b_0 0000_{16} E_H(B) E_L(A)	32							31 A0	3	2																									

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
21 FE	23	4	21 FF	24	4	21 F6	24	4	21 FC	24	5	21 FD	25	5																						21 F3	24	3	21 F4	27	3							•	•	•	N	V	•	•	•	I	Z	C
																																																•	•	•	•	•	•	•	•	•	•	•
																																																•	•	•	•	•	•	•	•	•	•	•
7E	3	3	7F	4	3	11 76	5	4	11 7C	5	5	11 7D	6	5																						11 73	5	3	11 74	8	3							•	•	•	N	•	•	•	•	•	Z	•
81 7E	4	4	81 7F	5	4	91 76	5	4	91 7C	5	5	91 7D	6	5																						91 73	5	3	91 74	8	3																	
																																																•	•	•	N	•	•	•	•	•	Z	•
51 77	7	5																																														•	•	•	N	•	•	•	•	•	Z	•
51 76	7	5																																														•	•	•	N	•	•	•	•	•	Z	•
51 F7	10	8																																														•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	0	•	•	•	•	•	Z	•
																																																•	•	•	0	•	•	•	•	•	Z	•

Symbol	Function	Operation length (Bit)	IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
INC **(Note 1)**	Acc←Acc + 1 or M←M + 1	16/8							A3 81 A3	1 2	1 2	82	6	2	41 8B	8	3																		
INX	X←X + 1	16/8	C3	1	1																														
INY	Y←Y + 1	16/8	D3	1	1																														
JMP/JMPL	When ABS specified PC_L←AD_L PC_H←AD_M When ABL specified PC_L←AD_L PC_H←AD_M PG←AD_H When (ABS) specified PC_L←(AD_M, AD_L) PC_H←(AD_M, AD_L + 1) When L(ABS) specified PC_L←(AD_M, AD_L) PC_H←(AD_M, AD_L + 1) PG←(AD_M, AD_L + 2) When (ABS,X) specified PC_L←(AD_M, AD_L + X) PC_H←(AD_M, AD_L + X + 1)	–																																	
JSR/JSRL	When ABS specified PC←PC + 3 M(S)←PC_H S←S–1 M(S)←PC_L S←S–1 PC_L←AD_L PC_H←AD_M When ABL specified PC←PC + 4 M(S)←PG S←S – 1 M(S)←PC_H S←S – 1 M(S)←PC_L S←S – 1 PC_L←AD_L PC_H←AD_M PG←AD_H When (ABS,X) specified PC←PC + 3 M(S)←PC_H S←S – 1 M(S)←PC_L S←S – 1 PC_L←(AD_M, AD_L + X) PC_H←(AD_M, AD_L + X + 1)	–																																	
LDA **(Notes 1 and 2)**	Acc←M	16/8				16 81 16	1 2	2 3				1A 81 1A	3 4	2 3	1B 81 1B	4 5	2 3				11 10 91 10	6 6	3 3	11 11 91 11	7 7	3 3	18 81 18	6 7	2 3	11 12 91 12	8 8	3 3	19 81 19	8 9	2 3
LDAB **(Note 1)**	Acc←M8 (Extension zero)	16				28 81 28	1 2	2 3				0A 81 0A	3 4	2 3	0B 81 0B	4 5	2 3				11 00 91 00	6 6	3 3	11 01 91 01	7 7	3 3	08 81 08	6 7	2 3	11 02 91 02	8 8	3 3	09 81 09	8 9	2 3

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
87	6	3	41 8F	8	4																																											•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	N	•	•	•	•	•	Z	•
9C	4	3							AC	5	4				31 5C	7	4	31 5D	9	4	BC	7	3																									•	•	•	•	•	•	•	•	•	•	•
9D	6	3							AD	7	4										BD	8	3																									•	•	•	•	•	•	•	•	•	•	•
1E	3	3	1F	4	3	11 16	5	4	1C	4	4	1D	5	4																						11 13	5	3	11 14	8	3							•	•	•	N	•	•	•	•	•	Z	•
81 1E	4	4	81 1F	5	4	91 16	5	4	81 1C	5	5	81 1D	6	5																						91 13	5	3	91 14	8	3																	
0E	3	3	0F	4	3	11 06	5	4	0C	4	4	0D	5	4																						11 03	5	3	11 04	8	3							•	•	•	0	•	•	•	•	•	Z	•
81 0E	4	4	81 0F	5	4	91 06	5	4	81 0C	5	5	81 0D	6	5																						91 03	5	3	91 04	8	3																	

Symbol	Function	Operation length (Bit)	IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
LDAD	E←M32	32				2C	3	5				8A	6	2	8B	7	2				11 80	9	3	11 81	10	3	88	9	2	11 82	11	3	89	11	2
LDD n **(Notes 11 and 12)**	DPRn←IMM16 (n = 0 to 3. Multiple DPRs can be specified.)	16				B8 ?0 B8 ?0	13 11 + 2 i	4 2 + 2 i																												
LDT	DT←IMM8	8				31 4A	4	3																												
LDX **(Note 8)**	X←M	16/8				C6	1	2				02	3	2				41 05	5	3																
LDXB	X←IMM8 (Extension zero)	16				27	1	2																												
LDY **(Note 8)**	Y←M	16/8				D6	1	2				12	3	2	41 1B	5	3																			
LDYB	Y←IMM8 (Extension zero)	16				37	1	2																												
LSR **(Note 1)**	Logical shift to the right by 1 bit m = 0 Acc or M16 0→[b_{15} … b_0]→C m = 1 Acc_L or M8 0→[b_7 … b_0]→C	16/8							43 81 43	1 2	1 2	21 2A	7	3	21 2B	8	3																			
LSR #n **(Note 4)**	Logical shift to the right by n bits (n = 0 to 15) m = 0 A 0→[b_{15} … b_0]→C m = 1 A_L 0→[b_7 … b_0]→C	16/8							C1	6 + imm	2																									
LSRD #n **(Note 4)**	Logical shift to the right by n bits (n = 0 to 31) E 0→[b_{31} … b_0]→C	32							D1	8 + imm	2																									

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
8E	6	3	8F	7	3	11 86	8	4	8C	7	4	8D	8	4																						11 83	8	3	11 84	11	3							•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	•	•	•	•	•	•	•	•
																																																•	•	•	•	•	•	•	•	•	•	•
07	3	3				41 06	5	4																																								•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	0	•	•	•	•	•	Z	•
17	3	3	41 1F	5	4																																											•	•	•	N	•	•	•	•	•	Z	•
																																																•	•	•	0	•	•	•	•	•	Z	•
21 2E	7	4	21 2F	8	4																																											•	•	•	0	•	•	•	•	•	Z	C
																																																•	•	•	0	•	•	•	•	•	Z	C
																																																•	•	•	0	•	•	•	•	•	Z	C

Symbol	Function	Operation length (Bit)			Destination																																
					IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
					op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
MOVM **(Note 2)**	m = 0 M16(dest)←M16(source) m = 1 M8(dest)←M8(source)	16/8	Source	IMM										86	5	3	31 47	7	4																		
				DIR										58	6	3																					
				DIR, X																																	
				ABS										5C	6	4																					
				ABS, X										5D	7	4																					
MOVMB	M8(dest)←M8(source)	8	Source	IMM										A9	5	3	31 3A	7	4																		
				DIR										48	6	3																					
				DIR, X																																	
				ABS										4C	6	4																					
				ABS, X										4D	7	4																					
MOVR **(Notes 7 and 13)**	m = 0 M16(dest1) ←M16(source1) ⋮ ⋮ M16(dest n)←M16(source n) m = 1 M8(dest1) ←M8(source1) ⋮ ⋮ M8(dest n)←M8(source n) (n = 0 to 15)	16/8	Source	IMM										61 10 + n	3 + 5n	2 + 2n																					
				DIR										61 50 + n	3 + 6n	2 + 2n																					
				DIR, X																																	
				ABS										61 90 + n	3 + 6n	2 + 3n																					
				ABS, X										71 10 + n	3 + 6n	2 + 3n																					
MOVRB **(Note 7)**	M8(dest1) ←M8(source1) ⋮ ⋮ M8(dest n)←M8(source n) (n = to 15)	8	Source	IMM										61 00 + n	3 + 5n	2 + 2n																					
				DIR										61 40 + n	3 + 6n	2 + 2n																					
				DIR, X																																	
				ABS										61 80 + n	3 + 6n	2 + 3n																					
				ABS, X										71 00 + n	3 + 6n	2 + 3n																					

Destination																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
96	4	4	31 57	6	5																																											•	•	•	•	•	•	•	•	•	•	•
78	5	4																																																								
79	6	4																																																								
7C	5	5																																																								
B9	4	4	31 3B	6	5																																											•	•	•	•	•	•	•	•	•	•	•
68	5	4																																																								
69	6	4																																																								
6C	5	5																																																								
61 30 + n	3 + 4n	2 + 3n																																														•	•	•	•	•	•	•	•	•	•	•
61 70 + n	3 + 5n	2 + 3n																																																								
71 70 + n	3 + 6n	2 + 3n																																																								
61 B0 + n	3 + 5n	2 + 4n																																																								
61 20 + n	3 + 4n	2 + 3n																																														•	•	•	•	•	•	•	•	•	•	•
61 60 + n	3 + 5n	2 + 3n																																																								
71 60 + n	3 + 6n	2 + 3n																																																								
61 A0 + n	3 + 5n	2 + 4n																																																								

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
MPY **(Notes 2 and 14)**	(B, A)←A ✕ M	16/8				31 C7	8	3				21 CA	9	3	21 CB	10	3				21 C0	11	3	21 C1	12	3	21 C8	12	3	21 C2	13	3	21 C9	14	3
MPYS **(Notes 2 and 14)**	(B, A)←A ✕ M (Signed)	16/8				31 D7	8	3				21 DA	9	3	21 DB	10	3				21 D0	11	3	21 D1	12	3	21 D8	12	3	21 D2	13	3	21 D9	14	3
MVN **(Note 15)**	M(Y + k)←M(X + k) k = 0 to i – 1 (i: Number of transfer bytes specified by accumulator A)	16/8																																	
MVP **(Note 16)**	M(Y–k)←M(X–k) k = 0 to i–1 (i: Number of transfer bytes specified by accumulator A)	16/8																																	
NEG **(Note 1)**	Acc← –Acc	16/8							24	1	1																								
									81 24	2	2																								
NEGD	E← –E	32							31 80	4	2																								
NOP	PC←PC + 1 When catty occurs in PC PG←PG + 1	–	74	1	1																														
ORA **(Notes 1 and 2)**	Acc←Acc∨M	16/8				56	1	2				5A	3	2	5B	4	2				11 50	6	3	11 51	7	3	11 58	7	3	11 52	8	3	11 59	9	3
						81 56	2	3				81 5A	4	3	81 5B	5	3				91 50	6	3	91 51	7	3	91 58	7	3	91 52	8	3	91 59	9	3
ORAB **(Note 1)**	Acc_L←Acc_L∨IMM8	8				63	1	2																											
						81 63	2	3																											
ORAM **(Note 3)**	M←M∨IMM	16/8										51 33	7	4																					
ORAMB	M8←M8∨IMM8	8										51 32	7	4																					
ORAMD	M32←M32∨IMM32	32										51 B3	10	7																					
PEA	M(S)←IMM_H S←S – 1 M(S)←IMM_L S←S – 1	16																																	
PEI	M(S)←M((DPRn) + dd + 1) S←S + 1 M(S)←M((DPRn)+dd) S←S – 1 (n = 0 to 3)	16																																	

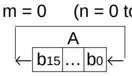
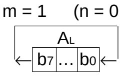
Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
21 CE	9	4	21 CF	10	4	21 C6	10	4	21 CC	10	5	21 CD	11	5																						21 C3	10	3	21 C4	13	3							•	•	•	N	•	•	•	•	•	Z	0
21 DE	9	4	21 DF	10	4	21 D6	10	4	21 DC	10	5	21 DD	11	5																						21 D3	10	3	21 D4	13	3							•	•	•	N	•	•	•	•	•	Z	0
																																										31 2B	5 + 5 i	4				•	•	•	•	•	•	•	•	•	•	•
																																										31 2A	9 + 5 i	4				•	•	•	•	•	•	•	•	•	•	•
																																																•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	•	•	•	•	•	•	•	•
5E	3	3	5F	4	3	11 56	5	4	11 5C	5	5	11 5D	6	5																						11 53	5	3	11 54	8	3							•	•	•	N	•	•	•	•	•	Z	•
81 5E	4	4	81 5F	5	4	91 56	5	4	91 5C	5	5	91 5D	6	5																						91 53	5	3	91 54	8	3																	
																																																•	•	•	N	•	•	•	•	•	Z	•
51 37	7	5																																														•	•	•	N	•	•	•	•	•	Z	•
51 36	7	5																																														•	•	•	N	•	•	•	•	•	Z	•
51 B7	10	8																																														•	•	•	N	•	•	•	•	•	Z	•
																								31 4C	5	4																						•	•	•	•	•	•	•	•	•	•	•
																								31 4B	7	3																						•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
PER	EAR←PC + IMM16 M(S)←EAR_H S←S – 1 M(S)←EAR_L S←S – 1	16																																	
PHA	m = 0 M(S)←A_H S←S – 1 M(S)←A_L S←S – 1 m = 1 M(S)←A_L S←S – 1	16/8																																	
PHB	m = 0 M(S)←B_H S←S – 1 M(S)←B_L S←S – 1 m=1 M(S)←B_L S←S – 1	16/8																																	
PHD	M(S)←$DPR0_H$ S←S – 1 M(S)←$DPR0_L$ S←S – 1	16																																	
PHD n **(Note 11)**	M(S)←$DPRn_H$ S←S – 1 M(S)←$DPRn_L$ S←S – 1 (n = 0 to 3) When multiple DPRs are specified, the above operations are repeated.	16																																	
PHG	M(S)←PG S←S – 1	8																																	
PHLD n **(Note 11)**	M(S)←$DPRn_H$ S←S – 1 M(S)←$DPRn_L$ S←S – 1 DPRn←IMM16 (n = 0 to 3) When multiple DPRs are specified, the above operations are repeated.	16																																	
PHP	M(S)←PS_H S←S – 1 M(S)←PS_L S←S – 1	16																																	
PHT	M(S)←DT S←S – 1	8																																	

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																								31 4D	6	4																								•	•	•	•	•	•	•	•	•	•	•
																								85	4	1																								•	•	•	•	•	•	•	•	•	•	•
																								81 85	5	2																								•	•	•	•	•	•	•	•	•	•	•
																								83	4	1																								•	•	•	•	•	•	•	•	•	•	•
																								B8 01 0F	12	2																								•	•	•	•	•	•	•	•	•	•	•
																								B8 01 | 0F	11 +	2 i																																		
																								31 60	4	2																								•	•	•	•	•	•	•	•	•	•	•
																								B8 01 | 0F	14	4																								•	•	•	•	•	•	•	•	•	•	•
																								B8 01 | 0F	11 + 3 i	2 + 2 i																																		
																								A5	4	1																								•	•	•	•	•	•	•	•	•	•	•
																								31 40	4	2																								•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
PHX	x = 0 M(S)←X_H S←S – 1 M(S)←X_L S←S – 1 x = 1 M(S)←X_L S←S – 1	16/8																																	
PHY	x = 0 M(S)←Y_H S←S – 1 M(S)←Y_L S←S – 1 x = 1 M(S)←Y_L S←S – 1	16/8																																	
PLA	m = 0 S←S + 1 A_L←M(S) S←S + 1 A_H←M(S) m = 1 S←S + 1 A_L←M(S)	16/8																																	
PLB	m = 0 S←S + 1 B_L←M(S) S←S + 1 B_H←M(S) m = 1 S←S + 1 B_L←M(S)	16/8																																	
PLD	S←S + 1 $DPR0_L$←M(S) S←S + 1 $DPR0_H$←M(S)	16																																	
PLD n **(Notes 11 and 12)**	S←S + 1 $DPRn_L$←M(S) S←S + 1 $DPRn_H$←M(S) (n = 0 to 3) When multiple DPRs are specified, the above operations are repeated.	16																																	
PLP **(Note 22)**	S←S + 1 PS_L←M(S) S←S + 1 PS_H←M(S)	16																																	
PLT	S←S + 1 DT←M(S)	8																																	

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																								C5	4	1																						•	•	•	•	•	•	•	•	•	•	•
																								E5	4	1																						•	•	•	•	•	•	•	•	•	•	•
																								95	4	1																						•	•	•	N	•	•	•	•	•	Z	•
																								81 95	5	2																						•	•	•	N	•	•	•	•	•	Z	•
																								93	5	1																						•	•	•	•	•	•	•	•	•	•	•
																								77 ?0	11	2																						•	•	•	•	•	•	•	•	•	•	•
																								77 ?0	8 + 3 i	2																																
																								B5	5	1																						Value restored from stack										
																								31 50	6	2																						•	•	•	N	•	•	•	•	•	Z	•

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
PLX	x = 0 S←S + 1 X_L←M(S) S←S + 1 X_H←M(S) x = 1 S←S + 1 X_L←M(S)	16/8																																	
PLY	x = 0 S←S + 1 Y_L←M(S) S←S + 1 Y_H←M(S) x = 1 S←S + 1 Y_L←M(S)	16/8																																	
PSH **(Note 17)**	M(S to S – i + 1)←A, B, X… S←S – i i: Number of bytes corresponding to register pushed on stack	16/8																																	
PUL **(Notes 18 and 22)**	A, B, X…←M(S + 1 to S + i) S←S + i i: Number of bytes corresponding to register restored from stack	16/8																																	
RLA **(Note 3)**	Rotate to the left by n bits m = 0 (n = 0 to 65535)  m = 1 (n = 0 to 255) 	16/8				31 07	5 + n	3																											
RMPA **(Note 19)**	m = 0 Repeat (B, A)←(B, A) + M(DT:X)✕ M(DT:Y) (Signed) X←X + 2 Y←Y + 2 i←i – 1 Until i = 0 m = 1 Repeat (B_L, A_L)←(B_L, A_L)+M(DT,X) M(DT,Y) (Signed) X←X + 1 Y←Y + 1 i←i – 1 Until i = 0 i: Numder of repetitions (0 to 255)	16/8																																	

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																								D5	4	1																						•	•	•	N	•	•	•	•	•	Z	•
																								F5	4	1																						•	•	•	N	•	•	•	•	•	Z	•
																								A8	11 + 2i1	2 + i2																						•	•	•	•	•	•	•	•	•	•	•
																								67	13 + 3 i1	2																						When the contents of PS is restored, this becomes the value. In the other cases, nothing changes.										
																																														•	•	•	•	•	•	•	•	•	•	•
																																												31 5A	5 + 14 imm	3	•	•	•	N	V	•	•	•	•	Z	C

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
ROL **(Note 1)**	Rotate to the left by 1 bit m = 0 Acc or M16: ←b_{15}…b_0←C← m = 1 Acc_L or M8: ←b_7…b_0←C←	16/8							13 81 13	1 2	1 2	21 1A	7	3	21 1B	8	3																			
ROL #n **(Note 4)**	Rotate to the left by n bits (n = 0 to 15) m = 0 A: ←b_{15}…b_0←C← m = 1 A_L: ←b_7…b_0←C←	16/8							C1 60 + imm	6 + imm	2																									
ROLD #n **(Note 4)**	Rotate to the left by n bits (n = 0 to 31) E: ←b_{31}…b_0←C←	32							D1 60 + imm	8 + imm	2																									
ROR **(Note 1)**	Rotate to the right by 1 bit m = 0 Acc or M16: →C→b_{15}…b_0→ m = 1 Acc_L or M8: →C→b_7…b_0→	16/8							53 81 53	1 2	1 2	21 3A	7	3	21 3B	8	3																			
ROR #n **(Note 4)**	Rotate to the right by n bits (n = 0 to 15) m = 0 A: →C→b_{15}…b_0→ m = 1 A_L: →C→b_7…b_0→	16/8							C1 20 + imm	6 + imm	2																									
RORD #n **(Note 4)**	Rotate to the right by n bits (n = 0 to 31) E: →b_{31}…b_0→C→	32							D1 20 + imm	8 + imm	2																									

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
21 1E	7	4	21 1F	8	4																																											•	•	•	N	•	•	•	•	•	Z	C
																																																•	•	•	N	•	•	•	•	•	Z	C
																																																•	•	•	N	•	•	•	•	•	Z	C
21 3E	7	4	21 3F	8	4																																											•	•	•	N	•	•	•	•	•	Z	C
																																																•	•	•	N	•	•	•	•	•	Z	C
																																																•	•	•	N	•	•	•	•	•	Z	C

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
RTI	S←S + 1 PS_L←M(S) S←S + 1 PS_H←M(S) S←S + 1 PC_L←M(S) S←S + 1 PC_H←M(S) S←S + 1 PG←M(S)	–	F1	12	1																														
RTL	S←S + 1 PC_L←M(S) S←S + 1 PC_H←M(S) S←S + 1 PG←M(S)	–	94	10	1																														
RTLD n **(Notes 11 and 12)**	S←S + 1 $DPRn_L$←M(S) S←S + 1 $DPRn_H$←M(S) S←S + 1 PC_L←M(S) S←S + 1 PC_H←M(S) S←S + 1 PG←M(S). (n = 0 to 3. Multiple DPRs can be specified.)	16																																	
RTS	S←S + 1 PC_L←M(S) S←S + 1 PC_H←M(S)	–	84	7	1																														
RTSD n **(Notes 11 and 12)**	S←S + 1 $DPRn_L$←M(S) S←S + 1 $DPRn_H$←M(S) S←S + 1 PC_L←M(S) S←S + 1 PC_H←M(S), (n = 0 to 3. Multiple DPRs can be specified.)	16																																	
SBC **(Notes 1 and 2)**	Acc←Acc – M – $\overline{C}$	16/8				31 A7 B1 A7	3 3	3 3				21 AA A1 AA	5 7	3 3	21 AB A1 AB	6 8	3 3				21 A0 A1 A0	7 9	3 3	21 A1 A1 A1	8 10	3 3	21 A8 A1 A8	8 10	3 3	21 A2 A1 A2	9 11	3 3	21 A9 A1 A9	10 12	3 3
SBCB **(Note 1)**	Acc_L←Acc_L – IMM8 – $\overline{C}$	8				31 1B B1 1B	3 3	3 3																											
SBCD	E←E – M32 – $\overline{C}$	32				31 1D	4	6				21 BA	7	3	21 BB	8	3				21 B0	9	3	21 B1	10	3	21 B8	10	3	21 B2	11	3	21 B9	12	3
SEC	C←1	–	04	1	1																														
SEI	I←1	–	05	4	1																														

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																																Value restored from stack										
																																																•	•	•	•	•	•	•	•	•	•	•
																								77 ?C	15	2																						•	•	•	•	•	•	•	•	•	•	•
																								77 ?C	12 + 3 i	2																																
																																														•	•	•	•	•	•	•	•	•	•	•
																								77 ?8	14	2																						•	•	•	•	•	•	•	•	•	•	•
																								77 ?8	11 + 3 i	2																																
21 AE	5	4	21 AF	6	4	21 A6	6	4	21 AC	6	5	21 AD	7	5																						21 A3	6	3	21 A4	9	3							•	•	•	N	V	•	•	•	•	Z	C
A1 AE	7	4	A1 AF	8	4	A1 A6	8	4	A1 AC	8	5	A1 AD	9	5																						A1 A3	8	3	A1 A4	11	3																	
																																																•	•	•	N	V	•	•	•	•	Z	C
21 BE	7	4	21 BF	8	4	21 B6	8	4	21 BC	8	5	21 BD	9	5																						21 B3	8	3	21 B4	11	3							•	•	•	N	V	•	•	•	•	Z	C
																																																•	•	•	•	•	•	•	•	•	•	1
																																																•	•	•	•	•	•	•	•	1	•	•

Symbol	Function	Operation length (Bit)	Addressing Modes																																
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
SEM	m←1	–	25	3	1																														
SEP	PSL(bit n)←1 (n = 0, 1, 3 to 7. Multiple bits can be specified.)	–				99	3	2																											
STA **(Note 1)**	M←Acc	16/8										DA	4	2	DB	5	2				11 D0	7	3	11 D1	8	3	D8	7	2	11 D2	9	3	D9	9	2
												81 DA	5	3	81 DB	6	3				91 D0	7	3	91 D1	8	3	81 D8	8	3	91 D2	9	3	81 D9	10	3
STAB **(Note 1)**	M8←Acc$_L$	8										CA	4	2	CB	5	2				11 C0	7	3	11 C1	8	3	C8	7	2	11 C2	9	3	C9	9	2
												81 CA	5	3	81 CB	6	3				91 C0	7	3	91 C1	8	3	81 C8	8	3	91 C2	9	3	81 C9	10	3
STAD	M32←E	32										EA	6	2	EB	7	2				11 E0	9	3	11 E1	10	3	E8	9	2	11 E2	11	3	E9	11	2
STP	Oscillation stopped	–	31 30	–	2																														
STX	M←X	16/8										E2	4	2				41 F5	6	3															
STY	M←Y	16/8										F2	4	2	41 FB	6	3																		
SUB **(Notes 1 and 2)**	Acc←Acc – M	16/8				36	1	2				3A	3	2	3B	4	2				11 30	6	3	11 31	7	3	11 38	7	3	11 32	8	3	11 39	9	3
						81 36	2	3				81 3A	4	3	81 3B	5	3				91 30	6	3	91 31	7	3	91 38	7	3	91 32	8	3	91 39	9	3
SUBB **(Note 1)**	Acc$_L$←Acc$_L$ – IMM8	8				39	1	2																											
						81 39	2	3																											
SUBD	E←E – M32	32				3D	3	5				AA	6	2	AB	7	2				11 A0	9	3	11 A1	10	3	11 A8	10	3	11 A2	11	3	11 A9	12	3
SUBM **(Note 3)**	M←M – IMM	16/8										51 13	7	4																					
SUBMB	M8←M8 – IMM8	8										51 12	7	4																					
SUBMD	M32←M32 – IMM32	32										51 93	10	7																					

Addressing Modes																																																Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																																•	•	•	•	•	1	•	•	•	•	•
																																																•	•	•	Specified flag becomes “1” **(Note 21)**.							
DE	4	3	DF	5	3	11 D6	6	4	DC	5	4	DD	6	4																						11 D3	6	3	11 D4	9	3							•	•	•	•	•	•	•	•	•	•	•
81 DE	5	4	81 DF	6	4	91 D6	6	4	81 DC	6	5	81 DD	7	5																						91 D3	6	3	91 D4	9	3																	
CE	4	3	CF	5	3	11 C6	6	4	CC	5	4	CD	6	4																						11 C3	6	3	11 C4	9	3							•	•	•	•	•	•	•	•	•	•	•
81 CE	5	4	81 CF	6	4	91 C6	6	4	81 CC	6	5	81 CD	7	5																						91 C3	6	3	91 C4	9	3																	
EE	6	3	EF	7	3	11 E6	8	4	EC	7	4	ED	8	4																						11 E3	8	3	11 E4	11	3							•	•	•	•	•	•	•	•	•	•	•
																																																•	•	•	•	•	•	•	•	•	•	•
E7	4	3																																														•	•	•	•	•	•	•	•	•	•	•
F7	4	3																																														•	•	•	•	•	•	•	•	•	•	•
3E	3	3	3F	4	3	11 36	5	4	11 3C	5	5	11 3D	6	5																						11 33	5	3	11 34	8	3							•	•	•	N	V	•	•	•	•	Z	C
81 3E	4	4	81 3F	5	4	91 36	5	4	91 3C	5	5	91 3D	6	5																						91 33	5	3	91 34	8	3																	
																																																•	•	•	N	V	•	•	•	•	Z	C
AE	6	3	AF	7	3	11 A6	8	4	11 AC	8	5	11 AD	9	5																						11 A3	8	3	11 A4	11	3							•	•	•	N	V	•	•	•	•	Z	C
51 17	7	5																																														•	•	•	N	V	•	•	•	•	Z	C
51 16	7	5																																														•	•	•	N	V	•	•	•	•	Z	C
51 97	10	8																																														•	•	•	N	V	•	•	•	•	Z	C

Symbol	Function	Operation length (Bit)	Addressing Modes																																			
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
SUBS	S←S – IMM8	16				31 0B	2	3																														
SUBX **(Note 4)**	X←X – IMM (IMM = 0 to 31)	16/8				01 40 + imm	2	2																														
SUBY **(Note 4)**	Y←Y – IMM (IMM = 0 to 31)	16/8				01 60 + imm	2	2																														
TAD n **(Note 20)**	DPRn←A (n = 0 to 3)	16	31 n2	3	2																																	
TAS	S←A	16	31 82	2	2																																	
TAX	X←A	16/8	C4	1	1																																	
TAY	Y←A	16/8	D4	1	1																																	
TBD n **(Note 20)**	DPRn←B (n = 0 to 3)	16	B1 n2	3	2																																	
TBS	S←B	16	B1 82	2	2																																	
TBX	X←B	16/8	81 C4	2	2																																	
TBY	Y←B	16/8	81 D4	2	2																																	
TDA n **(Note 20)**	A←DPRn (n = 0 to 3)	16/8	31 40 + n2	2	2																																	
TDB n **(Note 20)**	B←DPRn (n = 0 to 3)	16/8	B1 40 + n2	2	2																																	
TDS	S←DPR0	16	31 73	2	2																																	

Addressing Modes																																																		
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#

Processor Status register										
10	9	8	7	6	5	4	3	2	1	0
IPL			N	V	m	x	D	I	Z	C
•	•	•	N	V	•	•	•	•	Z	C
•	•	•	N	V	•	•	•	•	Z	C
•	•	•	N	V	•	•	•	•	Z	C
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Operation length (Bit)	Addressing Modes																																			
			IMP			IMM			A			DIR			DIR, X			DIR, Y			(DIR)			(DIR, X)			(DIR), Y			L(DIR)			L(DIR), Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
TSA	A←S	16/8	31 92	2	2																														
TSB	B←S	16/8	B1 92	2	2																														
TSD	DPR0←S	16	31 70	4	2																														
TSX	X←S	16/8	31 F2	2	2																														
TXA	A←X	16/8	A4	1	1																														
TXB	B←X	16/8	81 A4	2	2																														
TXS	S←X	16/8	31 E2	2	2																														
TXY	Y←X	16/8	31 C2	2	2																														
TYA	A←Y	16/8	B4	1	1																														
TYB	B←Y	16/8	81 B4	2	2																														
TYX	X←Y	16/8	31 D2	2	2																														
WIT	CPU clock stopped	–	31 10	–	2																														
XAB	A⇆B	16/8	55	2	1																														

Addressing Modes																																																			Processor Status register										
ABS			ABS, X			ABS, Y			ABL			ABL, X			(ABS)			L(ABS)			(ABS, X)			STK			REL			DIR, b, R			ABS, b, R			SR			(SR), Y			BLK			MAA			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	•	•	•	•	•	•	•	•
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	•	•	•	•	•	•	•	•
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	N	•	•	•	•	•	Z	•
																																																			•	•	•	•	•	•	•	•	•	•	•
																																																			•	•	•	N	•	•	•	•	•	Z	•

Notes for machine instructions table

The table lists the minimum number of instruction cycles for each instruction. The number of cycle is changed by the following condition.

- The value of the low-order bytes of DPR ($DPRn_L$)
  The number of cycle of the addressing mode related with DPRn (n = 0 to 3) is applied when DPRn = 0. When DPRn ≠ 0, add 1 to the number of cycles.
- The number of bytes of instruction which fetched into the instruction queue buffer
- The address at read and write of memory (either even or odd)
- When the external area accessed in BYTE = Vcc level (at external data bus width 8 bits)
- The number of wait

Note 1. The op code at the upper row is used for accumulator A, and the op code at the lower row is used for accumulator B.

Note 2. When handing 16-bit data with flag m = 0 in the immediate addressing mode, add 1 to the numder of bytes.

Note 3. When handing 16-bit data with flag m = 0, add 1 to the numder of bytes.

Note 4. Imm is the immediate value specified with an operand (imm = 0–31).

Note 5. The op code at the upper row is used for branching in the range of –128 to +127, and the op code at the lower row is used for branching in the range of –32768 to +32767.

Note 6. The BRK instruction is a instruction for debugger; it cannot be used.

Note 7. Any value from 0 through 15 is placed in an “n.”

Note 8. When handling 16-bit data with flag x = 0 in the immediate addressing mode, add 1 to the numder of bytes.

Note 9. The number of cycles is the case of the 16-bit ÷ 8-bit operation. In the case of the 32-bit ÷ 16-bit operation, add 8 to the number of cycles.

Note 10. When a zero division interrupt occurs, the number of cycles is 16 cycles. It is regardless of the data length.

Note 11. When placing a value in any of DPRs, the op code at the upper row is applied. When placing values to multiple DPRs, the op code at the lower row is applied. The letter “i” represents the number of DPRn specified: 1 to 4.

Note 12. A “?” indicates to the value of 4 bits which the bit corressing to the specified DPRn becomes “1.”

Note 13. When the source is in the immediate addressing mode and flag m = 0, add n (n = 0 to 15) to the number of bytes.

Note 14. The number of cycles of the case of the 8-bit × 8-bit operation. In the case of the 16-bit × 16-bit operation, add 4 to the number of cycles.

Note 15. The number of cycles is the case where the number of bytes to be transferred (i) is even. When the number of bytes to be transferred (i) is odd, the number is calculated as;

$$5 \times i + 10$$

Note 16. The number of cycles is the case where the number of bytes to be transferred (i) is even. When the number of bytes to be transferred (i) is odd, the number is calculated as;

$$5 \times i + 14$$

Note that it is 10 cycles in the case of 1-byte thanster.

Note 17. i_1 is the number of registers to be stored among A, B, X, Y, DPR0, and PS. i_2 is the number of registers to be stored between DT and PG.

Note 18. Letter "i_1" indicates the number of registers to be restored.

Note 19. The number of cycles is applied when flag m = "1." When flag m="0," the number is calculated as;

$$18 \times imm + 5$$

Note 20. Any value from 0 through 3 is placed in an "n" in op code."

Appendix 7. Countermeasure against noise

General countermeasure examples against noise are described below. Although the effect of these countermeasure depends on each system.
The user shall modify them according to the actual application and test them.

1. Short wiring length

The wiring on a printed circuit board may function as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less possibility of noise insertion into the microcomputer.

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

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

Reason: If noise is input to the $\overline{\text{RESET}}$ pin, the microcomputer restarts operation before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

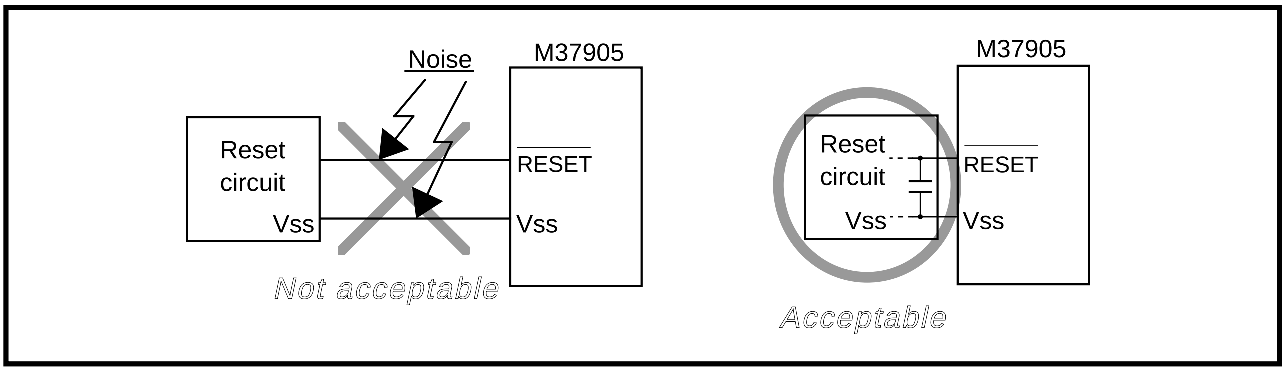


Fig. 2 Wiring for $\overline{\text{RESET}}$ pin

(2) Wiring for clock input/output pins

- Make the length of wiring connected to the clock input/output pins as short as possible.
- Make the length of wiring between the grounding lead of the capacitor, which is connected to the oscillator, and the Vss pin of the microcomputer, as short as possible (within 20 mm).
- Separate the Vss pattern for oscillation from all other Vss patterns. (See Figure 10.)

Reason: The microcomputer's operation synchronizes with a clock generated by the oscillation circuit.
If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a malfunction or a program runaway.
Also, if the noise causes a potential difference between the Vss level of the microcomputer and the Vss level of an oscillator, the correct clock will not be input in the microcomputer.

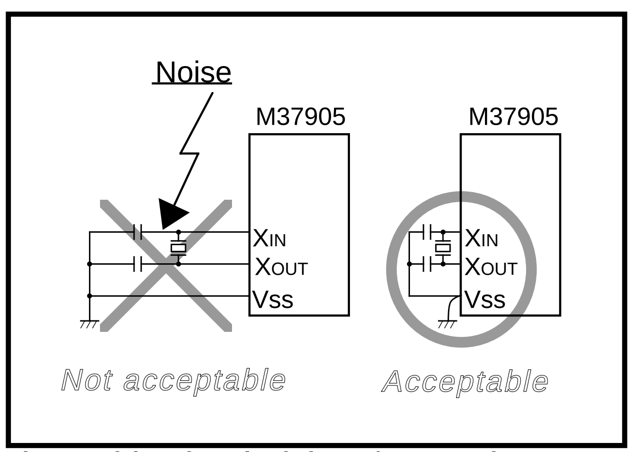


Fig. 3 Wiring for clock input/output pins

(3) Wiring for MD0 and MD1 pins

Connect MD0 and MD1 pins to the Vss pin (or Vcc pin) with the shortest possible wiring.

Reason: The processor mode of the microcomputer is influenced by a potential at the MD0 and MD1 pins when the MD0 and MD1 pins and the Vss pin (or Vcc pin) are connected.
If the noise causes a potential difference between the MD0 and MD1 pins and the Vss pin (or Vcc pin), the processor mode may become unstable. This may cause a microcomputer malfunction or a program runaway.

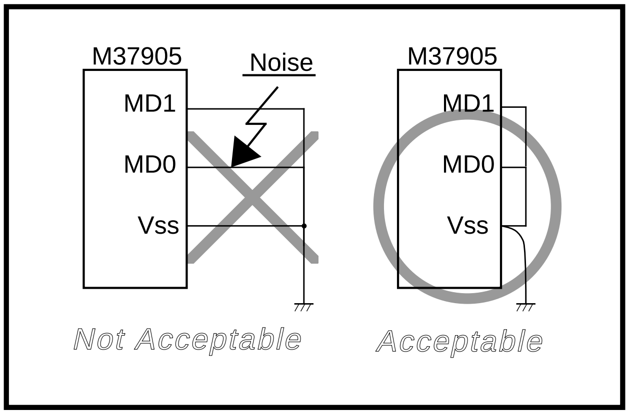


Fig. 4 Wiring for MD0 and MD1 pins

2. Connection of bypass capacitor between Vss and Vcc lines

Connect an approximate 0.1 μF bypass capacitor as follows:

- Connect a bypass capacitor between the Vss and Vcc pins, at equal lengths.
- The wiring connecting the bypass capacitor between the Vss and Vcc pins should be as short as possible.
- Use thicker wiring for the Vss and Vcc lines than that for the other signal lines.

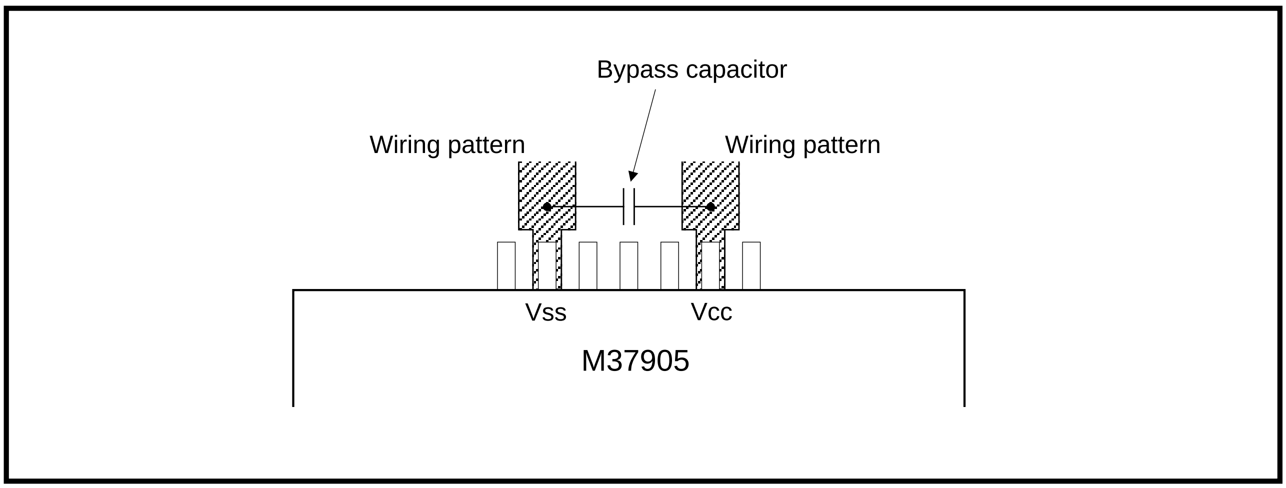


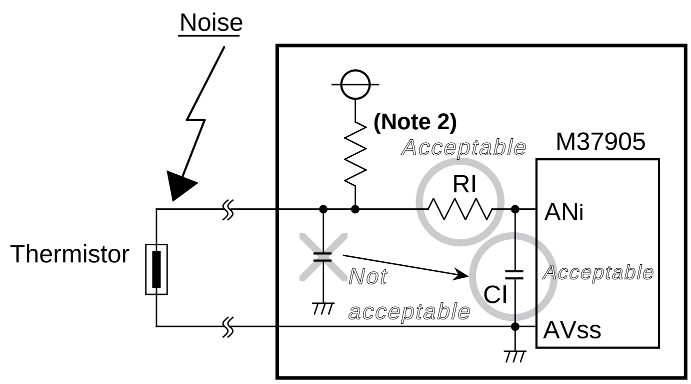
Fig. 5 Bypass capacitor between Vss and Vcc lines

3. Wiring for analog input pins, analog power source pins, etc.

(1) Processing for analog input pins

- Connect a resistor to the analog signal line, which is connected to an analog input pin, in series. Additionally, connect the resistor to the microcomputer as close as possible.
- Connect a capacitor between the analog input pin and the AVss pin, as close to the AVss pin as possible.

Reason: A signal which is input to the analog input pin is usually an output signal from a sensor. The sensor, which detects changes in status, is installed far from the microcomputer's printed circuit board. Therefore, this long wiring between them becomes an antenna which picks up noise and feeds it into the microcomputer's analog input pin.
If a capacitor between an analog input pin and the AVss pin is grounded far away from the AVss pin, noise on the GND line may enter the microcomputer through the capacitor.



Reference values
RI : Approximate 100 Ω to 1000 Ω
CI : Approximate 100 pF to 1000 pF

Notes 1: Design an external circuit for the ANi pin so that charge/discharge is available within 1 cycle of ϕ_{AD}.
2: This resistor and thermistor are used to divide resistance.

Fig. 6 Countermeasure example against noise for analog input pin using thermistor

(2) Processing for analog power source pins, etc.

- Use independent power sources for the Vcc, AVcc and V_{REF} pins.
- Insert capacitors between the AVcc and AVss pins, and between the V_{REF} and AVss pins.

Reasons: Prevents the A-D converter and D-A converter from noise on the Vcc line.

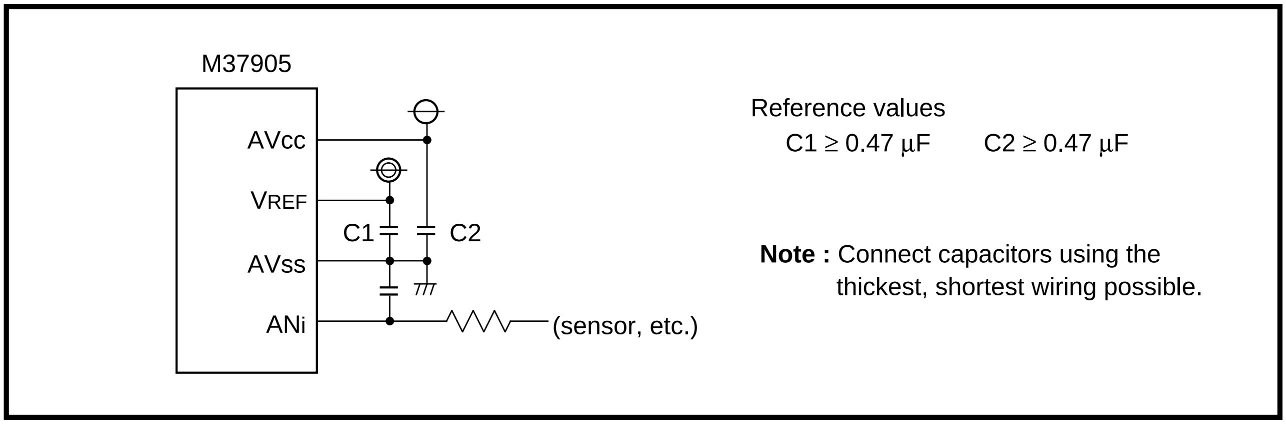


Fig. 7 Processing for analog power source pins, etc.

4. Oscillator protection

The oscillator, which generates the basic clock for the microcomputer operations, must be protected from the affect of other signals.

(1) Distance oscillator from signal lines with large current flows

Install the microcomputer, especially the oscillator, as far as possible from signal lines which handle currents larger than the microcomputer current value tolerance.

Reason: The microcomputer is used in systems which contain signal lines for controlling motors, LEDs, thermal heads, etc. Noise occurs due to mutual inductance when a large current flows through the signal lines.

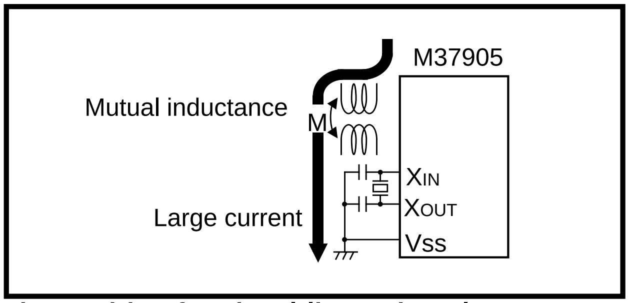


Fig. 8 Wiring for signal lines where large current flows

(2) Distance oscillator from signal lines with frequent potential level changes

- Install an oscillator and its wiring pattern away from signal lines where potential levels change frequently.
- Do not cross these signal lines over the clock-related or noise-sensitive signal lines.

Reason: Signal lines with frequently changing potential levels may affect other signal lines at a rising or falling edge. In particular, if the lines cross over a clock-related signal line, clock waveforms may be deformed, which causes a microcomputer malfunction or a program runaway.

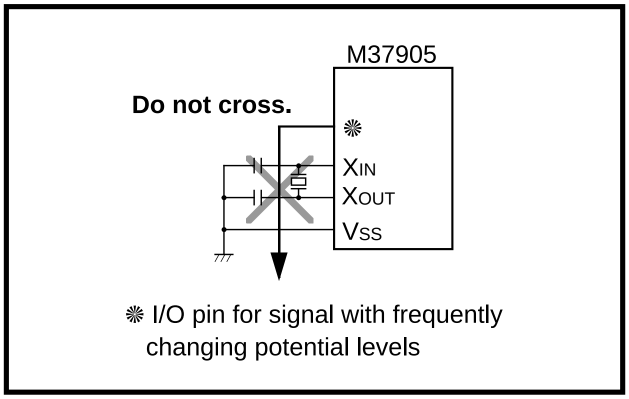


Fig. 9 Wiring for signal lines where potential levels frequently change

(3) Oscillator protection using Vss pattern

Print a Vss pattern on the bottom (soldering side) of a double-sided printed circuit board, under the oscillator mount position.

Connect the Vss pattern to the Vss pin of the microcomputer with the shortest possible wiring, separating it from other Vss patterns.

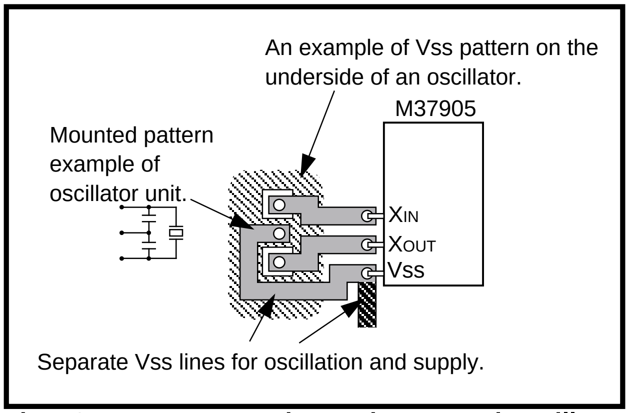


Fig. 10 Vss pattern underneath mounted oscillator

5. Setup for I/O ports
Setup I/O ports by hardware and software as follows:

<Hardware protection>
● Connect a resistor of 100 Ω or more to an I/O port in series.

<Software protection>
● Read the data of an input port several times to confirm that input levels are equal.
● Since the output data may reverse because of noise, rewrite data to the output port's Pi register periodically.
● Rewrite data to port Pi direction registers periodically.

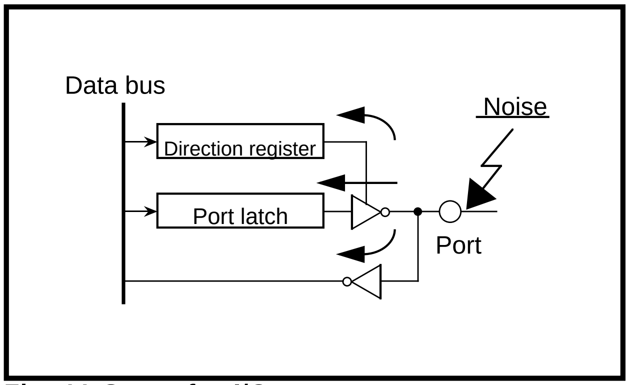


Fig. 11 Setup for I/O ports

6. Reinforcement of the power source line
● For the Vss and Vcc lines, use thicker wiring than that of other signal lines.
● When using a multilayer printed circuit board, the Vss and Vcc patterns must each be one of the middle layers.
● The following is necessary for double-sided printed circuit boards:
•On one side, the microcomputer is installed at the center, and the Vss line is looped or meshed around it. The vacant area is filled with the Vss line.
•On the opposite side, the Vcc line is wired the same as the Vss line.

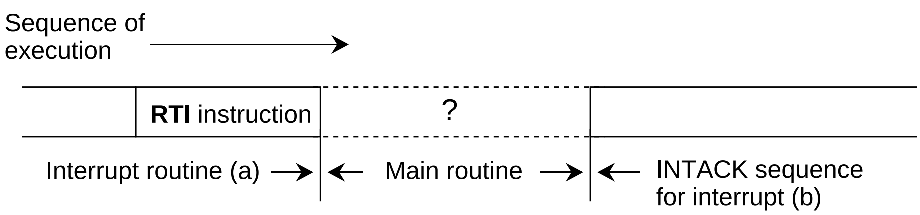
Appendix 8. 7905 Group Q & A

Information which may be helpful in fully utilizing the 7905 Group is provided in Q & A format.
In Q & A, as a rule, one question and its answer are summarized within one page. The upper box on each page is a question, and a box below the question is its answer. (If a question or an answer extends to two or more pages, there is a page number at the lower right corner.)
At the upper right corner of each page, the main function related to the contents of description in that page is listed.

Interrupts

Q

If an interrupt request (b) occurs while an interrupt routine (a) is executed, is it true that the main routine is not executed at all after the execution of the interrupt routine (a) is completed until the execution of the INTACK sequence for the next interrupt (b) starts?



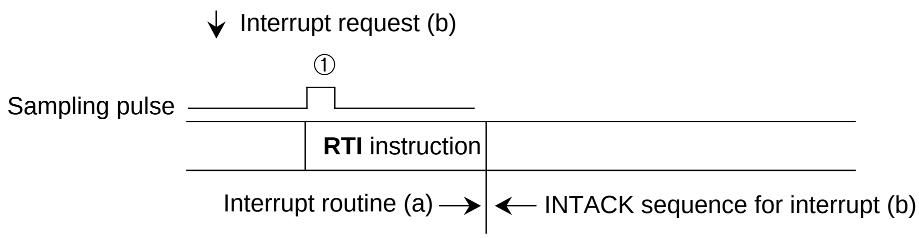
Conditions:

- Flag I is cleared to “0” by executing the **RTI** instruction.
- The interrupt priority level of interrupt (b) is higher than IPL of the main routine.
- The interrupt priority detection time = 2 cycles of f_{sys}.

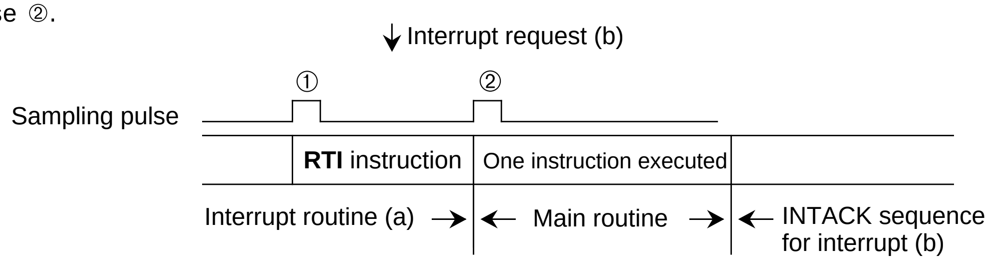
A

An interrupt request is sampled by a sampling pulse generated synchronously with the CPU’s op-code fetch cycle.

(1) If the next interrupt request (b) occurs before sampling pulse ① for the **RTI** instruction is generated, the microcomputer executes the INTACK sequence for (b) without executing the main routine. (No instruction of the main routine is executed.) It is because that sampling is completed while executing the **RTI** instruction.



(2) If the next interrupt request (b) occurs immediately after sampling pulse ① is generated, the microcomputer executes one instruction of the main routine before executing the INTACK sequence for (b). It is because that the interrupt request is sampled by the next sampling pulse ②.



Q

Suppose that there is a routine which should not accept a certain interrupt request. (This routine can accept any of the other interrupt request.)
Although the interrupt priority level select bits for a certain interrupt are set to “000_2” (in other words, although this interrupt is set to be disabled), this interrupt request is actually accepted immediately after the change of the priority level. Why did this occur, and what should I do about it?

```
                                       :
Interrupt request is    MOVMB  XXXIC, #00H ; Writes "0002" to the interrupt priority level select bits.
accepted in this    →                      ; Clears the interrupt request bit to "0."
interval                LDA     A,DATA     ; Instruction at the beginning of the routine which
                                             should not accept a certain interrupt request.
                                       :   ;
```

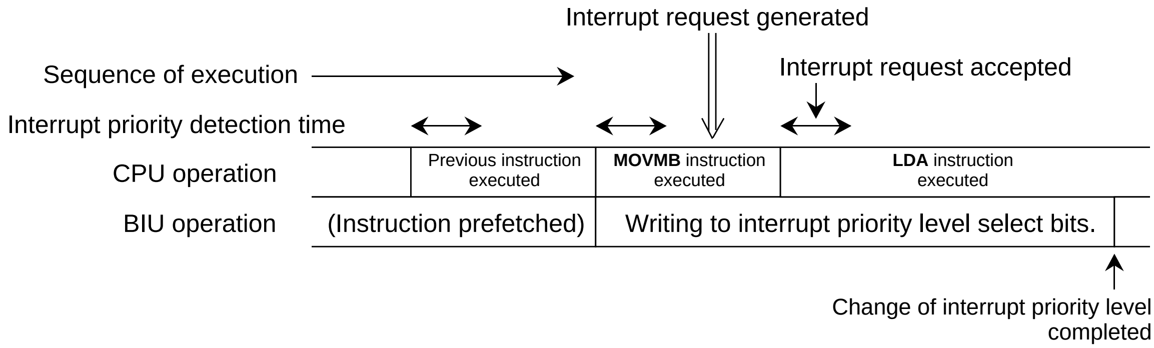
A

As for the change of the interrupt priority level, if the following are met, the microcomputer may pretend to accept an interrupt request immediately after this interrupt is set to be disabled:

- The next instruction (in the above example, it is the **LDA** instruction) is already stored into a instruction queue buffer of the BIU.
- Requirements for accepting the interrupt request which should not be accepted are satisfied immediately before the next instruction in the instruction queue buffer is executed.

When writing to a memory or an I/O, the CPU passes an address and data to the BIU. Then, the CPU executes the next instruction in the instruction queue buffer while the BIU is writing data into the actual address. Detection of the interrupt priority level is performed at the beginning of each instruction.

In the above case, the CPU executes the next instruction before the BIU completes the change of the interrupt priority level. Therefore, in the detection of the interrupt priority level performed synchronously with the execution of the next instruction, actually, the interrupt priority level before the change is used to detection, and its interrupt request is accepted.



(1/2)

A

To prevent this problem, make sure that the routine which should not accept a certain interrupt request will be executed after the change of the interrupt priority level (IPL) has been completed. (This is to be made by software.)
The following is a sample program.

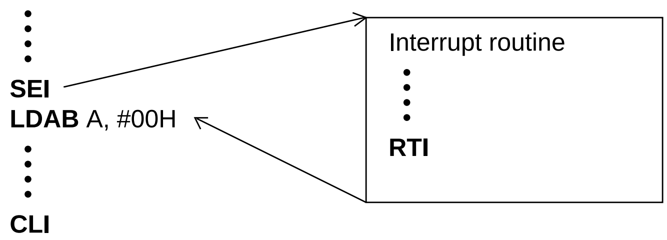
[Sample program]
After writing "000_2" to the interrupt priority level select bits, the instruction queue buffer is filled with several **NOP** instructions to make the next instruction not to be executed before this writing is completed.

```
  :
MOVMB XXXIC, #00H    ; Writes "0002" to the interrupt priority level select bits.
NOP                  ; Inserts ten NOP instructions.
  :
NOP                  ;
LDA     A,DATA       ; Instruction at the beginning of the routine that should not accept a
  :                    certain interrupt request
```

(2/2)

Q

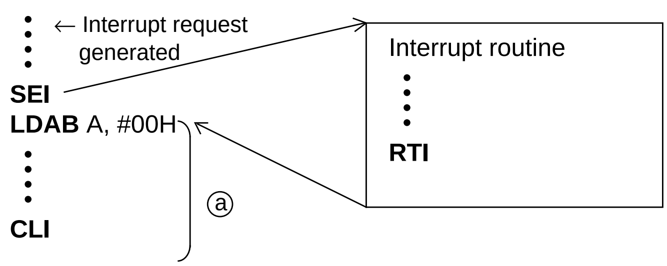
After execution of the **SEI** instruction, a branch is made in an interrupt routin.
Why did this occur?



A

When an interrupt request is generated before the **SEI** instruction is executed, this interrupt request may be accepted immediately before the execution of the **SEI** instruction. (This acceptance occurs depending on the timing when that interrupt request occurs.) In this case, a branch to the interrupt routine is made immediately after execution of the **SEI** instruction.
Accordingly, the interrupt routine which is executed immediately after the **SEI** instruction is due to an interrupt request generated <u>before execution of the **SEI** instruction</u>. Note that, in the routine (ⓐ) which should not accept the interrupt request, the following occur. (This routine follows the **SEI** instruction.):

- No interrupt request is accepted.
- No branch to the interrupt routine is made.



Note: "Interrupt" described here means "maskable interrupt" which can be disabled by the **SEI** instruction. (Refer to section **"6.2 Interruput source."**)

Interrupts

Q

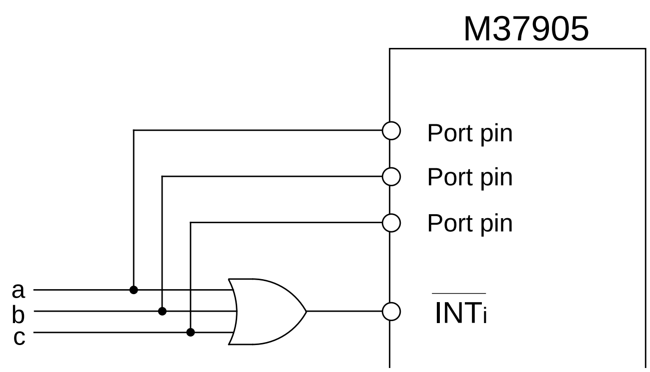
(1) Which timing of clock ϕ_1 is the external interrupts (input signals to the $\overline{INT_i}$ pin) detected?
(2) When external interrupt input ($\overline{INT_i}$) pins are not enough, what should I do?

A

(1) In both of the edge sense and level sense, an external interrupt request occurs when the input signal to the $\overline{INT_i}$ pin changes its level. This is independent of clock ϕ_1.
In the edge sense, also, the interrupt request bit is set to "1" at this time.

(2) There are two methods: one uses external interrupt's level sense, and the other uses the timer's event counter mode.

① **Method using external interrupt's level sense**
As for hardware, input a logical sum of multiple interrupt signals (e.g., 'a', 'b', and 'c') to the $\overline{INT_i}$ pin, and input each signal to each corresponding port pin.
As for software, check the port pin's input levels in the $\overline{INT_i}$ interrupt routine in order to detect which signal ('a', 'b', or 'c') was input.



② **Method using timer's event counter mode**
As for hardware, input an interrupt signal to the TAi_{IN} pin or TBi_{IN} pin.
As for software, set the timer's operating mode to the event counter mode. Then, set a value of "0000_{16}" into the timer register and select the valid edge.
A timer's interrupt request occurs when an interrupt signal (selected valid edge) is input.

Watchdog timer

Q

In detection of a program runaway with usage of the watchdog timer, if the same value as that at the reset vector address is set to the watchdog timer interrupt's vector address, not performing software reset, how does it occur?
When a branch is made to the branch destination address for reset within the watchdog timer interrupt routine, how does it occur?

A

The CPU registers and the SFR are not initialized in the above-mentioned way. Accordingly, the user must initialize all of them by software.
Note that the processor interrupt priority level (IPL) retains "7" and is not initialized. Consequently, all interrupt requests cannot be accepted.
When rewriting the IPL by software, be sure to save the 16-bit immediate value to the stack area, and then restore that 16-bit immediate value to all bits of the processor status register (PS).

When a program runaway occurs, we recommend to perform software reset in order to initialize the microcomputer.

Appendix 9. M37905M4C-XXXFP electrical characteristics

The electrical characteristics of the M37905M4C-XXXFP are described below.
For the electrical characteristics, be sure to refer to the latest datasheet.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Ratings	Unit
V_{CC}	Power source voltage	–0.3 to 6.5	V
AV_{CC}	Analog power source voltage	–0.3 to 6.5	V
V_I	Input voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_1$–$P5_3$, $P5_5$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_3$, $P4OUT_{CUT}$, $P6OUT_{CUT}$, V_{CONT}, V_{REF}, X_{IN}, $\overline{RESET}$, MD0, MD1	–0.3 to V_{CC}+0.3	V
V_O	Output voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_1$–$P5_3$, $P5_5$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_3$, X_{OUT}	–0.3 to V_{CC}+0.3	V
P_d	Power dissipation	300	mW
T_{opr}	Operating ambient temperature	–20 to 85	°C
T_{stg}	Storage temerature	–40 to 150	°C

Appendix 9. M37905M4C-XXXFP electrical characteristics

RECOMMENDED OPERATING CONDITIONS (Vcc = 5 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Power source voltage	4.5	5.0	5.5	V
AV_{CC}	Analog power source voltage		V_{CC}		V
V_{SS}	Power source voltage		0		V
AV_{SS}	Analog power source voltage		0		V
V_{IH}	High-level input voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_1$–$P5_3$, $P5_5$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_3$, $\overline{P4OUT_{CUT}}$, $\overline{P6OUT_{CUT}}$, X_{IN}, $\overline{RESET}$, MD0, MD1	0.8 Vcc		Vcc	V
V_{IL}	Low-level input voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_1$–$P5_3$, $P5_5$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_3$, $\overline{P4OUT_{CUT}}$, $\overline{P6OUT_{CUT}}$, X_{IN}, $\overline{RESET}$, MD0, MD1	0		0.2 V_{CC}	V
$I_{OH(peak)}$	High-level peak output current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_1$–$P5_3$, $P5_5$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_3$			–10	mA
$I_{OH(avg)}$	High-level average output current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_7$, $P5_1$–$P5_3$, $P5_5$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_3$			–5	mA
$I_{OL(peak)}$	Low-level peak output current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_1$–$P5_3$, $P5_5$–$P5_7$, $P7_0$–$P7_7$, $P8_0$–$P8_3$			10	mA
$I_{OL(peak)}$	Low-level peak output current $P4_0$–$P4_7$, $P6_0$–$P6_7$			20	mA
$I_{OL(avg)}$	Low-level average output current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_1$–$P5_3$, $P5_5$–$P5_7$, $P7_0$–$P7_7$, $P8_0$–$P8_3$			5	mA
$I_{OL(avg)}$	Low-level average output current $P4_0$–$P4_7$, $P6_0$–$P6_7$			15	mA
$f(X_{IN})$	External clock input frequency **(Note 1)**			20	MHz
$f(f_{sys})$	System clock frequency			20	MHz

Notes 1: When using the PLL frequency multiplier, be sure that $f(f_{sys})$ = 20 MHz or less.
2: The average output current is the average value of an interval of 100 ms.
3: The sum of $I_{OL(peak)}$ must be 110 mA or less, the sum of $I_{OH(peak)}$ must be 80 mA or less.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5 V, V_{SS} = 0 V, Ta = –20 to 85 °C, f(f_{sys}) = 20 MHz, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage	P1$_0$–P1$_7$, P2$_0$–P2$_7$, P4$_0$–P4$_7$, P5$_1$–P5$_3$, P5$_5$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_3$	I_{OH} = –10 mA	3			V
V_{OL}	Low-level output voltage	P1$_0$–P1$_7$, P2$_0$–P2$_7$, P4$_0$–P4$_7$, P5$_1$–P5$_3$, P5$_5$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_3$	I_{OL} = 10 mA			2	V
V_{T+} –V_{T-}	Hysteresis	TA0$_{IN}$–TA9$_{IN}$, TA0$_{OUT}$–TA9$_{OUT}$, TB0$_{IN}$–TB2$_{IN}$, $\overline{INT_0}$–$\overline{INT_7}$, $\overline{CTS_0}$, $\overline{CTS_1}$, $\overline{CTS_2}$, CLK$_0$, CLK$_1$, CLK$_2$, RxD$_0$, RxD$_1$, RxD$_2$, $\overline{RTP_{TRG0}}$, $\overline{RTP_{TRG1}}$, $\overline{P4OUT_{CUT}}$, $\overline{P6OUT_{CUT}}$		0.4		1	V
V_{T+} –V_{T-}	Hysteresis	$\overline{\text{RESET}}$		0.5		1.5	V
V_{T+} –V_{T-}	Hysteresis	X_{IN}		0.1		0.3	V
I_{IH}	High-level input current	P1$_0$–P1$_7$, P2$_0$–P2$_7$, P4$_0$–P4$_7$, P5$_1$–P5$_3$, P5$_5$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_3$, $\overline{P4OUT_{CUT}}$, $\overline{P6OUT_{CUT}}$, X_{IN}, $\overline{\text{RESET}}$, MD0, MD1	V_I = 5.0 V			5	μA
I_{IL}	Low-level input current	P1$_0$–P1$_7$, P2$_0$–P2$_7$, P4$_0$–P4$_7$, P5$_1$–P5$_3$, P5$_5$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_3$, $\overline{P4OUT_{CUT}}$, $\overline{P6OUT_{CUT}}$, X_{IN}, $\overline{\text{RESET}}$, MD0, MD1	V_I = 0 V			–5	μA
V_{RAM}	RAM hold voltage		When clock is inactive.	2			V
I_{CC}	Power source current	Output-only pins are open, and the other pins are connected to V_{SS} or V_{CC}. An external square-waveform clock is input. (Pin X_{OUT} is open.) The PLL frequency multiplier is inactive.	f(f_{sys}) = 20 MHz. CPU is active.		25	50	mA
			Ta = 25 °C when clock is inactive.			1	μA
			Ta = 85 °C when clock is inactive.			20	μA

A-D CONVERTER CHARACTERISTICS

(V_{CC} = AV_{CC} = 5 V ± 0.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
——	Resolution	V_{REF} = V_{CC}	A-D converter			10	Bits
			Comparator			$\frac{1}{256} V_{REF}$	V
——	Absolute accuracy	V_{REF} = V_{CC}	10-bit resolution mode			± 3	LSB
			8-bit resolution mode			± 2	LSB
			Comparater			± 40	mV
R_{LADDER}	Ladder resistance	V_{REF} = V_{CC}		5			kΩ
t_{CONV}	Conversion time	$f(f_{sys}) \leq 20$ MHz	10-bit resolution mode	5.9			μs
			8-bit resolution mode	2.45 **(Note)**			
			Comparater	0.7 **(Note)**			
V_{REF}	Reference voltage			2.7		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

Note: This is applied when A-D conversion freguency (ϕ_{AD}) = f_1 (ϕ).

D-A CONVERTER CHARACTERISTICS

(V_{CC} = 5 V, V_{SS} = AV_{SS} = 0 V, V_{REF} = 5 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
——	Resolution				8	Bits
——	Absolute accuracy				± 1.0	%
t_{su}	Set time				3	μs
R_O	Output resistance		2	3.5	4.5	kΩ
I_{VREF}	Reference power source input current	**(Note)**			3.2	mA

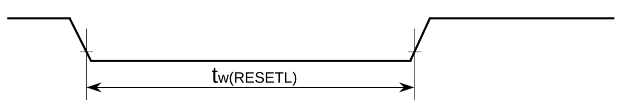
Note: The test conditions are as follows:
- One D-A converter is used.
- The D-A register value of the unused D-A converter is "00_{16}."
- The reference power source input current for the ladder resistance of the A-D converter is excluded.

RESET INPUT

Reset input timing requirements (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits Min.	Limits Typ.	Limits Max.	Unit
$t_{w(RESETL)}$	$\overline{RESET}$ input low-level pulse width	10			μs

$\overline{RESET}$ input



PERIPHERAL DEVICE INPUT/OUTPUT TIMING

(V_{CC} = 5 V±0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, $f(f_{sys})$ = 20 MHz, unless otherwise noted)

✻ For limits depending on $f(f_{sys})$, their calculation formulas are shown below. Also, the values at $f(f_{sys})$ = 20 MHz are shown in ( ).

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	80		ns
$t_{w(TAH)}$	TAiIN input high-level pulse width	40		ns
$t_{w(TAL)}$	TAiIN input low-level pulse width	40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	$f(f_{sys}) \le 20$ MHz	$\frac{16 \times 10^9}{f(f_{sys})}$ (800)		ns
$t_{w(TAH)}$	TAiIN input high-level pulse width	$f(f_{sys}) \le 20$ MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns
$t_{w(TAL)}$	TAiIN input low-level pulse width	$f(f_{sys}) \le 20$ MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns

Note : The TAiIN input cycle time requires 4 or more cycles of a count source. The TAiIN input high-level pulse width and the TAiIN input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at $f(f_{sys}) \le 20$ MHz.

Timer A input (External trigger input in one-shot pulse mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	$f(f_{sys}) \le 20$ MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns
$t_{w(TAH)}$	TAiIN input high-level pulse width		80		ns
$t_{w(TAL)}$	TAiIN input low-level pulse width		80		ns

Timer A input (External trigger input in pulse width modulation mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input high-level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low-level pulse width	80		ns

Timer A input (Up-down input and Count input in event counter mode)

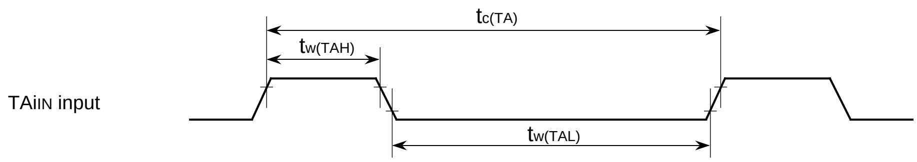
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input high-level pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input low-level pulse width	1000		ns
$t_{su(UP-T_{IN})}$	TAiOUT input setup time	400		ns
$t_{h(T_{IN}-UP)}$	TAiOUT input hold time	400		ns

Appendix 9. M37905M4C-XXXFP electrical characteristics

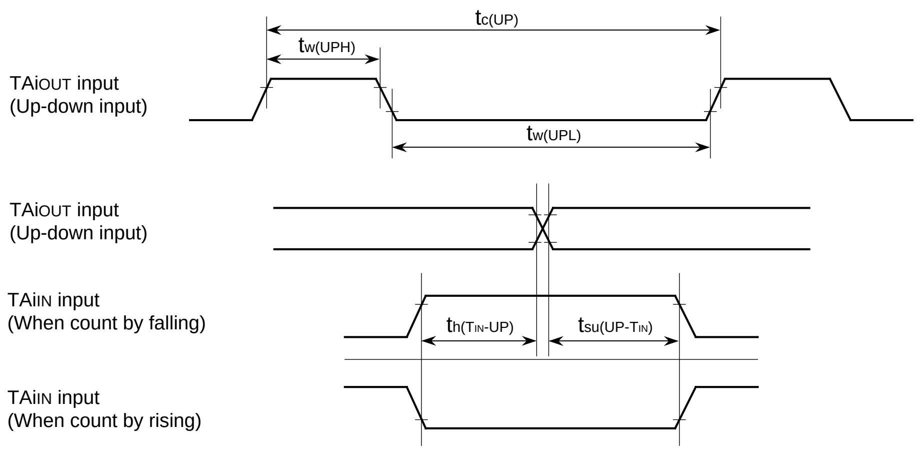
Timer A input (Two-phase pulse input in event counter mode) (j = 2 to 4, 7 to 9)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAj_{IN} input cycle time	800		ns
$t_{su(TAj_{IN}-TAj_{OUT})}$	TAj_{IN} input setup time	200		ns
$t_{su(TAj_{OUT}-TAj_{IN})}$	TAj_{OUT} input setup time	200		ns

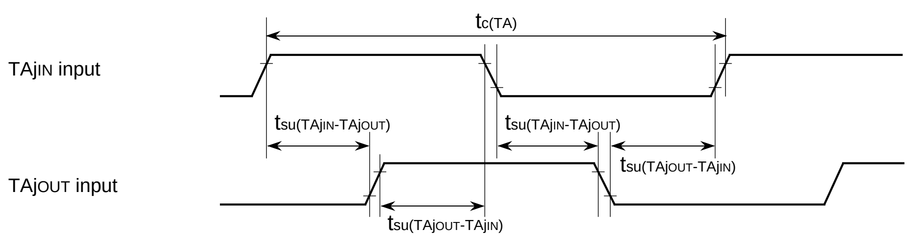
- Gating input in timer mode
- Count input in event counter mode
- External trigger input in one-shot pulse mode
- External trigger input in pulse width modulation mode



- Up-down and Count input in event counter mode



- Two-phase pulse input in event counter mode



Test conditions
- V_{CC} = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time (one edge count)	80		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width (one edge count)	40		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width (one edge count)	40		ns
$t_{c(TB)}$	TBi_{IN} input cycle time (both edge count)	160		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width (both edge count)	80		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width (both edge count)	80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	$f(f_{sys}) \leq 20$ MHz	$\frac{16 \times 10^9}{f(f_{sys})}$ (800)		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	$f(f_{sys}) \leq 20$ MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	$f(f_{sys}) \leq 20$ MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns

Note: The TBi_{IN} input cycle time requires 4 or more cycles of a count source. The TBi_{IN} input high-level pulse width and the TBi_{IN} input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at $f(f_{sys}) \leq 20$ MHz.

Timer B input (Pulse width measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	$f(f_{sys}) \leq 20$ MHz	$\frac{16 \times 10^9}{f(f_{sys})}$ (800)		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	$f(f_{sys}) \leq 20$ MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	$f(f_{sys}) \leq 20$ MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns

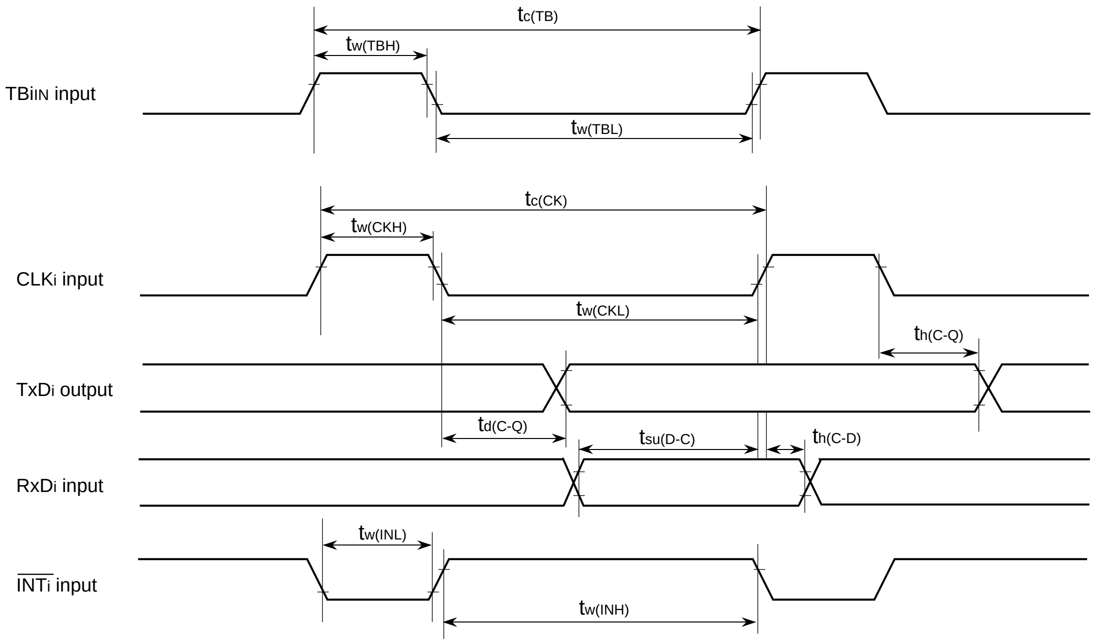
Note: The TBi_{IN} input cycle time requires 4 or more cycles of a count source. The TBi_{IN} input high-level pulse width and the TBi_{IN} input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at $f(f_{sys}) \leq 20$ MHz.

Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK_i input cycle time	200		ns
$t_{w(CKH)}$	CLK_i input high-level pulse width	100		ns
$t_{w(CKL)}$	CLK_i input low-level pulse width	100		ns
$t_{d(C-Q)}$	TxD_i output delay time		80	ns
$t_{h(C-Q)}$	TxD_i hold time	0		ns
$t_{su(D-C)}$	RxD_i input setup time	20		ns
$t_{h(C-D)}$	RxD_i input hold time	90		ns

External interrupt ($\overline{INT_i}$) input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		ns
$t_{w(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		ns



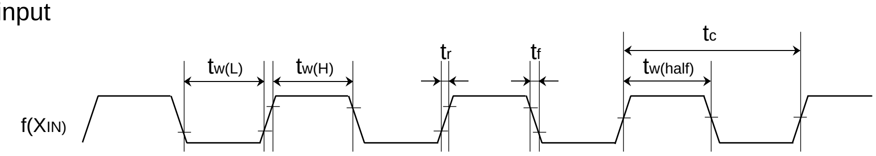
Test conditions
- V_{CC} = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 50 pF

External clock input

Timing Requirements (V_{CC} = 5 V±0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time	50		ns
$t_{w(half)}$	External clock input pulse width with half input-voltage	0.45 tc	0.55 tc	ns
$t_{w(H)}$	External clock input high-level pulse width	0.5 t_c – 8		ns
$t_{w(L)}$	External clock input low-level pulse width	0.5 t_c – 8		ns
t_r	External clock input rise time		8	ns
t_f	External clock input fall time		8	ns

External clock input



Test conditions
- Vcc = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V ($t_{w(H)}$, $t_{w(L)}$, t_r, t_f)
- Input timing voltage : 2.5 V (t_c, $t_{w(half)}$)

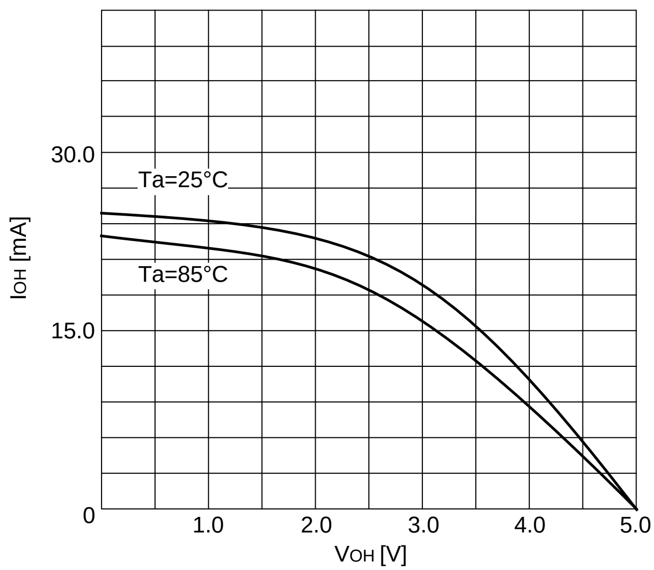
Appendix 10. M37905M4C-XXXFP standard characteristics

Standard characteristics described below are just examples of the M37905M4C-XXXFP's characteristics and are not guaranteed. For each parameter's limits, refer to sections **"Appendix 9. M37905M4C-XXXFP electrical characteristics."**

1. Programmable I/O port (CMOS output) standard characteristics: P1, P2, P5, P7, P8

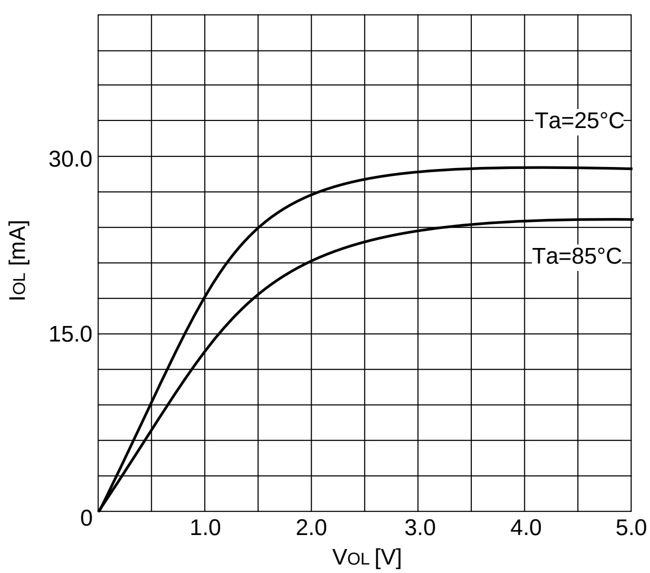
(1) P-channel I_{OH}–V_{OH} characteristics

Power source voltage: Vcc = 5 V



(2) N-channel I_{OL}–V_{OL} characteristics

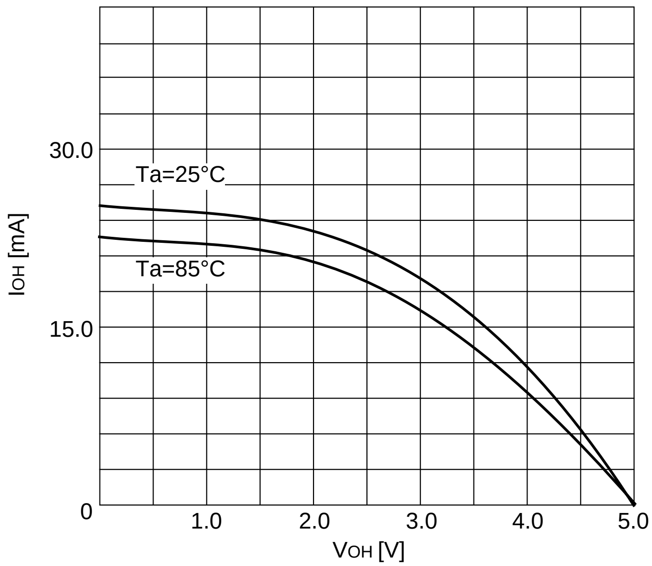
Power source voltage: Vcc = 5 V



2. Programmable I/O port (CMOS output) standard characteristics: P4, P6

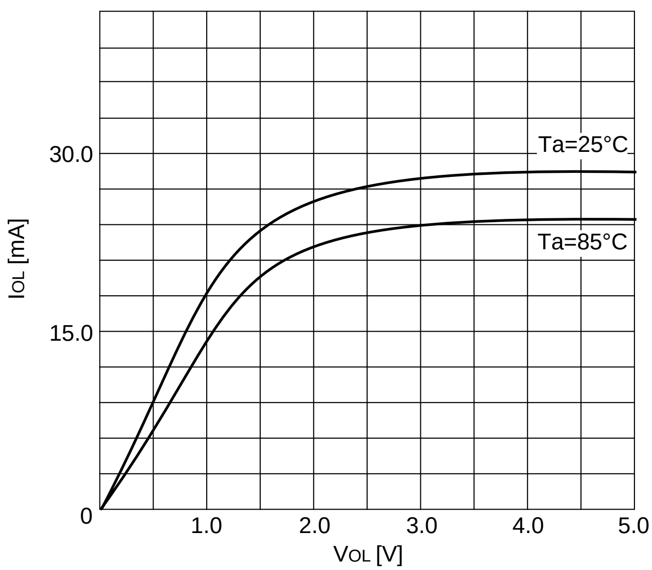
(1) P-channel I_{OH}–V_{OH} characteristics

Power source voltage: Vcc = 5 V



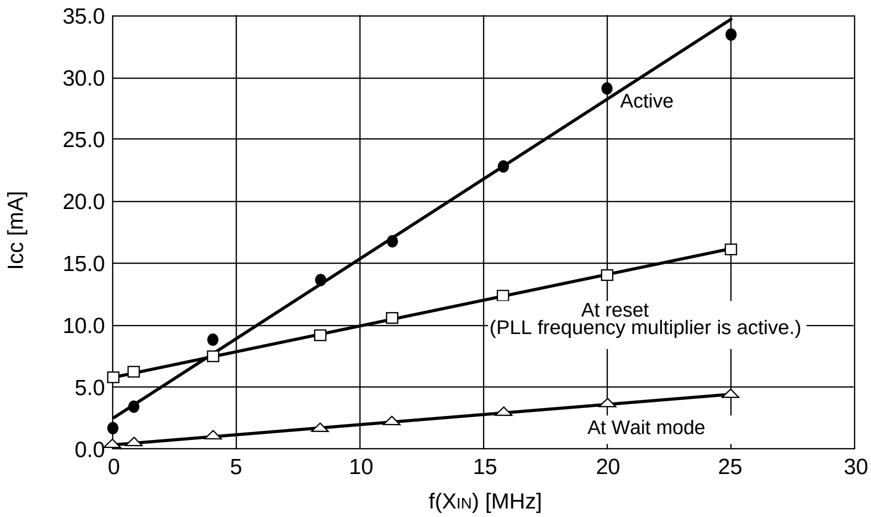
(2) N-channel I_{OL}–V_{OL} characteristics

Power source voltage: Vcc = 5 V



Appendix 10. M37905M4C-XXXFP standard characteristics

4. Icc–f(XIN) standard characteristics



Measurement condition
- •Vcc = 5.0 V
- •Ta = 25 °C
- •f(XIN) : square waveform input
- •Single-chip mode
- •PLL frequency multiplier is inactive.
- •External clock input select bit = “1”

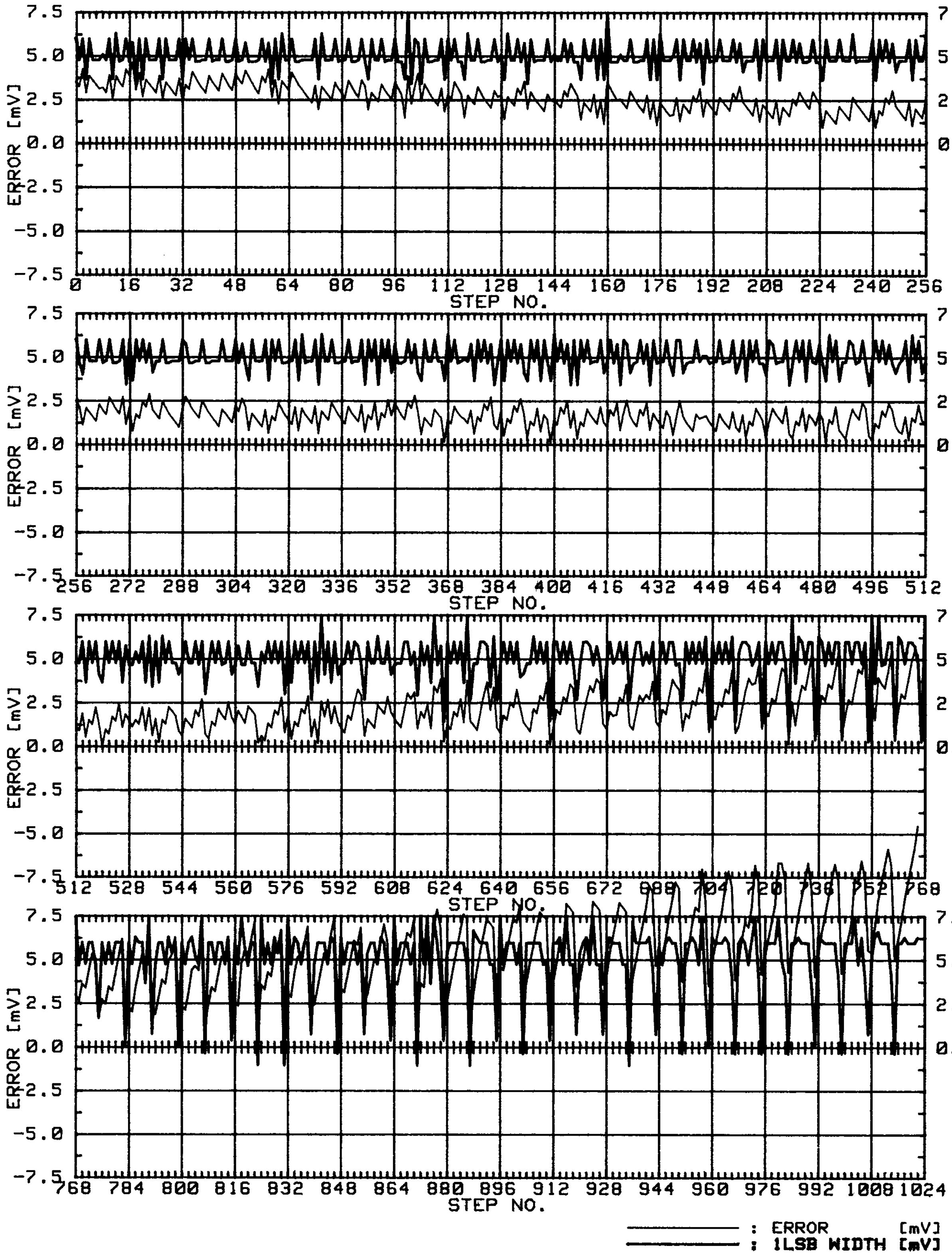
4. A-D converter standard characteristics

The lower lines of the graph indicate the absolute precision errors. These are expressed as the deviation from the ideal value when the output code changes. For example, the change in M37905M4C-XXXFP's output code from 159 to 160 should occur at 797.5 mV, but the measured value is +2.75 mV. Accordingly, the measured point of change is 797.5 + 2.75 = 800.25 mV.

The upper lines of the graph indicate the input voltage width for which the output code is constant. For example, the measured input voltage width for which the output code is 56 is 6.0 mV, so that the differential non-linear error is 6.0 – 5 = 1.0 mV (0.20 LSB).

Appendix 10. M37905M4C-XXXFP standard characteristics

(Measurement conditions Vcc = 5.0 V, V_{REF} = 5.12 V, $f(f_{sys})$ = 20 MHz, Ta = 25 °C, ϕ_{AD} = $f(f_{sys})$ divided by 2)



Appendix 11. Memory assignment of 7905 Group

1. M37905F8, M37905M8

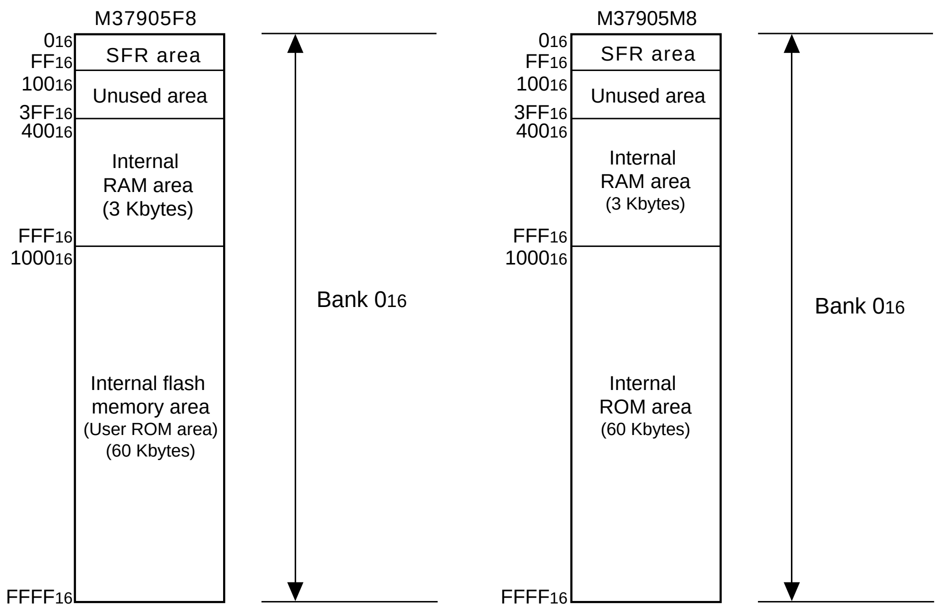
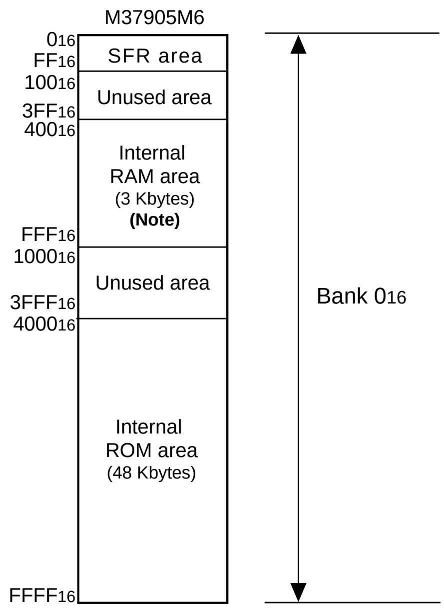


Fig. 12 Memory assigments of M37905F8, M37905M8

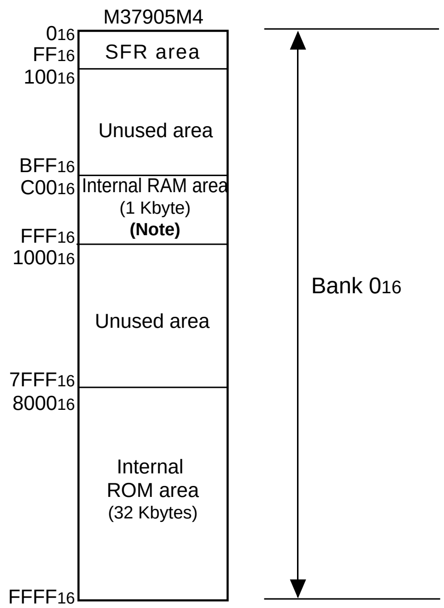
2. M37905M6



Note: Do not assign a program to the last 8 bytes of the internal RAM area.

Fig. 13 Memory assigment of M37905M6

3. M37905M4



Note: Do not assign a program to the last 8 bytes of the internal RAM area.

Fig. 14 Memory assigment of M37905M4

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
7905 Group Rev.1.0

Nov.28, 2001

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

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

User's Manual
7905 Group

Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

New publication, effective Nov. 2001.
Specifications subject to change without notice.