To all our customers

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

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

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

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

MITSUBISHI 8-BIT SINGLE-CHIP MICROCOMPUTER
740 FAMILY / 7470 SERIES

7480 Group
7481 Group

User's Manual

RENESAS
Renesas Technology Corp.

Keep safety first in your circuit designs!

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

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams and charts, represent information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

Preface

This user's manual of the Mitsubishi CMOS 8-bit microcomputer 7480 Group and 7481 Group describes the hardware specifications and applications in detail. For software information, refer to **SERIES 740 <SOFTWARE> USER'S MANUAL**, and for development support tools (assemblers, debuggers, etc.) refer to the manual attached to each tool, as well as data book **DEVELOPMENT SUPPORT TOOLS FOR MICROCOMPUTERS**.

BEFORE USING THIS USER'S MANUAL

1. Manual Contents

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

- **CHAPTER 1 – HARDWARE**
  This chapter describes the features of the microcomputers, the operation of their peripherals, and their electrical characteristics.

- **CHAPTER 2 – APPLICATIONS**
  This chapter describes usage of peripheral functions and application examples of the microcomputers, focusing on the settings of the related registers.

- **CHAPTER 3 – APPENDICES**
  This chapter describes all the control register configurations, and the mask ROM confirmation forms (mask ROM version), the ROM programming confirmation forms (one time PROM version), and the mark specification forms to be submitted at the ordering.

2. Register Configurations

An example of control register configurations of the 7480 Group and 7481 Group and the description of symbols used in them are explained below.

Contents immediately after system is released from reset (Note 1)

Bit | **Bit attributes (Note 2)**

b7	b6	b5	b4	b3	b2	b1	b0
■		■	■			0	0

CPU mode register (CPUM) [Address 00FB16]

b	Name	Function	At reset	R	W
0	Fix these bits to '0'		0	O	0
1			0	O	0
2	Stack page selection bit (**Note**)	0 : Zero page 1 : 1 page	0	O	O
3	Watchdog timer L count source selection bit	0 : f(XIN)/8 1 : f(XIN)/16	0	O	O
4	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
5	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
6	Clock division ratio selection bit	0 : f(XIN)/2 (high-speed mode) 1 : f(XIN)/8 (medium-speed mode)	0	O	O
7	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×

Note: In the products whose RAM size is 192 bytes or less, set this bit to '0'.

■ indicates the bit which is not implemented.

Notes 1: Contents after system is released from reset
0: '0' after system is released from reset
1: '1' after system is released from reset
undefined: undefined after system is released from reset

2: Bit attributes

R (Read)
- O: Read enabled
- ×: Read disabled
- Undefined: Undefined at reading
- 0: '0' at reading
- 1: '1' at reading

W (Write)
- O: Write enabled
- ×: Write disabled
- 0: Fixed to '0'
- ∗: This bit can be set to '0' by software, but cannot be set to '1'.

Table of Contents

CHAPTER 1 HARDWARE

Table of contents

CHAPTER 2 APPLICATIONS

CHAPTER 3 APPENDICES

List of Figures

CHAPTER 1 HARDWARE

List of Figures

List of Figures

CHAPTER 2 APPLICATIONS

CHAPTER 3 APPENDICES

List of Tables

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATIONS

CHAPTER 1
HARDWARE

1.1 Product Summary

The 7480 Group and 7481 Group are 8-bit microcomputers fabricated using Mitsubishi's silicon gate CMOS process. They have a simple instruction set with ROM, RAM, and input/output (I/O) interface that are located in the same memory area.

These microcomputers contain a serial I/O, an A-D converter, and a watchdog timer on a single chip, so that they are most suitable for control use in automotive controls, office machines, and home appliances. The 7480 Group and 7481 Group offer products with various types and sizes of built-in memories, as well as several choice of packages.

1.2 Group Expansion

The 7480 Group and 7481 Group are included in the 7470 series microcomputers, based on the M37470M2-XXXSP.

The 7470 series is classified as follows:

- 7470 series
  - 7470 Group
  - 7471 Group
  - 7477 Group
  - 7478 Group
  - **7480 Group**
  - **7481 Group**

Figure 1.2.1 shows the ROM/RAM expansion plan for the 7480 Group and 7481 Group.
Since these products are different only in the type and size of built-in memory, and the number of ports, the most suitable product for user's system can be easily selected.
The following products are supported in the 7480 Group and 7481 Group in addition to the mask ROM version.

(1) One Time PROM Version

This is a programmable microcomputer with built-in programmable ROM (PROM) that can be written to only one time.
For details, refer to **Section 1.20 Built-in PROM Version**.

(2) Built-in EPROM Version (with Window)

This is a programmable microcomputer with a transparent window on top of its package. Built-in EPROM can be written and erased.
For details, refer to **Section 1.20 Built-in PROM Version**.

Table 1.2.1 lists the products currently supported in the 7480 Group and 7481 Group.

1.2 Group Expansion

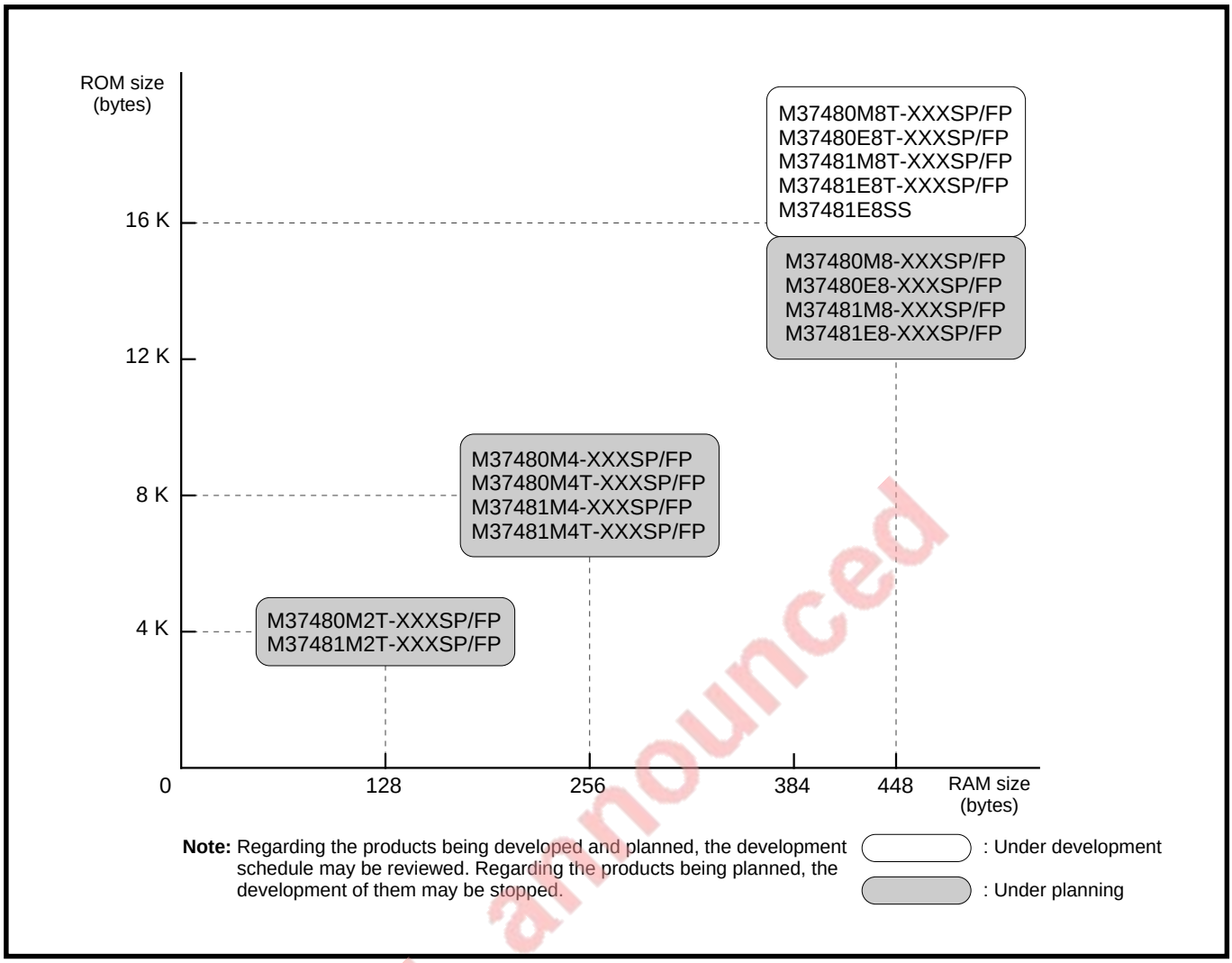


Figure 1.2.1 ROM/RAM Expansion Plan of 7480 Group and 7481 Group (As of September 1997)

Table 1.2.1 Supported Products of 7480 Group and 7481 Group (As of September 1997)

Product	ROM (bytes)	RAM (bytes)	I/O Ports	Package	Remarks
M37480M2T-XXXSP	4096	128	I/O ports: 18 Input ports: 8 (Including 4 analog input pins.)	32P4B	Mask ROM version **(Note)**
M37480M2T-XXXFP				32P2W-A	
M37480M4-XXXSP	8192	256		32P4B	Mask ROM version
M37480M4-XXXFP				32P2W-A	
M37480M4T-XXXSP				32P4B	Mask ROM version **(Note)**
M37480M4T-XXXFP				32P2W-A	
M37480M8-XXXSP	16384	448		32P4B	Mask ROM version
M37480M8-XXXFP				32P2W-A	
M37480M8T-XXXSP				32P4B	Mask ROM version **(Note)**
M37480M8T-XXXFP				32P2W-A	
M37480E8SP				32P4B	One Time PROM version (Shipped in blank)
M37480E8FP				32P2W-A	
M37480E8-XXXSP				32P4B	One Time PROM version
M37480E8-XXXFP				32P2W-A	
M37480E8T-XXXSP				32P4B	One Time PROM version **(Note)**
M37480E8T-XXXFP				32P2W-A	
M37481M2T-XXXSP	4096	128	I/O ports: 24 Input ports: 12 (Including 8 analog input pins.)	42P4B	Mask ROM version **(Note)**
M37481M2T-XXXFP				44P6N-A	
M37481M4-XXXSP	8192	256		42P4B	Mask ROM version
M37481M4-XXXFP				44P6N-A	
M37481M4T-XXXSP				42P4B	Mask ROM version **(Note)**
M37481M4T-XXXFP				44P6N-A	
M37481M8-XXXSP	16384	448		42P4B	Mask ROM version
M37481M8-XXXFP				44P6N-A	
M37481M8T-XXXSP				42P4B	Mask ROM version **(Note)**
M37481M8T-XXXFP				44P6N-A	
M37481E8SP				42P4B	One Time PROM version (Shipped in blank)
M37481E8FP				44P6N-A	
M37481E8-XXXSP				42P4B	One Time PROM version
M37481E8-XXXFP				44P6N-A	
M37481E8T-XXXSP				42P4B	One Time PROM version **(Note)**
M37481E8T-XXXFP				44P6N-A	
M37481E8SS				42S1B-A	Built-in EPROM version

Note: Extended Operating Temperature Range Version.

1.3 Performance Overviews

Tables 1.3.1 and 1.3.2 list the performance overviews of the 7480 Group and 7481 Group, respectively.

Table 1.3.1 Performance Overview of 7480 Group

Items			Performance
Number of Basic Instructions			71 (69 basic instructions of 740 Family and 2 Multiply and Divide instructions)
Instruction Execution Time			0.5 μs (the minimum instructions at f(XIN) = 8 MHz)
Clock Input Oscillation Frequency			8 MHz (Max.)
Memory Size	ROM	M37480M2	4096 bytes
		M37480M4	8192 bytes
		M37480M8/E8	16384 bytes
	RAM	M37480M2	128 bytes
		M37480M4	256 bytes
		M37480M8/E8	448 bytes
Input/ Output Ports	I/O	P0	8 bits
		P1	8 bits
		P4	**2 bits**
	Input	P2	**4 bits**
		P3	4 bits
Input/Output Characteristics		Input/Output Voltage	5 V
		Output Current	–5 mA to 10 mA (P0, P1: CMOS 3-State Buffer) 10 mA (P4: N-Channel open-drain)
Serial I/O			8 bits × 1
Timers			16-bit timer × 2 8-bit timer × 2
Subroutine Nesting		M37480M2	64 levels (Max.)
		M37480M4	96 levels (Max.)
		M37480M8/E8	192 levels (Max.)
Interrupt Sources			5 external, 8 internal, and 1 software interrupt sources
A-D Converter (Successive Comparison Conversion)			**4-channel analog inputs (alternative function of Port 2 pins)**
Clock Generator			Built-in circuit with a feedback resistor; a ceramic resonator external
Watchdog Timer			Built-in circuit
Power Supply			2.7 V to 4.5 V (f(XIN) = (2.2 VCC–2) MHz) 4.5 V to 5.5 V (f(XIN) = 8 MHz)
Power Dissipation			35 mW (typical value at f(XIN) = 8 MHz)
Operating Temperature Range			–20 °C to 85 °C (–40 °C to 85 °C for Extended Operating Temperature Range Version)
Device Structure			CMOS Silicon Gate
Package		M37480Mx/E8-XXXSP	**32-Pin Shrink Plastic DIP**
		M37480MxT/E8T-XXXSP	
		M37480Mx/E8-XXXFP	**32-Pin Plastic SOP**
		M37480MxT/E8T-XXXFP	

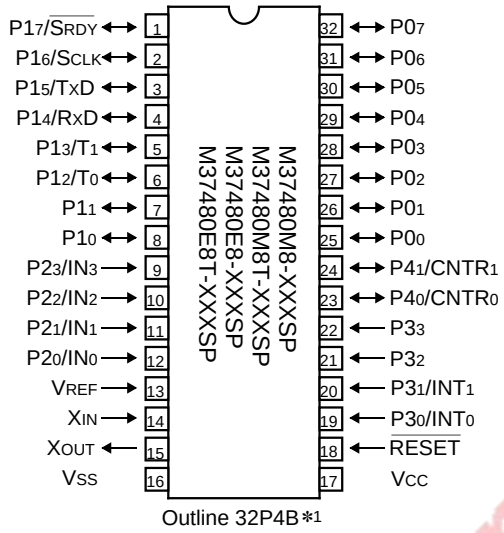
Table 1.3.2 Performance Overview of 7481 Group

Items			Performance
Number of Basic Instructions			71 (69 basic instructions of 740 Family and 2 Multiply and Divide instructions)
Instruction Execution Time			0.5 μs (the minimum instructions at f(XIN) = 8 MHz)
Clock Input Oscillation Frequency			8 MHz (Max.)
Memory Size	ROM	M37481M2	4096 bytes
		M37481M4	8192 bytes
		M37481M8/E8	16384 bytes
	RAM	M37481M2	128 bytes
		M37481M4	256 bytes
		M37481M8/E8	448 bytes
Input/ Output Ports	I/O	P0	8 bits
		P1	8 bits
		P4	**4 bits**
		P5	**4 bits**
	Input	P2	**8 bits**
		P3	4 bits
Input/Output Characteristics		Input/Output Voltage	5 V
		Output Current	–5 mA to 10 mA (P0, P1: CMOS 3-State Buffer) 10 mA (P4, P5: N-Channel open-drain)
Serial I/O			8 bits × 1
Timers			16-bit timer × 2 8-bit timer × 2
Subroutine Nesting		M37481M2	64 levels (Max.)
		M37481M4	96 levels (Max.)
		M37481M8/E8	192 levels (Max.)
Interrupt Sources			5 external, 8 internal, and 1 software interrupt sources
A-D Converter (Successive Comparison Conversion)			**8-channel analog inputs (alternative function of Port 2 pins)**
Clock Generator			Built-in circuit with a feedback resistor; a ceramic resonator external
Watchdog Timer			Built-in circuit
Power Supply			2.7 V to 4.5 V (f(XIN) = (2.2 VCC–2) MHz) 4.5 V to 5.5 V (f(XIN) = 8 MHz)
Power Dissipation			35 mW (typical value at f(XIN) = 8 MHz)
Operating Temperature Range			–20 °C to 85 °C (–40 °C to 85 °C for Extended Operating Temperature Range Version)
Device Structure			CMOS Silicon Gate
Package		M37481Mx/E8-XXXSP	**42-Pin Shrink Plastic DIP**
		M37481MxT/E8T-XXXSP	
		M37481E8SS	**42-Pin Shrink Ceramic DIP**
		M37481Mx/E8-XXXFP	**44-Pin Plastic QFP**
		M37481MxT/E8T-XXXFP	

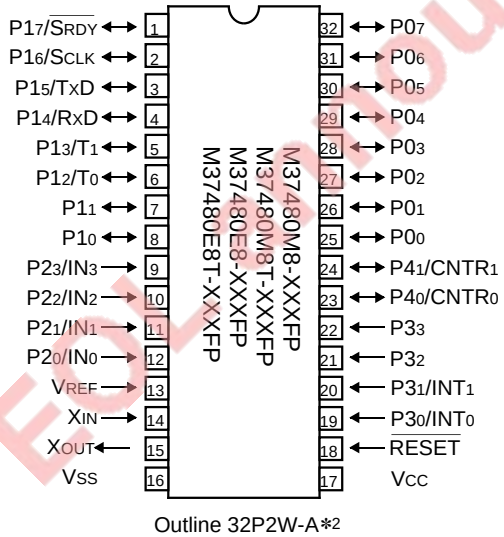
1.4 Pinouts

Figures 1.4.1 and 1.4.2 show the pinouts of the 7480 Group and 7481 Group, respectively. For the pinouts of the built-in PROM versions used in the EPROM mode, refer to **Section 1.20.1 EPROM mode**.

Pinout (top view)



Outline 32P4B ∗1



Outline 32P2W-A ∗2

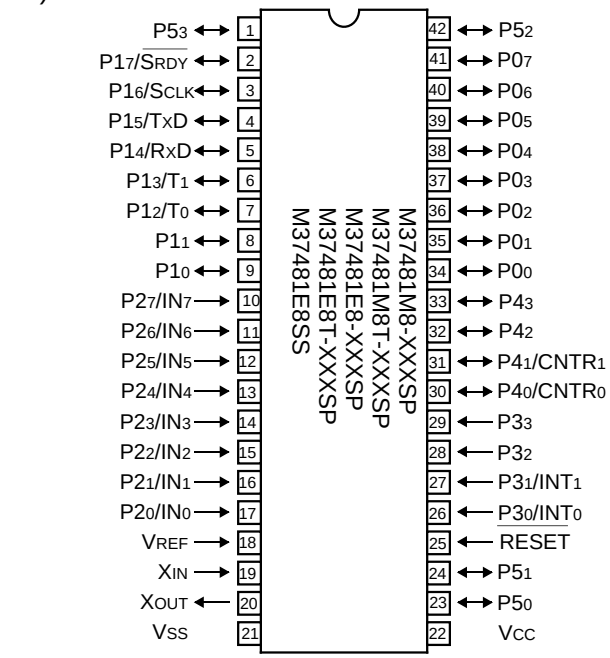
∗ **1:** The M37480M2T-XXXSP, M37480M4-XXXSP and M37480M4T-XXXSP are also included in the 32P4B packages, respectively. All of these products are pin-compatible.

2: The M37480M2T-XXXFP, M37480M4-XXXFP and M37480M4T-XXXFP are also included in the 32P2W-A packages, respectively. All of these products are pin-compatible.

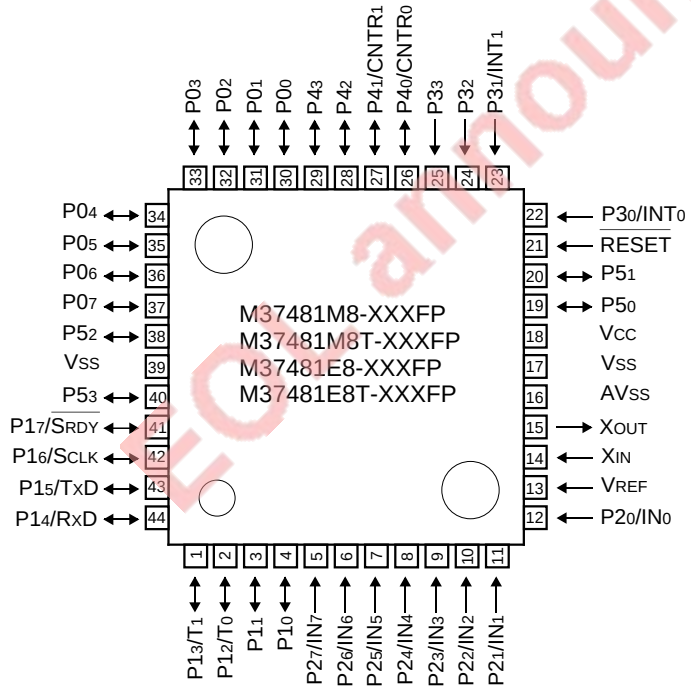
Note: The only differences between the 32P4B package product and the 32P2W-A package product are package outline and absolute maximum ratings.

Figure 1.4.1 Pinout of 7480 Group (top view)

Pinout (top view)



Outline 42P4B*1
42S1B-A (M37481E8SS)



Outline 44P6N-A *2

*** 1:** The M37481M2T-XXXSP, M37481M4-XXXSP and M37481M4T-XXXSP are also included in the 42P4B packages, respectively. All of these products are pin-compatible.
2: The M37481M2T-XXXFP, M37481M4-XXXFP and M37481M4T-XXXFP are also included in the 44P6N-A packages, respectively. All of these products are pin-compatible.
Note: The only differences between the 42P4B package product and the 44P6N-A package product are package outline, absolute maximum ratings and the fact that the 44P6N-A package product has the AVss pin.

Figure 1.4.2 Pinout of 7481 Group (top view)

1.5 Pin Descriptions

Tables 1.5.1 and 1.5.2 list the pin descriptions.
For pin functions in the EPROM mode of the built-in PROM version, refer to **Section 1.20.2 Pin Descriptions.**

Table 1.5.1 Pin Descriptions (1)

Pin	Name	Input/ Output	Function
V_{CC}, V_{SS}	Power source		• Apply the following voltage to the V_{CC} pin: 2.7 V to 4.5 V (at f(X_{IN}) = (2.2 V_{CC}–2) MHz), or 4.5 V to 5.5 V (at f(X_{IN}) = 8 MHz). • Apply 0 V to the V_{SS} pin.
AV_{SS}	Analog power source		• Ground level input pin for the A-D converter • Apply the same voltage as for the V_{SS} pin to the AV_{SS} pin. **Note:** This pin is dedicated to the 44P6N-A package products in the 7481 Group.
V_{REF}	Reference voltage input	Input	• Reference voltage input pin for A-D converter • Apply the following voltage to the V_{REF} pin: 2 V to V_{CC} V when V_{CC} = 2.7 V to 4.0 V, or 0.5 V_{CC} (≥ 2) to V_{CC} V when V_{CC} = 4.0 V to 5.5 V. **Note:** When not using A-D converter, connect V_{REF} pin to V_{CC}.
$\overline{RESET}$	Reset input	Input	• Reset input pin • System Reset: Holding the LOW level for 2 μs or more forces CPU into reset state.
X_{IN}	Clock input	Input	• I/O pins for clock generator • A ceramic resonator is connected between pins X_{IN} and X_{OUT}. • When an external clock is used, it is input to X_{IN} pin, and leave X_{OUT} pin open. • A feedback resistor is built in between pins X_{IN} and X_{OUT}.
X_{OUT}	Clock output	Output	

Table 1.5.2 Pin Descriptions (2)

Pin	Name	Input/ Output	Function
$P0_0$–$P0_7$	I/O port P0	I/O	• 8-bit I/O port pins • The output structure is CMOS output. • When an input port is selected, a pull-up transistor can be connectable by the bit. • In input mode, a key-on wake up function is provided.
$P1_0$–$P1_7$	I/O port P1	I/O	• 8-bit I/O port pins • The output structure is CMOS output. • When an input port is selected, a pull-up transistor can be connected by the 4 bits. • $P1_2$ and $P1_3$ serve the alternative functions of the timer output pins T_0 and T_1. • $P1_4$, $P1_5$, $P1_6$, and $P1_7$ serve the alternative functions of the serial I/O pins RxD, TxD, SCLK and $\overline{\text{SRDY}}$, respectively.
$P2_0$–$P2_7$	Input port P2	Input	• 8-bit input port pins • $P2_0$–$P2_7$ serve the alternative functions of the analog input pins IN_0–IN_7. **Note:** The 7480 Group has only four pins of $P2_0$–$P2_3$ (IN_0–IN_3).
$P3_0$–$P3_3$	Input port P3	Input	• 4-bit input port pins • $P3_0$ and $P3_1$ serve the alternative functions of the external interrupt input pins INT_0 and INT_1.
$P4_0$–$P4_3$	I/O port P4	I/O	• 4-bit I/O port pins • The output structure is N-channel open-drain outputs with built-in clamping diodes. • $P4_0$ and $P4_1$ serve the alternative functions of the timer I/O pins $CNTR_0$ and $CNTR_1$. **Note:** The 7480 Group has only two pins of $P4_0$ and $P4_1$.
$P5_0$–$P5_3$	I/O port P5	I/O	• 4-bit I/O port pins • The output structure is N-channel open-drain outputs with built-in clamping diodes. **Note:** The 7480 Group is not provided with port P5.

1.6 Functional Block Diagrams

Figures 1.6.1, 1.6.2 and 1.6.3 show the functional block diagrams of the 7480 Group and 7481 Group.

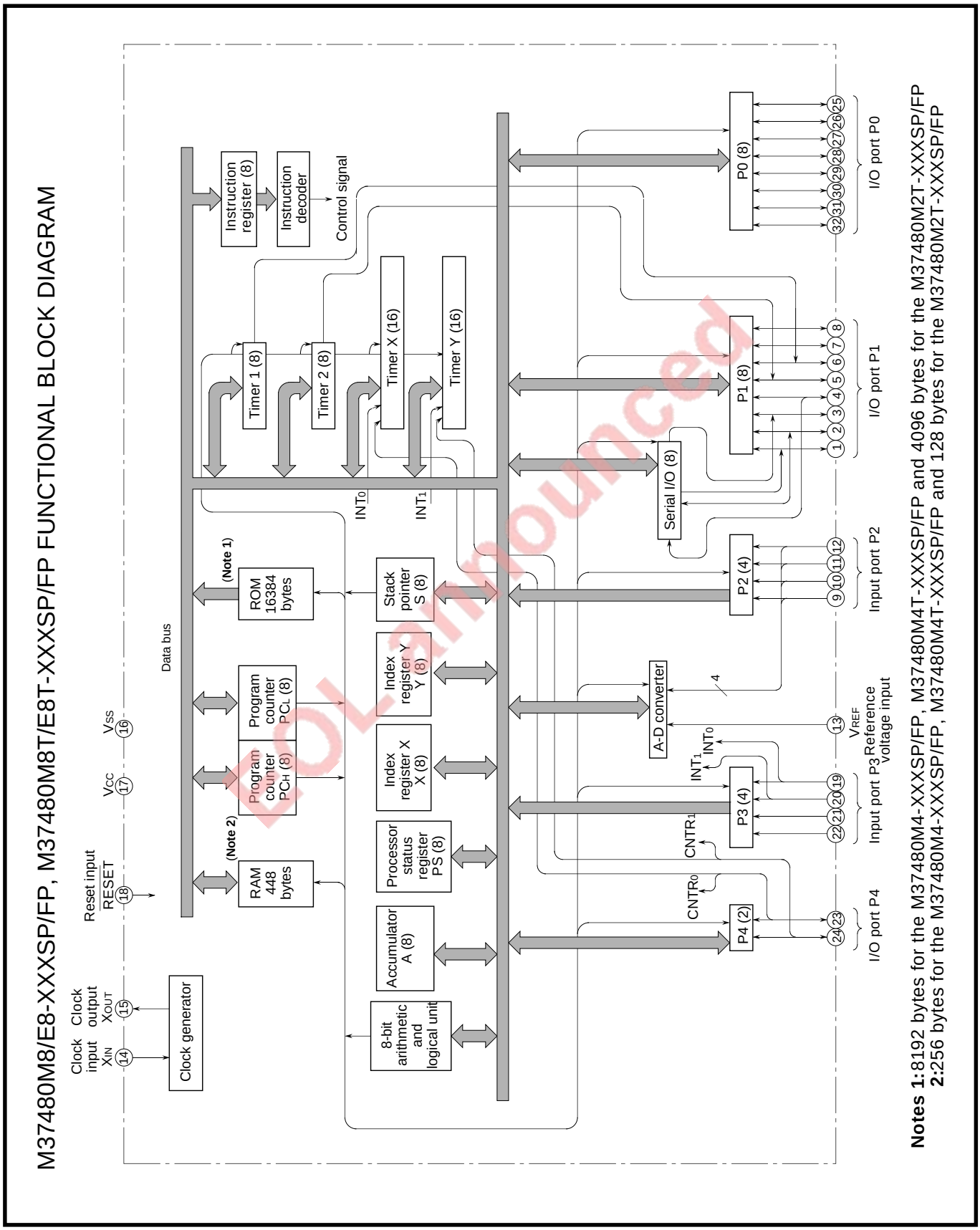


Figure 1.6.1 M37480Mx/E8-XXXSP/FP and M37480MxT/E8T-XXXSP/FP Functional Block Diagram

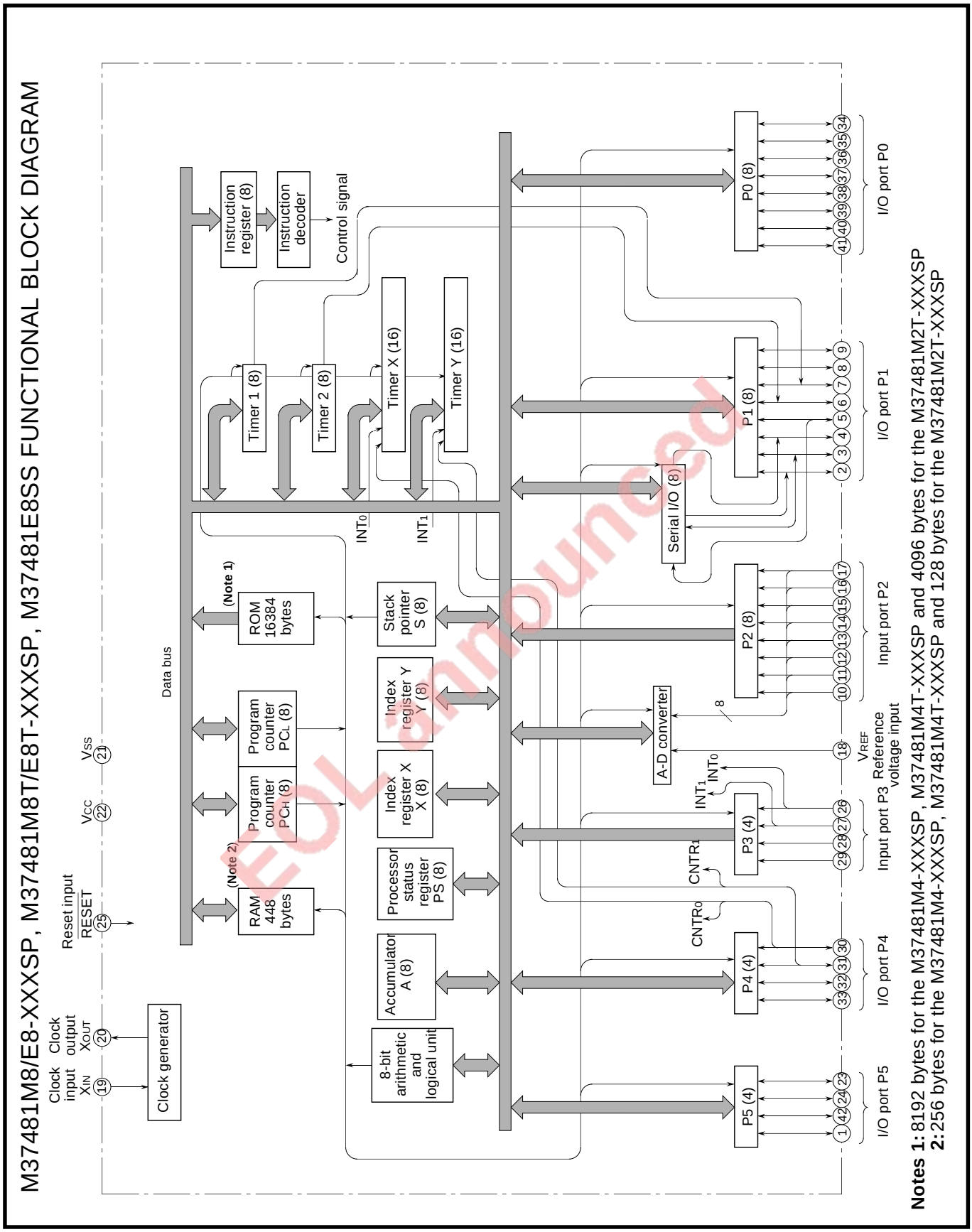


Figure 1.6.2 M37481Mx/E8-XXXSP, M37481MxT/E8T-XXXSP and M37481E8SS Functional Block Diagram

1.6 Functional Block Diagrams

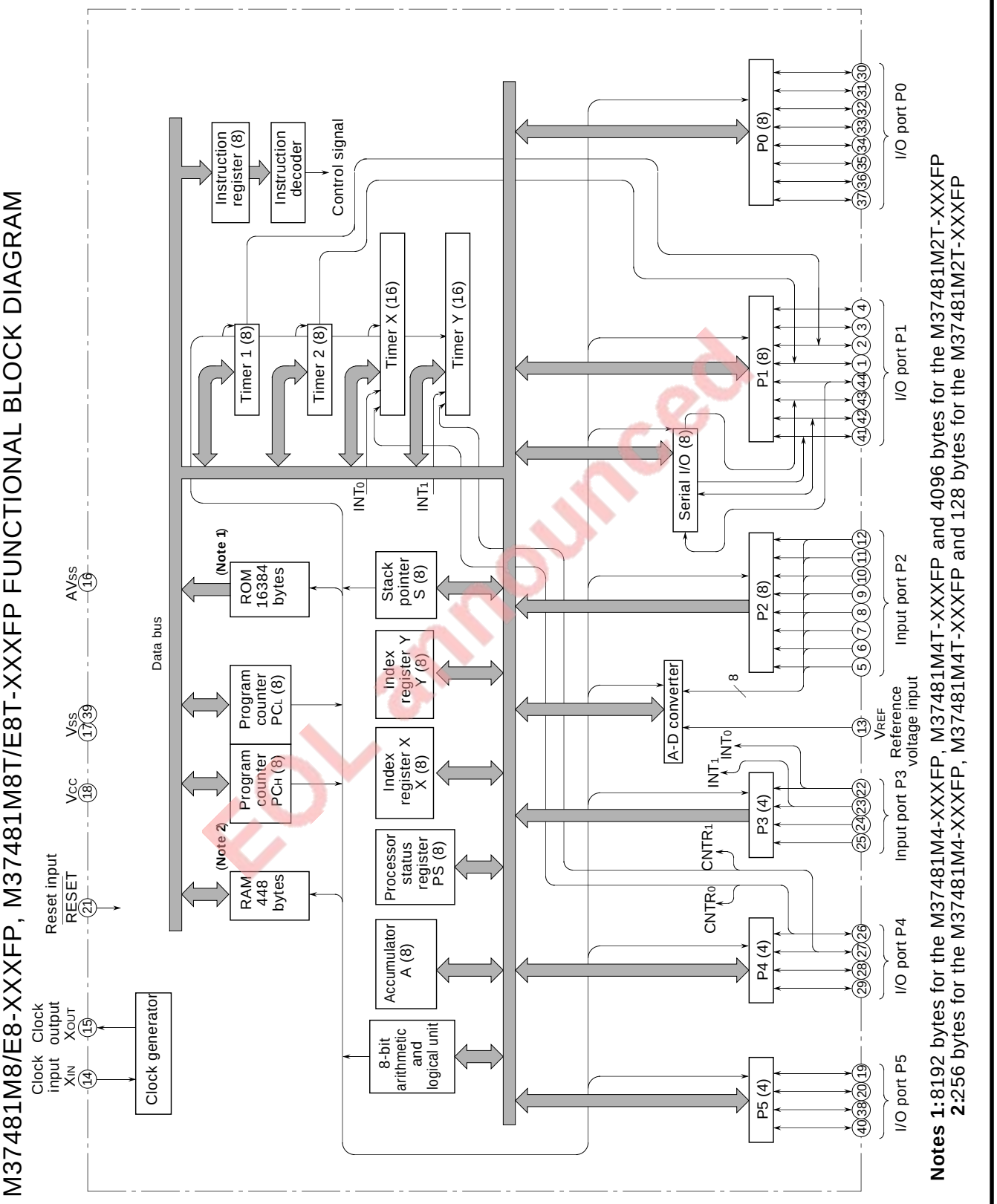


Figure 1.6.3 M37481Mx/E8-XXXFP, M37481MxT/E8T-XXXFP Functional Block Diagram

1.7 Central Processing Unit (CPU)

The 7480 Group and 7481 Group have the CPU common to the 740 family.
For the description of the instructions, refer to the following:

- **Section 3.6 Machine Instructions**
- **740 FAMILY CPU CORE BASIC FUNCTIONS: ADDRESSING MODE** in data book **SINGLE CHIP 8-BIT MICROCOMPUTERS**
- **SERIES 740 <SOFTWARE> USER'S MANUAL**

The instructions which characterize the group are as follows:

1. **FST** and **SLW** instructions are excluded.
2. **MUL** and **DIV** instructions are available.
3. **WIT** instruction is available (**Note**).
4. **STP** instruction is available (**Note**).

Note: For the above instructions, refer to **Section 1.19 Power Saving Function.**

The CPU has the six registers (CPU internal registers).
Figure 1.7.1 shows the CPU internal registers.

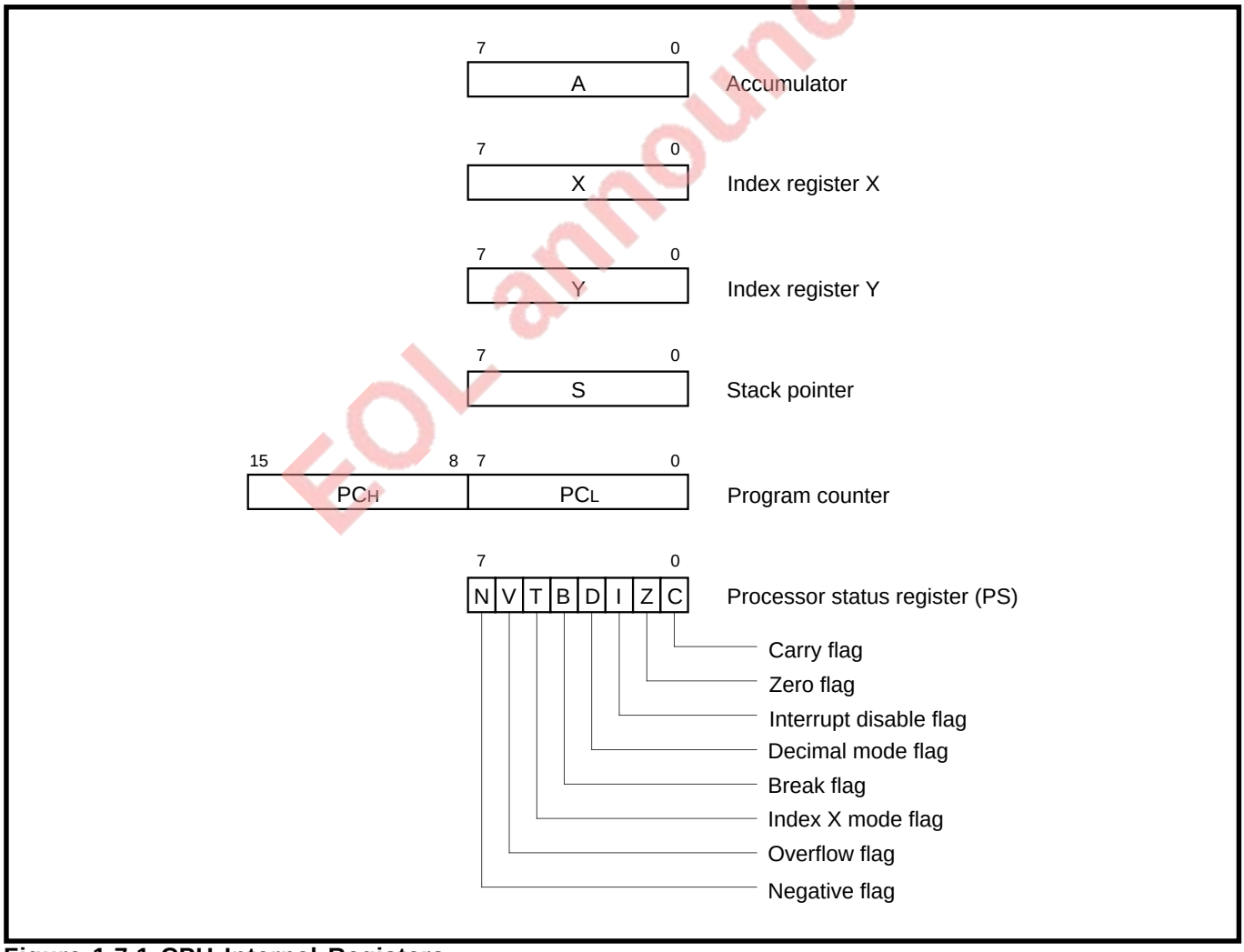


Figure 1.7.1 CPU Internal Registers

1.7 Central Processing Unit (CPU)

States of the CPU internal registers immediately after system is released from reset are as follows:
- The interrupt disable flag (I) of the processor status register (PS) is set to '1'.
- The high-order 8 bits (PC_H) of the program counter contain the contents of address '$FFFF_{16}$', and the low-order 8 bits (PC_L) contain the contents of address '$FFFE_{16}$'.

Since the contents of the CPU internal registers not mentioned above are undefined immediately after system is released from reset, it is necessary to initialize these registers by software.

1.7.1 Accumulator (A)

The accumulator is an 8-bit register. Data manipulations, such as arithmetic or logical operation and transfers, are performed using this register.

1.7.2 Index Register X (X)

Index register X is an 8-bit register that performs addressing in the index addressing mode.

1.7.3 Index Register Y (Y)

Index register Y is an 8-bit register that performs addressing for certain instructions in the index addressing mode.

1.7.4 Stack Pointer (S)

The stack pointer is an 8-bit register. It indicates the start address of the stack area where the contents of registers pushed at subroutine call or interrupt are stored.
The low-order 8 bits in the stack are addressed by the stack pointer, and the high-order 8 bits are addressed by the content of the stack page selection bit. When this bit is '0', the high-order 8 bits indicate '00_{16}', and when '1', they indicate '01_{16}'.

For the 7480 Group and 7481 Group, the stack page selection bit is assigned to bit 2 of the CPU mode register (address $00FB_{16}$). Set this bit to '1' if necessary, because it is cleared to '0' at reset.
Note: In the 7480 Group and 7481 Group, however, the product with RAM whose memory size is 192 bytes or less does not have RAM on 1 page. Therefore, clear this bit to '0'.
Figure 1.7.2 shows the operation for pushing onto and pulling from the stack. Push the contents of necessary registers other than those described here onto stack by software.
Table 1.7.1 lists the push and pull instructions for the accumulator and the processor status register.

Initialize the stack pointer by software because it is undefined immediately after system is released from reset.

Table 1.7.1 Push and Pull Instructions for Accumulator and Processor Status Register

	Push Instructions	Pull Instructions
Accumulator	**PHA**	**PLA**
Processor Status Register	**PHP**	**PLP**

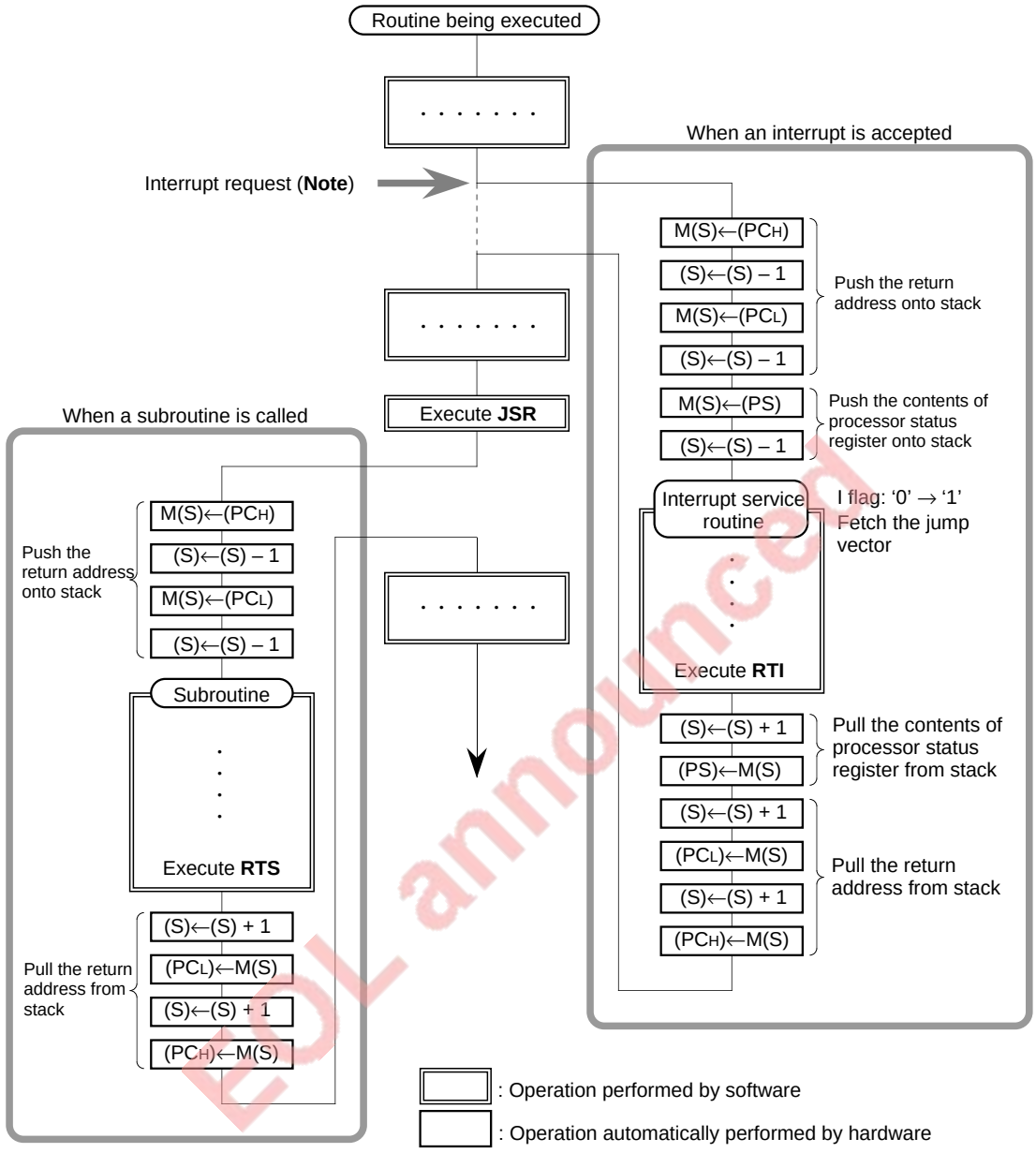


Figure 1.7.2 Operation for Pushing onto and Pulling from Stack

1.7 Central Processing Unit (CPU)

1.7.5 Program Counter (PC)

The program counter is a 16-bit counter consisting of the high-order 8 bits (PC_H) and the low-order 8 bits (PC_L). The program counter indicates the address of the program memory to be next fetched.
At reset, the high-order 8 bits (PC_H) of the program counter contain the contents of address '$FFFF_{16}$', and the low-order 8 bits (PC_L) contain the contents of address '$FFFE_{16}$'.

1.7.6 Processor Status Register (PS)

The processor status register is an 8-bit register. This register consists of 5 flags which hold the states immediately after arithmetic or logical operation, and 3 flags which determine the CPU operation.
C, Z, V, and N flags are used to test the branch instructions. However, Z, V, and N flags are invalid in the decimal mode.
Each flag of the processor status register is described below. Also, Table 1.7.2 lists the instructions that set these flags to '1' or '0'.

(1) Carry Flag C (bit 0)

This flag holds a carry or a borrow from the arithmetic logic unit after following an arithmetic or logical operation. Also, the shift and rotate instructions can affect the content of this flag.
The Carry flag is set to '1' by using the **SEC** instruction and cleared to '0' by using the **CLC** instruction.

(2) Zero Flag Z (bit 1)

This flag is '1' when the result of an arithmetic, logical or transfer operation is '0', otherwise it is '0'. The Zero flag is invalid in the decimal mode.
There is no instruction that can affect the content of this flag.

(3) Interrupt Disable Flag I (bit 2)

This flag disables all interrupts except the **BRK** instruction interrupt. When it is set to '1', interrupt is disabled. When an interrupt is accepted, the flag automatically goes to '1'.
This flag is set to '1' by using the **SEI** instruction and cleared to '0' by using the **CLI** instruction.
Note: This flag is set to '1' (interrupt disabled) at reset.

(4) Decimal Mode Flag D (bit 3)

This flag determines whether addition and subtraction are performed in the binary or decimal mode. When this flag is '0', ordinary binary operation is performed; On the other hand, when it is '1', an 8-bit word is handled as a decimal number of two digits. Decimal adjust is automatically performed in the decimal operation. However, the decimal operation can be performed only at the **ADC** and **SBC** instructions.
This flag is set to '1' by using the **SED** instruction and cleared to '0' by using the **CLD** instruction.
Note: This flag is undefined at reset; then it is necessary to initialize this flag because it directly affects the result of arithmetic operation.

(5) Break Flag B (bit 4)

This flag recognizes whether an interrupt occurs by using the **BRK** instruction. The contents of processor status register are pushed onto the stack when the following occurs;

• the contents of this flag is set to '1' when an interrupt occurs by using the **BRK** instruction, or

• this flag is set to '0' by the all other interrupts.

There is no instruction that can affect the content of this flag.

(6) Index X Mode Flag T (bit 5)

When this flag is '0', operation is performed between the accumulator and memories. When this flag is '1', operation is directly done between memories without using the accumulator. This flag is set to '1' by using the **SET** instruction and cleared to '0' by using the **CLT** instruction.

Note: This flag is undefined at reset; it is therefore necessary to initialize this flag because it directly affects the result of operation.

(7) Overflow Flag V (bit 6)

This flag is used in adding or subtracting an 8-bit word as signed binary digits. When the result of addition or subtraction exceeds the range of +127 to -128, this flag is set to '1'. When the **BIT** instruction is executed, the content of bit 6 of the activated memory is written to the flag.

This flag is cleared to '0' by using the **CLV** instruction. However, there is no instruction that can set this flag to '1'.

In the decimal mode, this flag is invalid.

(8) Negative Flag N (bit 7)

When the result of arithmetic, logical or transfer operation is negative (bit 7 is '1'), this flag is set to '1'. When the **BIT** instruction is executed, the content of bit 7 of the activated memory is written to the flag.

There is no instruction that can directly affect the content of this flag.

In the decimal mode, this flag is invalid.

Table 1.7.2 Instructions to Set Flags of Processor Status Register to '1' or '0'

	C Flag	Z Flag	I Flag	D Flag	B Flag	T Flag	V Flag	N Flag
Instructions to Set Flags to '1'	**SEC**	–	**SEI**	**SED**	–	**SET**	–	–
Instructions to Set Flags to '0'	**CLC**	–	**CLI**	**CLD**	–	**CLT**	**CLV**	–

1.8 Access Area

For the 7480 Group and 7481 Group, all ROM, RAM and I/O and various control registers are located in the same access area. Therefore, data transfer, arithmetic and logical operations can be accomplished by the same instructions without identifying between memory and I/O interface.
The program counter consists of 16 bits and can access the 64K-byte area of addresses '0000_{16}' through '$FFFF_{16}$'.
The area of the least significant 256 bytes (addresses '0000_{16}' through '$00FF_{16}$') is called the 'zero page'. Frequently accessed memory such as an internal RAM, I/O ports, timers, etc are located in this area. Furthermore, the area of the most significant 256 bytes (addresses '$FF00_{16}$' through '$FFFF_{16}$') is called the 'special page'. An internal ROM and interrupt vectors are located in this area.
Both the zero page and the special page can be accessed with two bytes, using the specific mode for each page.

Figure 1.8.1 shows the outline of the access area.

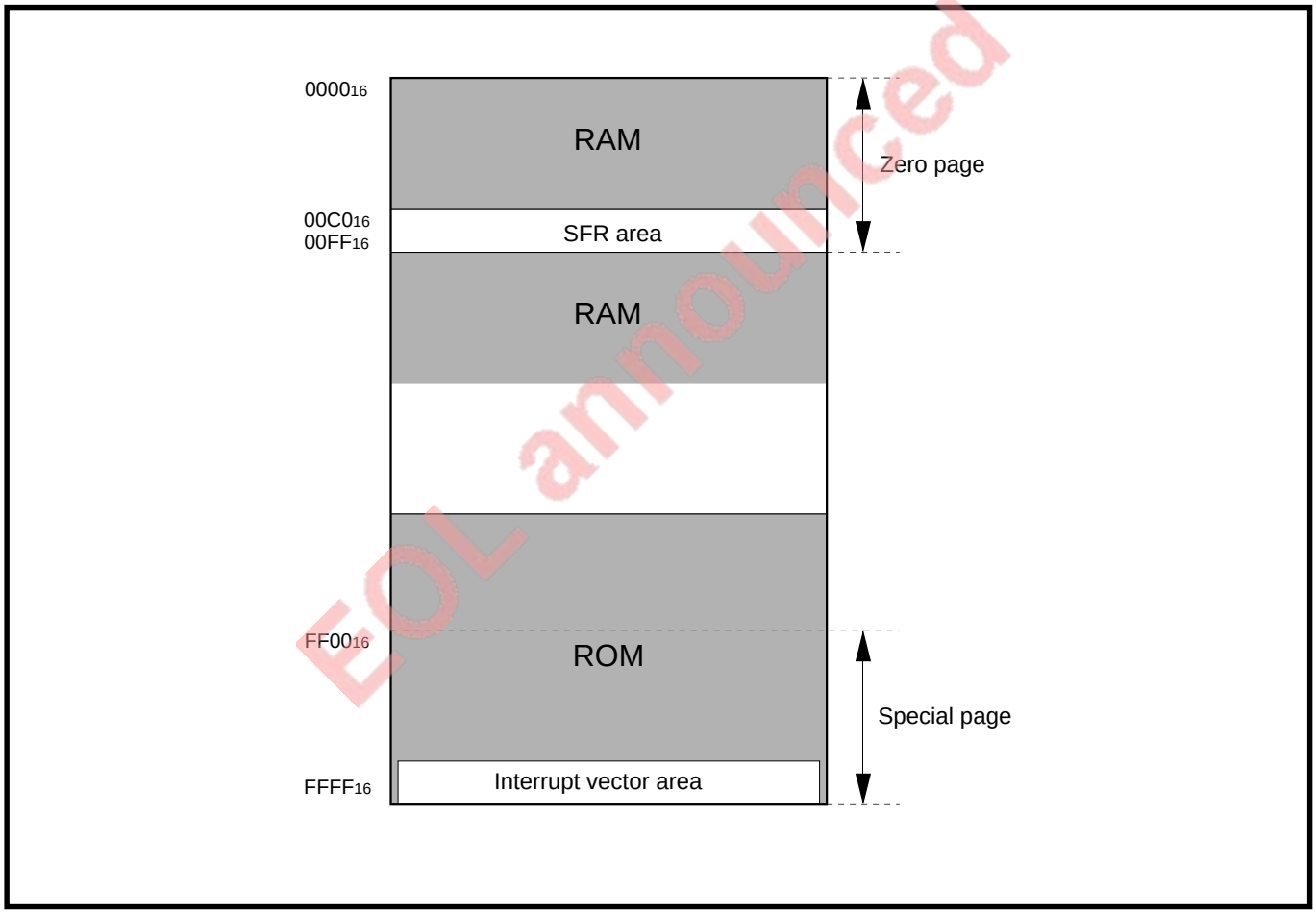


Figure 1.8.1 Access Area

1.8.1 Zero Page (Addresses '0000$_{16}$' through '00FF$_{16}$')

The area of 256 bytes from addresses '0000$_{16}$' through '00FF$_{16}$' is called the zero page. The internal RAM and the special function registers (SFR) are located in this area.

The addressing modes shown in Table 1.8.1 are used to specify memory or registers in this area. In the mode listed, the zero page addressing mode can be used to access this area by shorter instruction cycles.

1.8.2 Special Page (Addresses 'FF00$_{16}$' through 'FFFF$_{16}$')

The area of 256 bytes from addresses 'FF00$_{16}$' through 'FFFF$_{16}$' is called the special page. The internal ROM and the interrupt vector area are located in this area.

The addressing modes shown in Table 1.8.1 are used to specify memory or subroutines in this area. In the mode listed, the special page addressing mode can be used to jump to this area by shorter instruction cycles.

Ordinary, frequently used subroutines are located in this area.

Table 1.8.1 Addressing Mode Accessible to Each Area

Addressing Mode (Required Bytes)	Reference to Zero Page	Reference to Special Page	Reference to Other Areas
Zero Page (2)	O	–	–
Zero Page Indirect (2)	O	–	–
Zero Page X (2)	O	–	–
Zero Page Y (2)	O	–	–
Zero Page Bit (2)	O	–	–
Zero Page Bit Relative (3)	O	–	–
Absolute (3)	O	O	O
Absolute X (3)	O	O	O
Absolute Y (3)	O	O	O
Relative (2)	O	O	O
Indirect (3)	O	O	O
Indirect X (2)	O	O	O
Indirect Y (2)	O	O	O
Special Page (2)	–	O	–

1.9 Memory Maps

Figure 1.9.1 shows the memory maps of the 7480 Group and 7481 Group.
Memories and I/Os located in the access area are described below.

- **RAM**
  The internal RAM is located in the area listed in Table 1.9.1. Internal RAM is used for data storage, the stack area used subroutine call or interrupt generation.
  To prevent the contents of RAM from being destroyed, take the depth of subroutine nesting and the level of interrupt into consideration when using RAM as the stack area.

- **Special Function Registers (SFR) (Addresses '00C0$_{16}$' through '00FF$_{16}$')**
  Special function registers (SFR) are assigned to addresses '00C0$_{16}$' through '00FF$_{16}$'. Various control registers for the I/O ports, the timers, the serial I/O, the A-D converter, and the interrupts are located in the SFR area.
  Figure 1.9.2 shows the memory map of the SFR area.

- **ROM**
  The internal ROM is located in the area listed in Table 1.9.2. Internal ROM is used to store data tables and programs. In the 7480 Group and 7481 Group, addresses 'FFE4$_{16}$' through 'FFFF$_{16}$' of the ROM area are assigned to the vector area where the jump addresses after system is released from reset and interrupt generation are stored.
  Figure 1.9.3 shows the memory map of the interrupt vector area.

Table 1.9.1 RAM Area

Product	Range of RAM Area	RAM Size
M3748xM2	Addresses '0000$_{16}$' through '007F$_{16}$'	128 × 8 bits
M3748xM4	Addresses '0000$_{16}$' through '00BF$_{16}$', Addresses '0100$_{16}$' through '013F$_{16}$'	256 × 8 bits
M3748xM8/E8	Addresses '0000$_{16}$' through '00BF$_{16}$', Addresses '0100$_{16}$' through '01FF$_{16}$'	448 × 8 bits

Table 1.9.2 ROM Area

Product	Memory Type	Range of ROM Area	ROM Size
M3748xM2	Mask ROM	Addresses 'F000$_{16}$' through 'FFFF$_{16}$'	4K × 8 bits
M3748xM4	Mask ROM	Addresses 'E000$_{16}$' through 'FFFF$_{16}$'	8K × 8 bits
M3748xM8	Mask ROM	Addresses 'C000$_{16}$' through 'FFFF$_{16}$'	16K × 8 bits
M3748xE8	PROM		

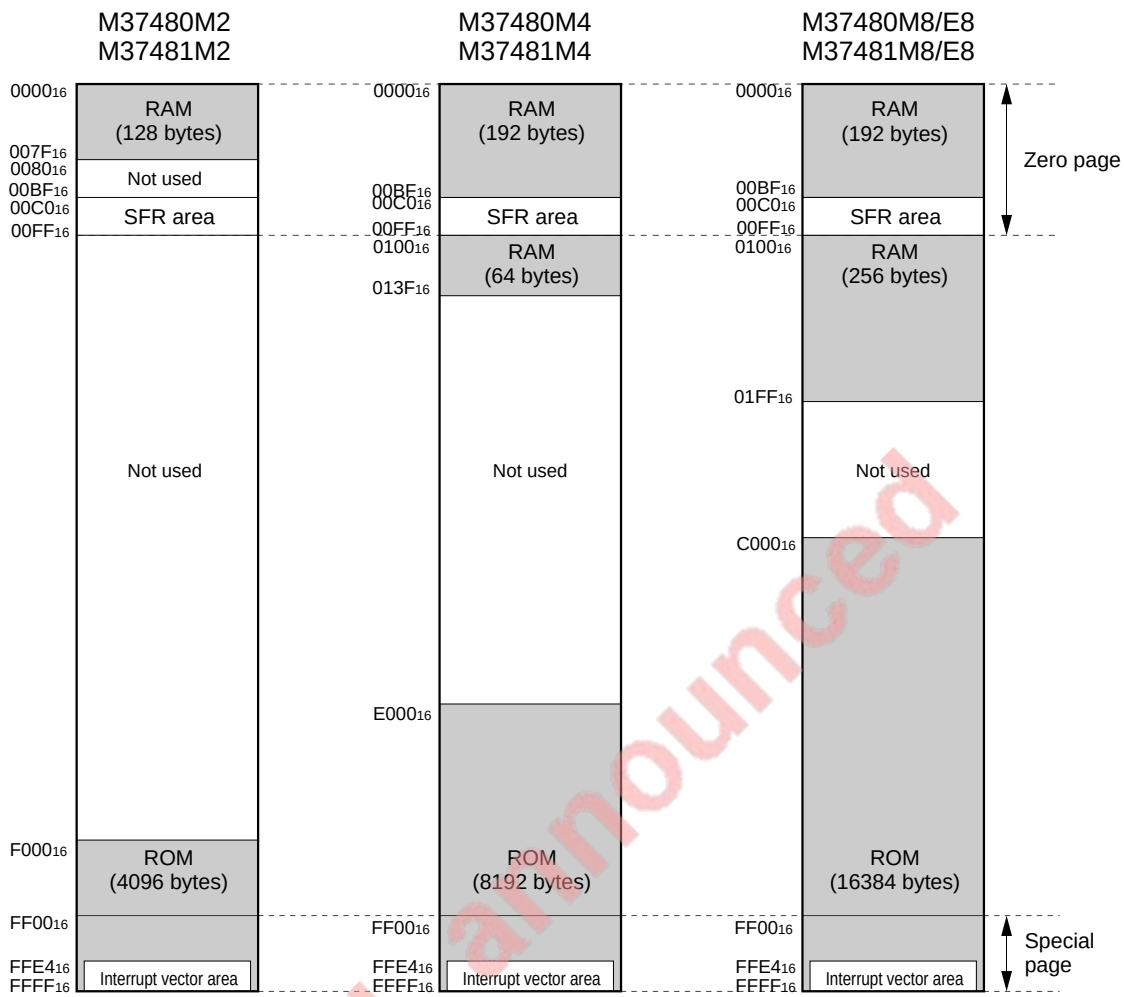


Figure 1.9.1 Memory Maps of 7480 Group and 7481 Group

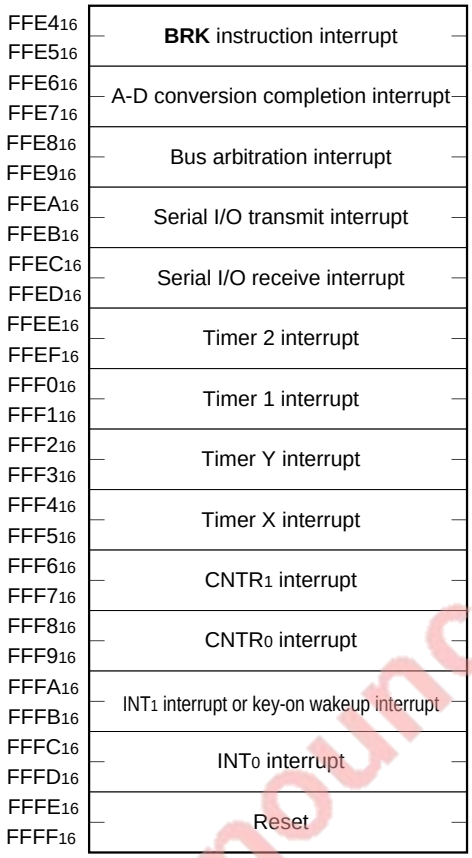
1.9 Memory Maps

00C0₁₆	Port P0 register (P0)	00E0₁₆	Transmit/receive buffer register (TB/RB)
00C1₁₆	Port P0 direction register (P0D)	00E1₁₆	Serial I/O status register (SIOSTS)
00C2₁₆	Port P1 register (P1)	00E2₁₆	Serial I/O control register (SIOCON)
00C3₁₆	Port P1 direction register (P1D)	00E3₁₆	UART control register (UARTCON)
00C4₁₆	Port P2 register (P2)	00E4₁₆	Baud rate generator (BRG)
00C5₁₆		00E5₁₆	Bus collision detection control register (BUSARBCON)
00C6₁₆	Port P3 register (P3)	00E6₁₆	
00C7₁₆		00E7₁₆	
00C8₁₆	Port P4 register (P4)	00E8₁₆	
00C9₁₆	Port P4 direction register (P4D)	00E9₁₆	
00CA₁₆	Port P5 register (P5)	00EA₁₆	
00CB₁₆	Port P5 direction register (P5D)	00EB₁₆	
00CC₁₆		00EC₁₆	
00CD₁₆		00ED₁₆	
00CE₁₆		00EE₁₆	
00CF₁₆		00EF₁₆	Watchdog timer H (WDTH)
00D0₁₆	Port P0 pull-up control register (P0PCON)	00F0₁₆	Timer X low-order (TXL)
00D1₁₆	Port P1 pull-up control register (P1PCON)	00F1₁₆	Timer X high-order (TXH)
00D2₁₆	Port P4P5 input control register (P4P5CON)	00F2₁₆	Timer Y low-order (TYL)
00D3₁₆		00F3₁₆	Timer Y high-order (TYH)
00D4₁₆	Edge polarity selection register (EG)	00F4₁₆	Timer 1 (T1)
00D5₁₆		00F5₁₆	Timer 2 (T2)
00D6₁₆		00F6₁₆	Timer X mode register (TXM)
00D7₁₆		00F7₁₆	Timer Y mode register (TYM)
00D8₁₆		00F8₁₆	Timer XY control register (TXYCON)
00D9₁₆	A-D control register (ADCON)	00F9₁₆	Timer 1 mode register (T1M)
00DA₁₆	A-D conversion register (AD)	00FA₁₆	Timer 2 mode register (T2M)
00DB₁₆		00FB₁₆	CPU mode register (CPUM)
00DC₁₆		00FC₁₆	Interrupt request register 1 (IREQ1)
00DD₁₆		00FD₁₆	Interrupt request register 2 (IREQ2)
00DE₁₆	**STP** instruction operation control register (STPCON)	00FE₁₆	Interrupt control register 1 (ICON1)
00DF₁₆		00FF₁₆	Interrupt control register 2 (ICON2)

(**Note**)

Note: These registers are not allocated in the 7480 Group.

Figure 1.9.2 Memory Map of SFR Area



Note: Refer to **Section 1.11 Interrupts** for each interrupt overview.

Figure 1.9.3 Memory Map of Interrupt Vector Area

1.10 Input/Output Pins

The Input/Output (I/O) pins of the 7480 Group and 7481 Group are classified as follows:

- I/O port pins ($P0_0$–$P0_7$, $P1_0$–$P1_7$, $P4_0$–$P4_3$, and $P5_0$–$P5_3$)
- Input port pins ($P2_0$–$P2_7$ and $P3_0$–$P3_3$)
- Reset input pin ($\overline{RESET}$)
- Clock input and output pins (X_{IN} and X_{OUT})
- A-D conversion reference voltage input pin (V_{REF})
- Power source pins (V_{CC}, V_{SS}, and AV_{SS})

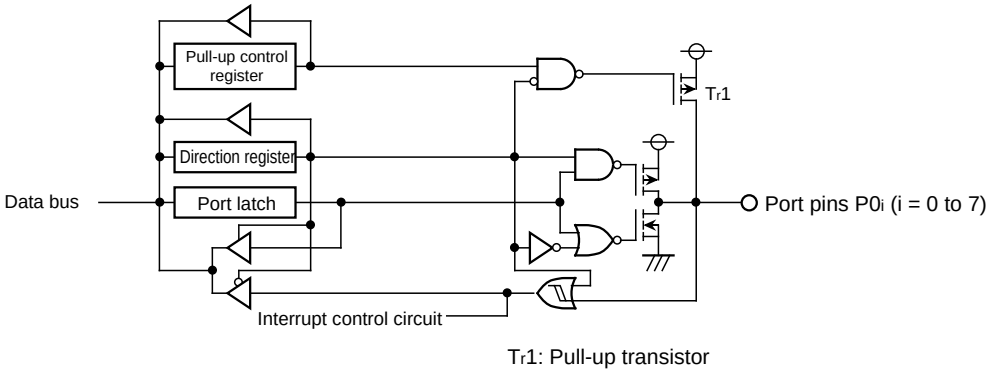
Notes 1: The 7480 Group does not have port pins $P2_4$–$P2_7$, $P4_2$, $P4_3$, and $P5_0$–$P5_3$.
2: The AV_{SS} pin is dedicated to the 44P6N-A package products in the 7481 Group.

For the functions of each pin, refer to **Section 1.5 Pin Descriptions.**

1.10.1 Block Diagrams

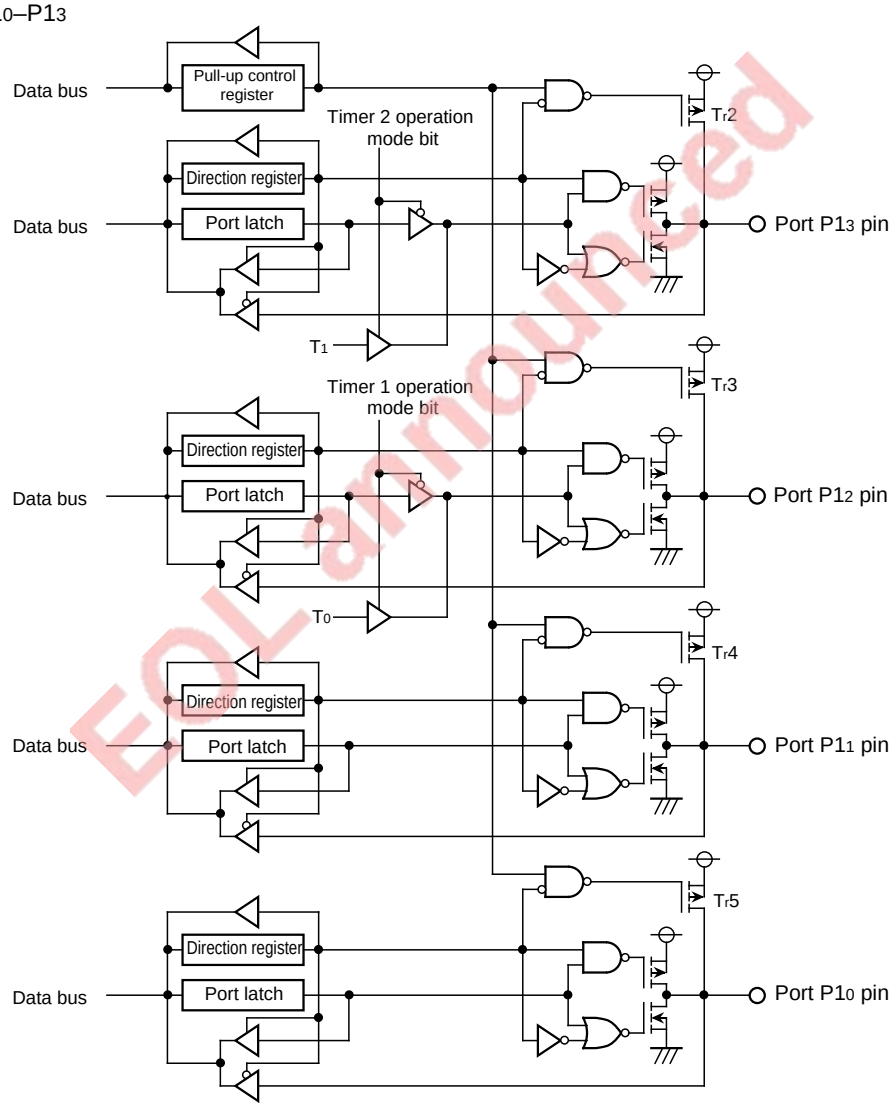
Figures 1.10.1, 1.10.2, and 1.10.3 show the block diagrams of the I/O and input port pins.

● Port P0



Tr1: Pull-up transistor

● Ports P10–P13



Tr2 to Tr5: Pull-up transistor

Figure 1.10.1 Block Diagrams of Port Pins P0i and P10–P13

1.10 Input/Output Pins

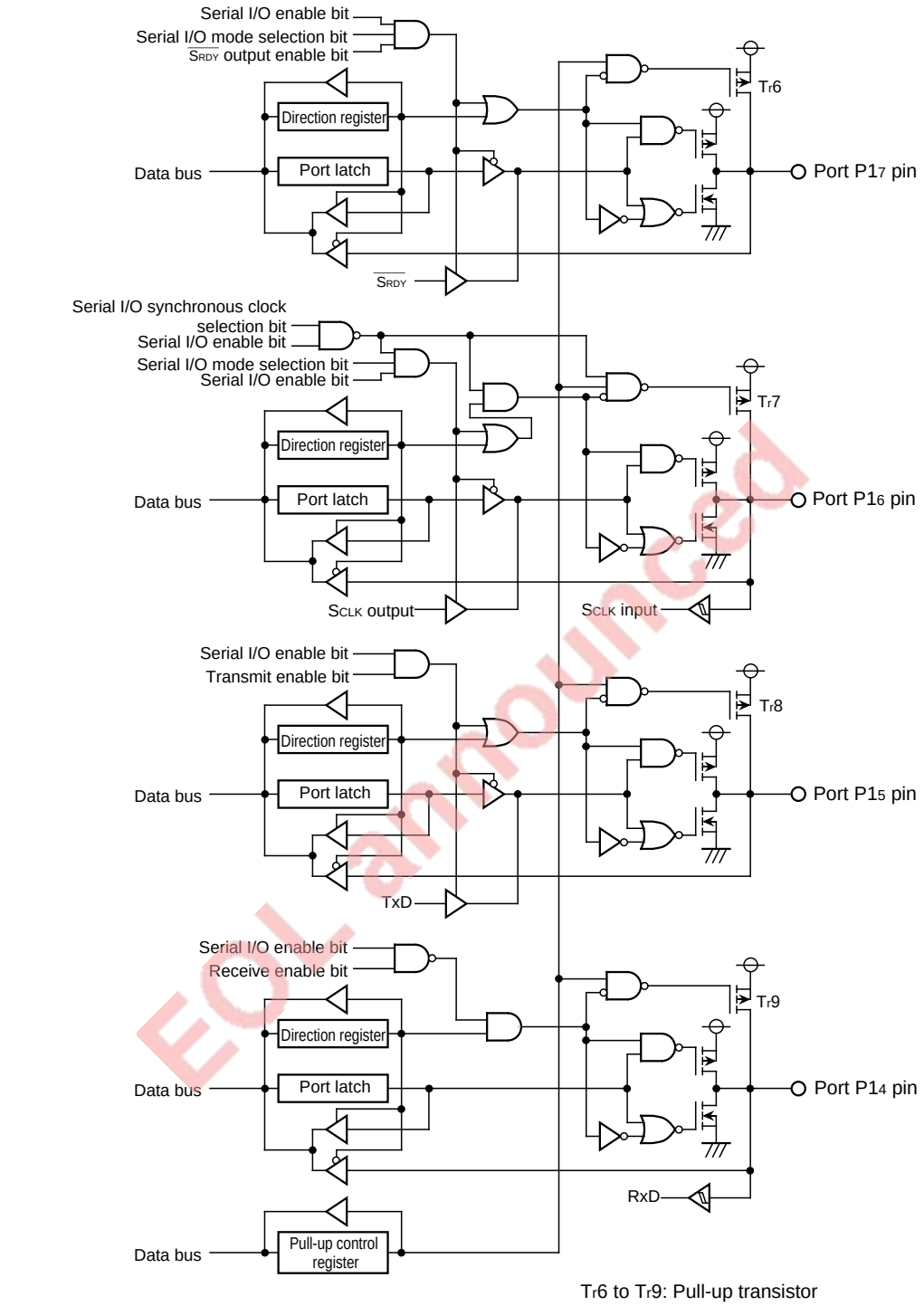
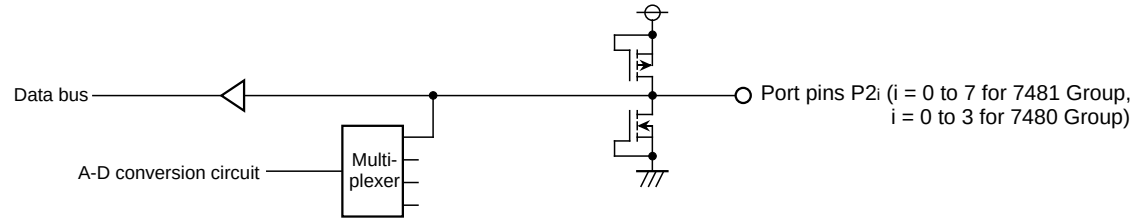
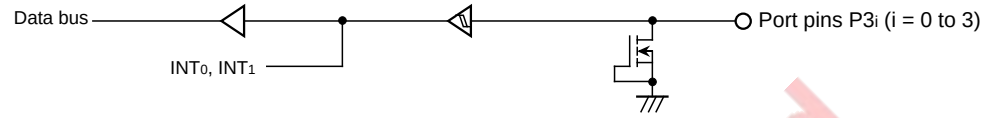


Figure 1.10.2 Block Diagram of Port Pins P14–P17

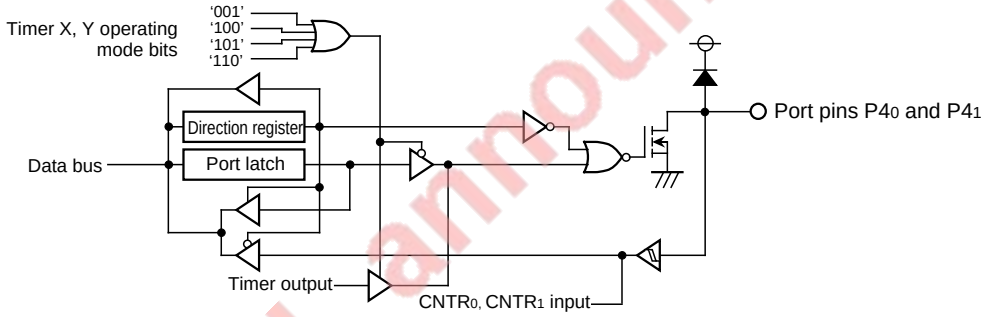
● Port P2



● Port P3



● Ports P40 and P41



● Ports P42, P43, and P5

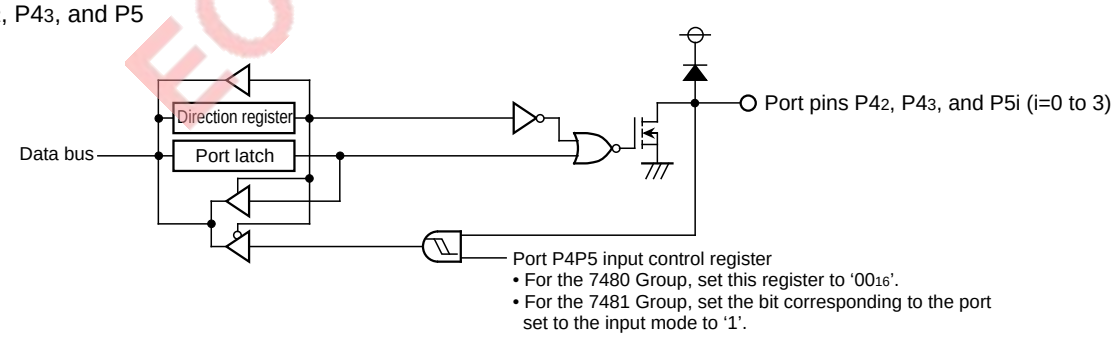


Figure 1.10.3 Block Diagrams of Port Pins P2i to P5i

1.10 Input/Output Pins

1.10.2 Registers Associated with I/O Pins

Figure 1.10.4 shows the memory map of the registers associated with I/O pins.

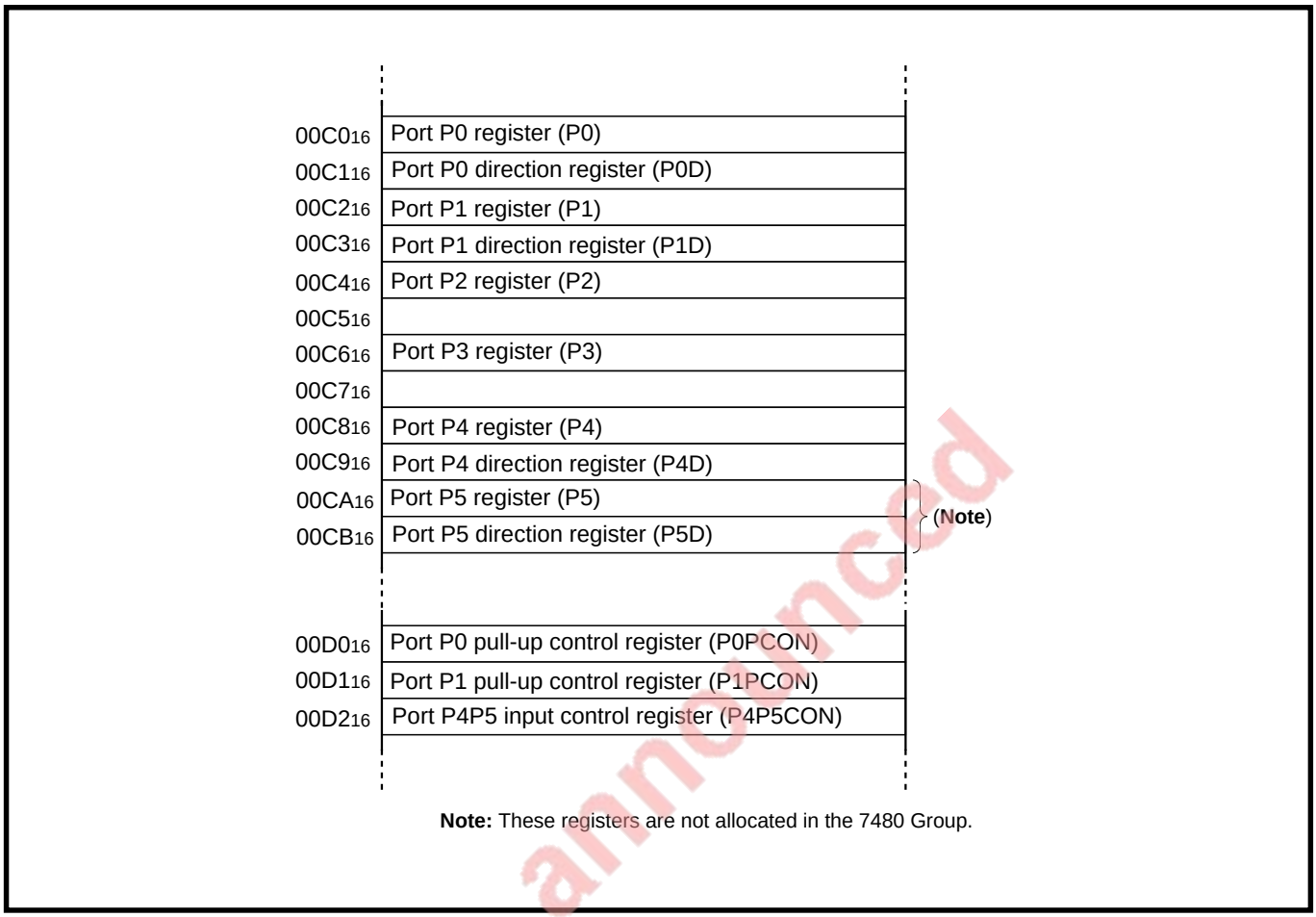


Figure 1.10.4 Memory Map of Registers Associated with I/O Pins

(1) Port Pi Registers (i = 0 to 5)

Each port register can read the states of the port pins and specify the output levels of them.

■Pin used for input (Ports P0 to P5)

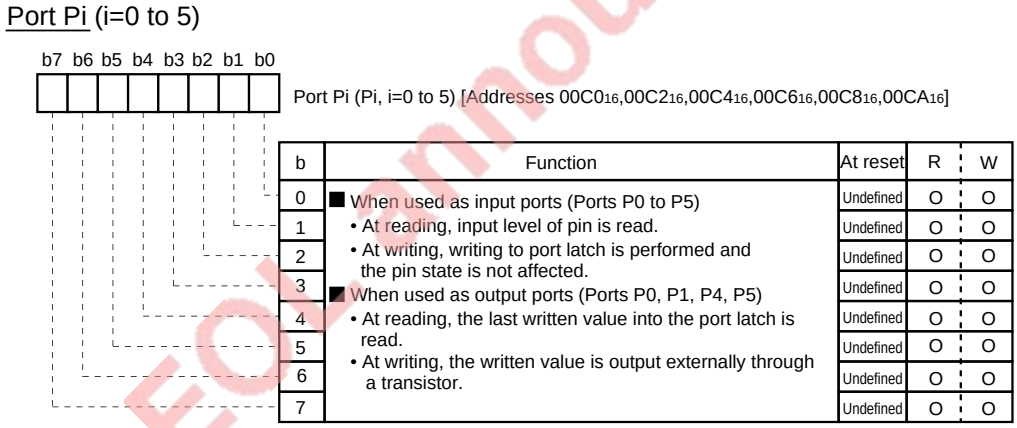
- A read: When reading from port register corresponding to each port, the input value (state of the pin) is read; the contents of port latch is not read.
- A write: When writing to port register corresponding to each port, data is written only into the port latch; the state of the pin is unaffected.

■Pin used for output (Ports P0, P1, P4, and P5)

- A read: When reading from port register corresponding to each port, the written value into the port latch is read; the state of the pin is not read. Therefore, even if the output voltage is affected by the external load etc., the last output value can correctly be read.
- A write: When writing to port register corresponding to each port, data written into a bit of the port register can be output to the external circuit through the output transistor.

Note: The 7480 Group does not have port P5 and, consequently, is not provided with the port P5 register.

Figure 1.10.5 shows the port Pi registers (i = 0 to 5).



b	Function	At reset	R	W
0	■ When used as input ports (Ports P0 to P5) • At reading, input level of pin is read. • At writing, writing to port latch is performed and the pin state is not affected. ■ When used as output ports (Ports P0, P1, P4, P5) • At reading, the last written value into the port latch is read. • At writing, the written value is output externally through a transistor.	Undefined	O	O
1		Undefined	O	O
2		Undefined	O	O
3		Undefined	O	O
4		Undefined	O	O
5		Undefined	O	O
6		Undefined	O	O
7		Undefined	O	O

Note: • In the 7480 Group, port P2 has only bits 0 to 3. The other bits are not implemented. (undefined at reading).
• Port P3 has only bits 0 to 3. The other bits are not implemented ('0' at reading).
• In the 7480 Group, port P4 has only bits 0 and 1. In the 7481 Group, port P4 has only bits 0 to 3. The other bits are not implemented. (bits 4 to 7: '0' at reading, bits 2 and 3 in the 7480 Group: undefined).
• The 7480 Group does not have port P5. In the 7481 Group, port P5 has only bits 0 to 3. The other bits are not implemented. ('0' at reading).

Figure 1.10.5 Port Pi Registers (i = 0 to 5)

1.10 Input/Output Pins

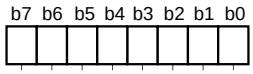
(2) Port Pi Direction Registers (i = 0, 1, 4, 5)

These registers switch the input and output of the programmable I/O port pins $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P4_0$–$P4_3$, and $P5_0$–$P5_3$.

Note: The 7480 Group does not have port P5 and, consequently, is not provided with the port P5 direction register.

Figure 1.10.6 shows the port Pi direction registers (i = 0, 1, 4, 5).

Port Pi direction register (i=0,1,4,5)



Port Pi direction register (PiD, i=0,1,4,5) [Addresses $00C1_{16}$,$00C3_{16}$,$00C9_{16}$,$00CB_{16}$]

b	Name	Function	At reset	R	W
0	Port Pi direction register (**Note**)	0 : Input mode 1 : Output mode	0	O	O
1		0 : Input mode 1 : Output mode	0	O	O
2		0 : Input mode 1 : Output mode	0	O	O
3		0 : Input mode 1 : Output mode	0	O	O
4		0 : Input mode 1 : Output mode	0	O	O
5		0 : Input mode 1 : Output mode	0	O	O
6		0 : Input mode 1 : Output mode	0	O	O
7		0 : Input mode 1 : Output mode	0	O	O

Note: • In the 7480 Group, port P4 direction register has only bits 0 and 1. In the 7481 Group, port P4 direction register has only bits 0 to 3. The other bits are not implemented (bits 4 to 7: '0' at reading, bits 2 and 3 in the 7480 Group: undefined).
• The 7480 Group does not have port P5 direction register. In the 7481 Group, port P5 has only bits 0 to 3. The other bits are not implemented ('0' at reading).

Figure 1.10.6 Port Pi Direction Registers (i = 0, 1, 4, 5)

(3) Port Pi Pull-up Control Registers (i = 0, 1)

When any pin of ports P0 and P1 is used for input, the corresponding bit of the port Pi pull-up control register controls the pull-up of the pin.
Pull-up control is performed by the ON/OFF switch of a pull-up transistor. Pull-up control is valid only when the pin is used for input and invalid when used for output or serial I/O.

Note: Port P1 controls the pull-up of high- or low-order four bits at one time. Even if only port $P1_0$ pin is pulled high, for example, port pins $P1_1$–$P1_3$ are also pulled high simultaneously.

Figures 1.10.7 and 1.10.8 show the port P0 and the port P1 pull-up control registers.

Port P0 pull-up control register

b7 b6 b5 b4 b3 b2 b1 b0

Port P0 pull-up control register (P0PCON) [Address $00D0_{16}$]

b	Name	Function	At reset	R	W
0	$P0_0$ pull-up control bit	0 : $P0_0$ No pull-up 1 : $P0_0$ Pull-up	0	O	O
1	$P0_1$ pull-up control bit	0 : $P0_1$ No pull-up 1 : $P0_1$ Pull-up	0	O	O
2	$P0_2$ pull-up control bit	0 : $P0_2$ No pull-up 1 : $P0_2$ Pull-up	0	O	O
3	$P0_3$ pull-up control bit	0 : $P0_3$ No pull-up 1 : $P0_3$ Pull-up	0	O	O
4	$P0_4$ pull-up control bit	0 : $P0_4$ No pull-up 1 : $P0_4$ Pull-up	0	O	O
5	$P0_5$ pull-up control bit	0 : $P0_5$ No pull-up 1 : $P0_5$ Pull-up	0	O	O
6	$P0_6$ pull-up control bit	0 : $P0_6$ No pull-up 1 : $P0_6$ Pull-up	0	O	O
7	$P0_7$ pull-up control bit	0 : $P0_7$ No pull-up 1 : $P0_7$ Pull-up	0	O	O

Note: Pull-up control is valid when the corresponding port is set to the input mode.

Figure 1.10.7 Port P0 Pull-up Control Register

Port P1 pull-up control register

b7 b6 b5 b4 b3 b2 b1 b0

Port P1 pull-up control register (P1PCON) [Address $00D1_{16}$]

b	Name	Function	At reset	R	W
0	$P1_3$–$P1_0$ pull-up control bit	0 : $P1_0$–$P1_3$ No pull-up 1 : $P1_0$–$P1_3$ Pull-up	0	O	O
1	$P1_7$–$P1_4$ pull-up control bit	0 : $P1_4$–$P1_7$ No pull-up 1 : $P1_4$–$P1_7$ Pull-up	0	O	O
2	Not implemented. Writing to these bits is disabled. These bits are undefined at reading.		Undefined	Undefined	×
3			Undefined	Undefined	×
4			Undefined	Undefined	×
5			Undefined	Undefined	×
6			Undefined	Undefined	×
7			Undefined	Undefined	×

Note: Pull-up control is valid only when the corresponding port is set to the input mode.
When port pins $P1_5$–$P1_7$ are used as serial I/O pins, pull-up control of the corresponding port pins is invalid.

Figure 1.10.8 Port P1 Pull-up Control Register

1.10 Input/Output Pins

(4) Port P4P5 Input Control Register

When port pins $P4_2$, $P4_3$, and $P5_0$–$P5_3$ of the 7481 Group are used as input ports, set the corresponding bits of the port P4P5 input control register to '1'.

Note: The 7480 Group does not have port pins $P4_2$, $P4_3$, and $P5_0$–$P5_3$; therefore, set the port P4P5 input control register to '00_{16}'.

Figure 1.10.9 shows the port P4P5 input control register.

Port P4P5 input control register

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

Port P4P5 input control register (P4P5CON) [Address $00D2_{16}$]

b	Name	Function	At reset	R	W
0	$P4_2$, $P4_3$ input control bit	When the $P4_2$, $P4_3$ are used as the input port, set this bit to '1'.	0	O	O
1	P5 input control bit	When the P5 is used as the input port, set this bit to '1'.	0	O	O
2	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
3			0	0	×
4			0	0	×
5			0	0	×
6			0	0	×
7			0	0	×

Note: 7480 Group does not have port pins $P4_2$, $P4_3$ and P5, so set this register to '00_{16}'.

Figure 1.10.9 Port P4P5 Input Control Register

1.10.3 I/O Ports

(1) Writes to and Reads from I/O Port Pins

■Pin used for input (Ports P0 to P5)
- A read: When reading from port register corresponding to each port, the input value (state of the pin) is read; the contents of port latch is not read.
- A write: When writing to port register corresponding to each port, data is written only into the port latch; the state of the pin is unaffected.

■Pin used for output (Ports P0, P1, P4, and P5)
- A read: When reading from port register corresponding to each port, the written value into the port latch is read; the state of the pin is not read. Therefore, even if the output voltage is affected by the external load etc., the last output value can correctly be read.
- A write: When writing to port register corresponding to each port, data written into a bit of the port register can be output to the external circuit through the output transistor.

Figure 1.10.10 shows a write and a read of an I/O port pin.

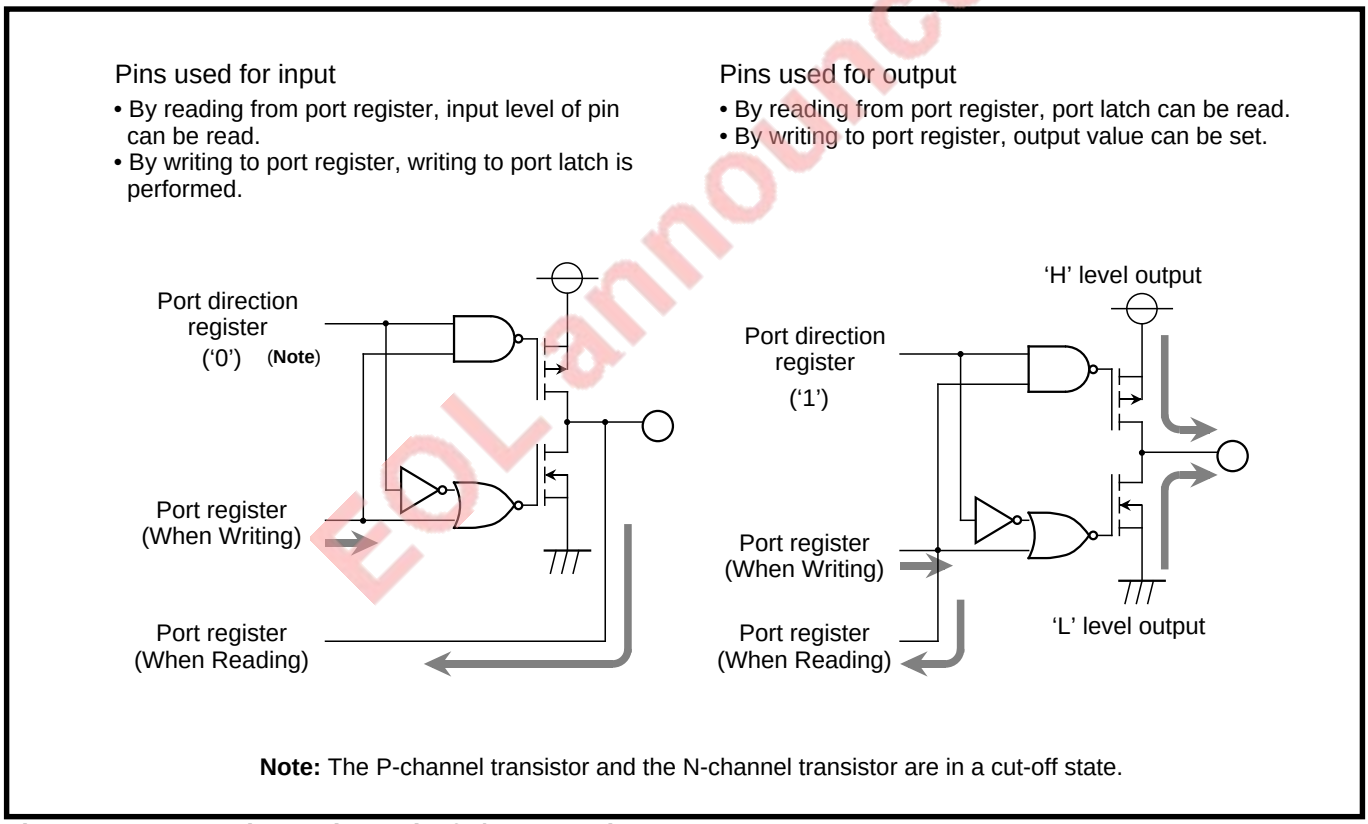


Figure 1.10.10 Write and Read of I/O Port Pin

1.10 Input/Output Pins

(2) Switching of Programmable I/O Port Pins

Any pin of the programmable I/O ports P0, P1, P4, and P5 can be switched from input to output or from output to input with the corresponding bit of their port direction registers.

- The pin is set to the input mode when the corresponding bit is '0'.
- The pin is set to the output mode when the corresponding bit is '1'.

Notes 1: In the 7480 Group, port P4 contains pins $P4_0$ and $P4_1$ only, while in the 7481 Group, port P4 contains pins $P4_0$–$P4_3$. In addition, the 7480 Group does not have port P5, while the 7481 Group has port P5, consisting of $P5_0$–$P5_3$.

2: After system is released from reset, all of the programmable I/O port pins are set to the input mode. (The corresponding direction registers are cleared to all '0'.)

3: When any of port pins $P4_2$, $P4_3$, and $P5_0$–$P5_3$ is used as an input port pin, clear the corresponding bit of the port P4 and P5 direction registers to '0'. In addition, set the corresponding input control bit of the port P4P5 input control register to '1'.
The 7480 Group does not have port pins $P4_2$, $P4_3$, and $P5_0$–$P5_3$; therefore, set the port P4P5 input control register to '00_{16}'.

(3) Pull-up Control

When any pin of ports P0 and P1 is used as an input port pin, its pull-up can be controlled with the corresponding bit of the port P0 and P1 pull-up control registers.

- A port pin is not pulled high when the corresponding bit is '0'.
- A port pin is pulled high when the corresponding bit is '1'.

Pull-up control is performed by the ON/OFF switch of a pull-up transistor. Pull-up control is valid only when the pin is used for input and invalid when used for output or serial I/O.

Note: Port P0 controls the pull-up by the bit at one time. Port P1, however, controls the pull-up of the high- or low-order four bits at one time. Even if only port $P1_0$ pin is pulled high, for example, port pins $P1_1$–$P1_3$ are also pulled high simultaneously.

(4) Level Shift Ports

Every pin of ports P4 and P5 acts as an N-channel open-drain output and is provided with a built-in clamping diode.

When voltage V_I is applied to a pin through a resistor as shown in Figure 1.10.11, and the current I flowing in a clamping diode is 1 mA or less, the condition $V_I > V_{CC}$ can be maintained.

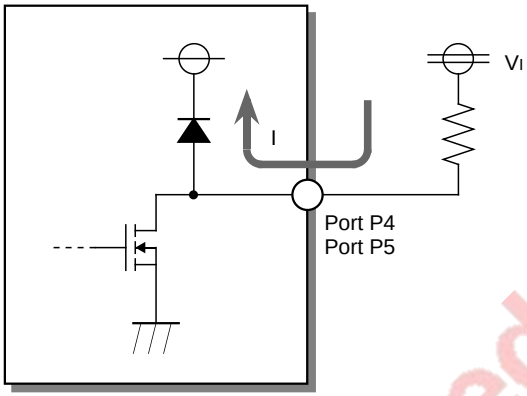


Figure 1.10.11 Port P4 and P5 Circuit

Notes 1: In the 7480 Group, port P4 contains pins $P4_0$ and $P4_1$ only, while in the 7481 Group, port P4 contains pins $P4_0$–$P4_3$. In addition, the 7480 Group does not have port P5, while the 7481 Group has port P5, consisting of $P5_0$–$P5_3$.

2: Total Input Current

It is required to keep the current flowing to the clamping diodes of port P4 or P5 equal to or less than 1.0 mA a pin; a current which is too large for the microcomputer can handle will raise the voltage on the power source pin. To protect the device, use an appropriate power circuit to stabilize the power source voltage within specifications.

3: Maximum Input Voltage

If the input signal voltage to port P4 or P5 exceeds V_{CC} + 0.3 V, a delay time of 2 μs/V or more is necessary immediately after the input waveform exceeds the above voltage. Delay time can be calculated by the following expression in CR integrating circuits.

$$\frac{dt}{dv} = \frac{t^{*1}}{0.6 \times V_{IN}{}^{*2}} \geq 2 \times 10^{-6}\ [s/V]$$

∗**1:** Delay time t = C × R

∗**2:** V_{IN} = maximum amplitude difference of input voltage

4: Clamping diodes used in the 7480 Group and 7481 Group differ from the normal switching diodes. These clamping diodes are used only for the DC signal level shifts. Therefore, sudden stress, such as a rush current must not be applied directly to the diodes.

(5) Ports with Built-in Schmidt Trigger Circuits

A Schmidt trigger circuit is built into every pin of ports P3, P4, and P5 of the 7481 Group.

When any of port pins $P4_2$, $P4_3$, and $P5_0$–$P5_3$ is used as an input port pin, clear the corresponding bit of the port P4 and P5 direction registers to '0'. In addition, set the corresponding input control bit of the port P4P5 input control register to '1'.

Note: The 7480 Group does not have pins $P4_2$, $P4_3$, and $P5_0$–$P5_3$; therefore, set the port P4P5 input control register to '00_{16}'.

1.10 Input/Output Pins

1.10.4 Termination of Unused Pins

Table 1.10.1 lists the termination of unused pins.

Table 1.10.1 Termination of Unused Pins

Port	Termination				
	Open	Pull-up to V_{CC} through a resistor (**Note 1**)	Pull-down to V_{SS} through a resistor (**Note 1**)	Connect to V_{CC}	Connect to V_{SS}
P0 $P1_0$–$P1_3$ $P1_5$, $P1_7$	O (**Note 2**)	O (**Note 3**)	O (**Note 4**)	×	×
$P1_4$, $P1_6$	O (**Note 5**)	O (**Note 3**)	O (**Note 4**)	×	×
P2 (**Note 8**)	O (**Note 2**)	O (**Note 6**)	O (**Note 6**)	O (**Note 6**)	O (**Note 6**)
P3	×	O (**Note 6**)	O (**Note 6**)	O (**Note 6**)	O (**Note 6**)
P4, P5 (**Note 8**)	O (**Note 5**)	O (**Note 3**)	O (**Note 7**)	×	×
AV_{SS} (**Note 9**)	×	×	×	×	O
V_{REF}	×	×	×	O	×
X_{OUT}	O	×	×	×	×

Notes 1: Do not connect several pins of programmable I/O ports together to V_{CC} or V_{SS} through a resistor.

2: Every pin that is allowed to be open when unused has a special circuit structure which prevents currents from flowing into the circuit unless the input to read signal is performed internally, even if an intermediate level input is applied to the pin.

3: When these pins are pulled high, set the corresponding bits of port direction registers, port registers and port P4P5 input control register so that each pin is in the input mode or output is HIGH.

4: When these pins are pulled low, set the corresponding bits of port direction registers, pull-up control registers, port registers so that each pin can be set to the input mode without pull-up transistor or in the output is LOW.

5: Set these pins to the output mode and keep them open.
However, these I/O pins retain the input mode until the pins are switched to the output mode by software after the microcomputer is released from reset. Therefore, the power source current may increase depending on the input level of each pin.
Since their port direction registers might be switched into the input mode by program runaway or noise, periodically set the port direction registers to the output mode by software.

6: A short wire can be used to directly connect any unused pin to the V_{CC} or V_{SS} pin without a resistor, but a long wire must connect it through a resistor. Since the $P3_3$ pin of the built-in PROM version has the alternative function of the V_{PP} pin, connect the $P3_3$ pin to V_{CC} or V_{SS} with the shortest wire through a resistor (about 5 kΩ), in series.

7: When these pins are pulled low, set the corresponding bits of port direction registers, port registers and port P4P5 input control register so that each pin is in the input mode or output is LOW.

8: The 7480 Group does not have port pins $P2_4$–$P2_7$, $P4_2$, $P4_3$, and $P5_0$–$P5_3$.

9: The AV_{SS} pin is dedicated to the 44P6N-A package products in the 7481 Group.

1.10.5 Notes on Usage

Pay attention to the following notes when the I/O ports are used.

(1) Rewriting to Port Register of I/O Port

Rewriting to the port register of an I/O port with a bit manipulation instruction*1 may affect the values of the bits not specified.

*1: 'Bit manipulation instructions': the **CLB** and **SEB** instructions

REASON: The bit manipulation instructions are read-modify-write instructions. These instructions read and write data by the byte. Therefore when these instructions are executed for any one bit of the port register of an I/O port, the following processing is performed to all of the bits of the register:

- For bits that are set to the input mode, the states of the corresponding pins are read into the CPU, and after the bit manipulation, the bits of the port register are rewritten.
- For bits that are set to the output mode, the values of the port latches are read into the CPU, and after the bit manipulation, the bits of the port register are rewritten.

Pay attention also to the following:

- Even if a port pin set to the output mode is switched to the input mode, output data is retained in the port register.
- When the state of a port pin and the content of the corresponding bit of the port register are different, the content of the bit of the port register set to the input mode may be affected even if this bit is not specified by a bit manipulation instruction.

(2) Pull-up Control of Ports P0 and P1

■ When any of pins $P1_5$–$P1_7$ is used as a serial I/O pin, the pull-up control of the pin is invalid. (The pin cannot be pulled high.)
For details, refer to **Figure 1.10.2 Block Diagram of Port Pins $P1_4$–$P1_7$.**

■ When any pin of ports P0 and P1 is used as an output port pin, the pull-up control of the pin is invalid. (The pin cannot be pulled high.)
For details, refer to **Figure 1.10.1 Block Diagrams of Port Pins $P0_i$ and $P1_0$–$P1_3$** and **Figure1.10.2 Block Diagram of Port Pins $P1_4$–$P1_7$.**

■ Port P1 controls pull-up the high- or low-order four bits at one time. Even if only port $P1_0$ pin is pulled high, for example, port pins $P1_1$–$P1_3$ are pulled high simultaneously.

(3) Transition to Standby State (Note)

At the transition to the standby state, do not leave the input levels of input port pins and I/O port pins undefined (especially pins $P1_4$, $P1_6$, P3, P4, and P5). In an N-channel open-drain I/O pin, when the corresponding bit of the port register is '1', its transistor remains in an off state even if the pin is set to output mode with the port direction register. As a result, the pin goes to a high impedance state, causing the level of the pin to be undefined depending on the external circuit. In such a case, a through current flows to the gate of the input stage, so that the power source current may increase.

Note: The standby state means the following:
The stop mode by an execution of the **STP** instruction
The wait mode by an execution of the **WIT** instruction.

Actual Example

Pull a pin high (connect to V_{CC}) or low (connect to V_{SS}) through a resistor.
Choose a resistor taking the following into consideration:
• External circuit condition
• Variation of output levels at normal operation
Also, take account of the variation of current when pull-up transistors of ports P0 and P1 are used.

(4) Usage of Pins $P1_2$, $P1_3$, $P4_0$, and $P4_1$ as Normal Output Pins

Pins $P1_2$ and $P1_3$ have the alternative functions of the 8-bit timer output pins T_0 and T_1 respectively. Pins $P4_0$ and $P4_1$ also have the alternative functions of 16-bit timer I/O pins $CNTR_0$ and $CNTR_1$ respectively. When the operating mode bits of the corresponding timer are set to any mode related to output (**Note**), these pins cannot operate as normal output pins. Refer to **Figure 1.10.1 Block Diagrams of Port Pins $P0_i$ and $P1_0$–$P1_3$** and **Figure 1.10.3 Block Diagrams of Port Pins $P2_i$ to $P5_i$.**

Note: Modes related to output:
For 8-bit timers (timer 1 and timer 2):
• Programmable waveform generation mode
For 16-bit timers (timer X and timer Y):
• Pulse output mode
• Programmable waveform generation mode
• Programmable one-shot output mode
• PWM mode

(5) Usage of Port Pins $P4_2$, $P4_3$, and $P5_0$–$P5_3$ as Input Ports

When any of port pins $P4_2$, $P4_3$, and $P5_0$–$P5_3$ of the 7481 Group are used as an input port pin, clear the corresponding bit of the port P4 and P5 direction registers to '0', and set the corresponding input control bit of the port P4P5 input control register to '1'.

1.11 Interrupts

The interrupt function is used to suspend the routine being executed by any interrupt source and to execute another routine. An interrupt is used in the following cases:

- When processing of a higher priority than the routine being executed is requested.
- When processing is requested to be performed according to a special timing.

Table 1.11.1 lists the interrupt sources available in the 7480 Group and 7481 Group.

Table 1.11.1 Interrupt Sources

Priority Order	Interrupt Source	Vector Address		Comments
		High-order	Low-order	
1	Reset (**Note 1**)	FFFF16	FFFE16	Non-maskable (**Note 2**)
2	INT0	FFFD16	FFFC16	External interrupt (Polarity programmable)
3	INT1	FFFB16	FFFA16	External interrupt (Polarity programmable)
	Key-on Wakeup			External interrupt
4	CNTR0	FFF916	FFF816	External interrupt (Polarity programmable)
5	CNTR1	FFF716	FFF616	External interrupt (Polarity programmable)
6	Timer X	FFF516	FFF416	Internal interrupt
7	Timer Y	FFF316	FFF216	Internal interrupt
8	Timer 1	FFF116	FFF016	Internal interrupt
9	Timer 2	FFEF16	FFEE16	Internal interrupt
10	Serial I/O Receive	FFED16	FFEC16	Internal interrupt
11	Serial I/O Transmit	FFEB16	FFEA16	Internal interrupt
12	Bus arbitration	FFE916	FFE816	Internal interrupt
13	A-D conversion complete	FFE716	FFE616	Internal interrupt
14	**BRK** instruction	FFE516	FFE416	Non-maskable software interrupt

Notes 1: Reset is included in the above table, as well, because it performs the same operation as interrupts.

2: 'Non-maskable interrupt':
this is the interrupt not having the corresponding interrupt request bit and interrupt enable bit. This interrupt request is accepted regardless of the state of the interrupt disable flag.

1.11 Interrupts

1.11.1 Block Diagram

Figure 1.11.1 shows the block diagram of the interrupt inputs and the key-on wakeup circuit.

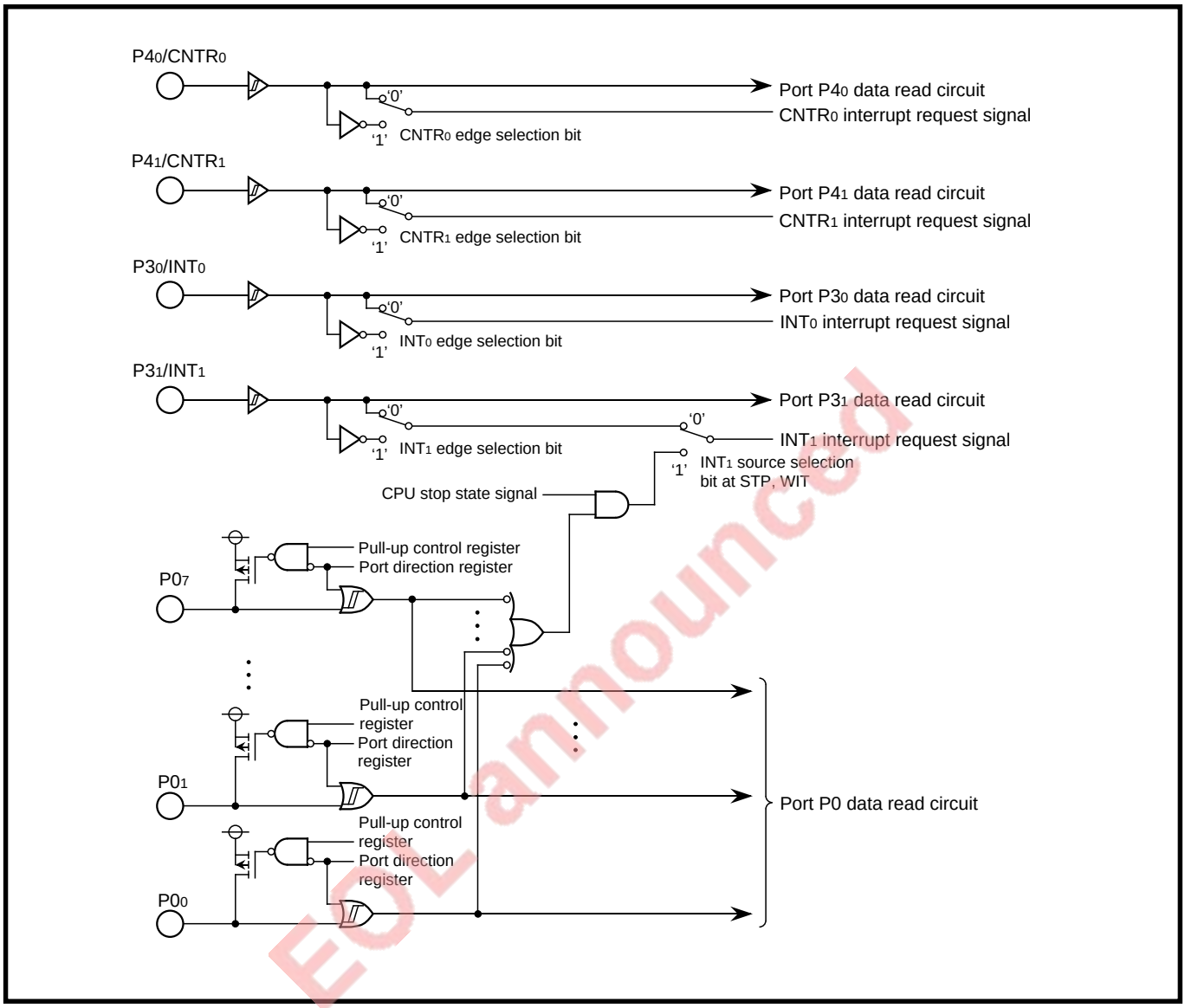


Figure 1.11.1 Block Diagram of Interrupt Inputs and Key-On Wakeup Circuit

1.11.2 Registers Associated with Interrupt Control

Figure 1.11.2 shows the memory map of the registers associated with interrupt control.

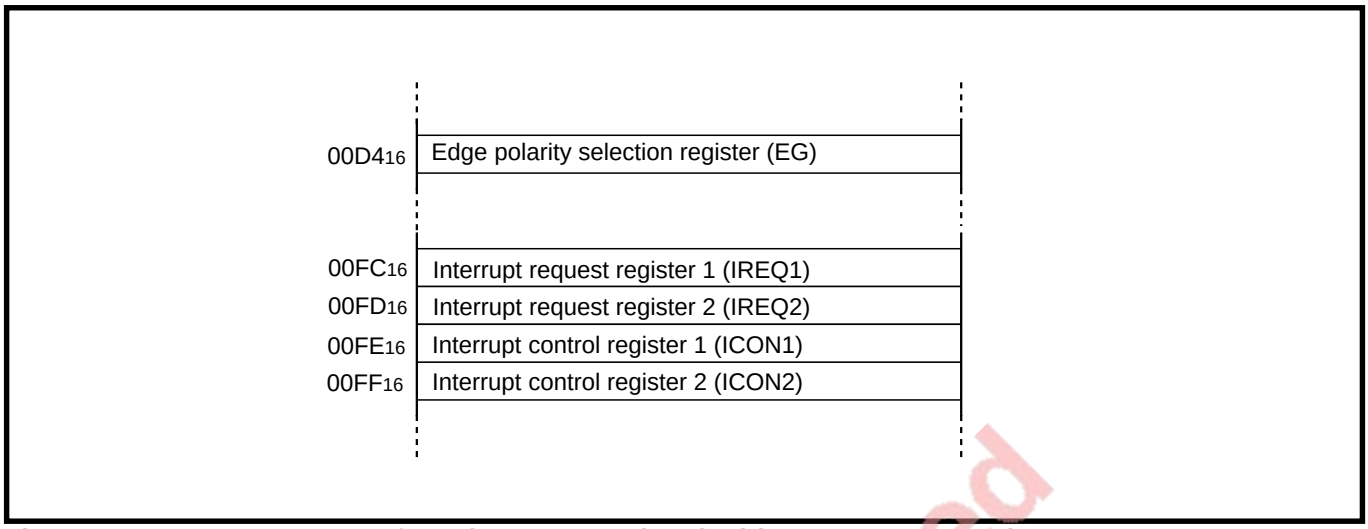


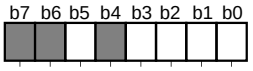
Figure 1.11.2 Memory Map of Registers Associated with Interrupt Control

1.11 Interrupts

(1) Edge Polarity Selection Register

The edge polarity selection register consists of the bits that select the polarity of the valid edge of the INT and CNTR pins, as well as the bit that selects the valid/invalid of the key-on wakeup. Figure 1.11.3 shows the edge polarity selection register.

Edge polarity selection register



Edge polarity selection register (EG) [Address $00D4_{16}$]

b	Name	Function	At reset	R	W
0	INT_0 edge selection bit	0 : Falling edge 1 : Rising edge	0	O	O
1	INT_1 edge selection bit	0 : Falling edge 1 : Rising edge	0	O	O
2	$CNTR_0$ edge selection bit	0: In event count mode, rising edge counted. : In pulse output mode, operation started at HIGH level output. : In pulse period measurement mode, a period from falling edge until falling edge measured. : In pulse width measurement mode, HIGH-level period measured. : In programmable one-shot output mode, one-shot HIGH pulse generated after operation started at LOW level output. : Interrupt request is generated by detecting falling edge. 1: In event count mode, falling edge counted : In pulse output mode, operation started at LOW level output. : In pulse period measurement mode, a period from rising edge until rising edge measured. : In pulse width measurement mode, LOW-level period measured. : In programmable one-shot output mode, one-shot LOW pulse generated after operation started at HIGH level output. : Interrupt request is generated by detecting rising edge.	0	O	O
3	$CNTR_1$ edge selection bit		0	O	O
4	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
5	INT_1 source selection bit at STP or WIT	0 : $P3_1/INT_1$ 1 : $P0_0$–$P0_7$ LOW level (for key-on wake-up)	0	O	O
6	Not implemented. Writing to these bits are disabled. These bits are undefined at reading.		Undefined	Undefined	×
7			Undefined	Undefined	×

Note: When setting bits 0 to 3, the interrupt request bit may be set to '1'. After setting the following, enable the interrupt.
① Disable interrupts
② Set the edge polarity selection register
③ Set the interrupt request bit to '0'

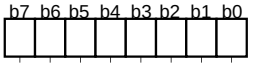
Figure 1.11.3 Edge Polarity Selection Register

(2) Interrupt Request Register 1 and Interrupt Request Register 2

Interrupt request registers 1 and 2 consist of the bits that indicate whether or not there is an interrupt request.

Figures 1.11.4 and 1.11.5 show the interrupt request registers 1 and 2.

Interrupt request register 1



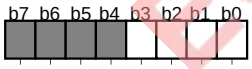
Interrupt request register 1 (IREQ1) [Address 00FC16]

b	Name	Function	At reset	R	W
0	Timer X interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
1	Timer Y interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
2	Timer 1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
3	Timer 2 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
4	Serial I/O receive interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
5	Serial I/O transmit interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
6	Bus arbitration interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
7	A-D conversion completion interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*

*: The bit can be set to '0' by software, but cannot be set to '1'.

Figure 1.11.4 Interrupt Request Register 1

Interrupt request register 2



Interrupt request register 2 (IREQ2) [Address 00FD16]

b	Name	Function	At reset	R	W
0	INT0 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
1	INT1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
2	CNTR0 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
3	CNTR1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
4	Not implemented. Writing to these bits is disabled. These bits are undefined at reading.		Undefined	Undefined	×
5			Undefined	Undefined	×
6			Undefined	Undefined	×
7			Undefined	Undefined	×

*: The bit can be set to '0' by software, but cannot be set to '1'.

Figure 1.11.5 Interrupt Request Register 2

(3) Interrupt Control Register 1 and Interrupt Control Register 2

Interrupt control registers 1 and 2 consist of the bits that control the acceptance of interrupts. Figures 1.11.6 and 1.11.7 show the interrupt control registers 1 and 2.

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1 (ICON1) [Address 00FE16]

b	Name	Function	At reset	R	W
0	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
1	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
2	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
3	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
4	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
5	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
6	Bus arbitration interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
7	A-D conversion completion interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O

Figure 1.11.6 Interrupt Control Register 1

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 2 (ICON2) [Address 00FF16]

b	Name	Function	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
1	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
2	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
3	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
4	Not implemented. Writing to these bits are disabled. These bits are undefined at reading.		Undefined	Undefined	×
5			Undefined	Undefined	×
6			Undefined	Undefined	×
7			Undefined	Undefined	×

Figure 1.11.7 Interrupt Control Register 2

1.11.3 Interrupt Sources

In the 7480 Group and 7481 Group, the interrupt requests can be generated by 14 sources (5 external, 8 internal, and 1 software).

The interrupts are vectored interrupts whose priority levels are fixed, and each interrupt has its own priority level. When two or more interrupt requests are generated at the same sampling time, which is a timing to test the generation of interrupt requests, the interrupt with a higher priority is acceptable.

For the priority levels of interrupts, refer to **Table 1.11.1 Interrupt Sources**.

Each interrupt source is described below.

(1) INT_0 and INT_1 Interrupts

When a rising edge or a falling edge of the input signal to the INT_0 or INT_1 pin is detected, an interrupt request is generated.

The edge polarity to be detected can be selected by the INT_0 edge selection bit or the INT_1 edge selection bit of the edge polarity selection register.

The request bit, the enable bit, and the interrupt vector of the INT_1 interrupt have the alternative functions of those of the key-on wakeup interrupt respectively. When the INT_1 interrupt is used, clear the INT_1 source selection bit at the STP/WIT of the edge polarity selection register to '0'.

- State after system is released from reset

  After system is released from reset, the INT_0 edge selection bit, INT_1 edge selection bit and the INT_1 source selection bit at the STP/WIT of the edge polarity selection register are all cleared to '0'.

  In such conditions, though an interrupt request is generated by detecting a falling edge of the INT_0 or INT_1 pin, the interrupt request cannot be accepted because the corresponding interrupt enable bit is '0' and the interrupt disable flag is '1'.

Notes 1: The INT_0 and INT_1 pins have the alternative functions of input port pins $P3_0$ and $P3_1$, respectively. When these pins are used as input port pins, valid edges can still be detected because the 7480 Group and 7481 Group does not have the function to switch the INT pins to input port pins. Therefore, when these pins are used as input port pins, clear all the corresponding interrupt enable bits to '0' (disabled).

2: Keep the trigger width input to the INT pins 250 ns or more.

(2) Key-On Wakeup Interrupt

When the INT_1 source selection bit at the STP/WIT of the edge polarity selection register is '1' and the LOW level is applied to any pin of port P0 which is used as input in the stop/wait mode at the execution of **STP/WIT**, a key-on wakeup interrupt request is generated. In other states than the stop/wait mode, the key-on wakeup interrupt is invalid.

The request bit, the enable bit, and the interrupt vector of the key-on wakeup interrupt have the alternative functions of those of the INT_1 interrupt respectively. When the key-on wakeup interrupt is used, set the INT_1 source selection bit at the STP/WIT of the edge polarity selection register to '1'.

Note: When the key-on wakeup interrupt is used, execute the **STP/WIT** instruction after all inputs to port P0 are held HIGH. If the LOW level is applied to any input pin of port P0, an execution of the **STP/WIT** instruction generates an interrupt request instantly.

1.11 Interrupts

(3) $CNTR_0$ and $CNTR_1$ Interrupts

When a rising edge or a falling edge of the input signal to the $CNTR_0$ or the $CNTR_1$ pin is detected, an interrupt request is generated. The edge polarity to be detected can be selected by the $CNTR_0$ edge selection bit or the $CNTR_1$ edge selection bit of the edge polarity selection register.

- State after system is released from reset
  After system is released from reset, the port pins with the alternative functions of CNTR pins are placed in the input mode, and their edge selection bits are held all '0' also. In such conditions, though an interrupt request is generated by detecting a falling edge of the $CNTR_0$ or $CNTR_1$ pin, the interrupt request cannot be accepted because the corresponding interrupt enable bit is '0' and the interrupt disable flag is '1'.

Note: The $CNTR_0$ and $CNTR_1$ pins have the alternative functions of I/O port pins $P4_0$ and $P4_1$, respectively. When these pins are used as input port pins, valid edges can still be detected because the 7480 Group and 7481 Group does not have the function to switch the CNTR pins to input port pins. Therefore, when these pins are used as input port pins, clear all the corresponding interrupt enable bits to '0' (disabled).

(4) Timer X, Timer Y, Timer 1, and Timer 2 Interrupts

At an underflow in each timer, the corresponding interrupt request is generated.
For timer X and timer Y, refer to **Section 1.12 Timer X and Timer Y,** and for timer 1 and timer 2, refer to **Section 1.13 Timer 1 and Timer 2.**

(5) Serial I/O Receive Interrupt, Serial I/O Transmit Interrupt, and Bus Arbitration Interrupt

- Serial I/O receive interrupt
  During serial I/O reception, a serial I/O receive interrupt request is generated when the received data stored completely in the receive shift register is transferred to the receive buffer register.

- Serial I/O transmit interrupt
  During serial I/O transmission, a serial I/O transmit interrupt request is generated when the transmit buffer register is emptied or the transmit shift operation is complete.

- Bus arbitration interrupt
  In the bus collision detection enable state during the serial I/O communication, the mismatch of levels between transmitter pin TxD and receiver pin RxD generates a bus arbitration interrupt request.
  The bus collision detection can be enabled by setting the bus collision detection enable bit of the bus collision detection control register.

For serial I/O, refer to **Section 1.14 Serial I/O**.

(6) A-D Conversion Complete Interrupt

When A-D conversion is completed, an A-D conversion complete interrupt request is generated.
For A-D conversion, refer to **Section 1.15 A-D Converter.**

(7) BRK Instruction Interrupt

The **BRK** instruction interrupt is a non-maskable software interrupt. Program branches to the jump address stored in the vector address when the **BRK** instruction is executed.
For the **BRK** instruction, refer to the section of the **BRK** instruction in **SERIES 740 <SOFTWARE> USER'S MANUAL**.

1.11.4 Interrupt Sequence

Interrupt sequence is described below.

Generation of Interrupt Requests

When an interrupt request other than the **BRK** instruction interrupt is generated, the interrupt request bit of the corresponding interrupt request register is set to '1'. At this time, the interrupt request is accepted when both the following conditions are satisfied:

- The interrupt enable bit of the corresponding interrupt control register is '1'.
- The interrupt disable flag of the processor status register is '0'.

When the **BRK** instruction interrupt request is generated, the break flag of the processor status register is set to '1', causing the interrupt request to be accepted unconditionally.

For interrupt sources, refer to **Section 1.11.3 Interrupt Sources**. Also for interrupt control, refer to **Section 1.11.5 Interrupt Control**.

Acceptance of Interrupt Request

When an interrupt request is accepted, the following operations are performed:

[1] Upon the completion of the instruction being executed, the processing is temporarily suspended.

[2] The contents of the program counter and the processor status register are pushed onto the stack in the following order:
① High-order 8 bits of the program counter
② Low-order 8 bits of the program counter
③ Processor status register

[3] The jump address (the start address of an interrupt service routine) stored in the vector address of the accepted interrupt is set in the program counter, and the interrupt service routine is executed. At this time, the interrupt disable flag is set to '1', and multiple interrupts are disabled. Also, the corresponding interrupt request bit is cleared to '0' for any interrupt other than the **BRK** instruction interrupt.

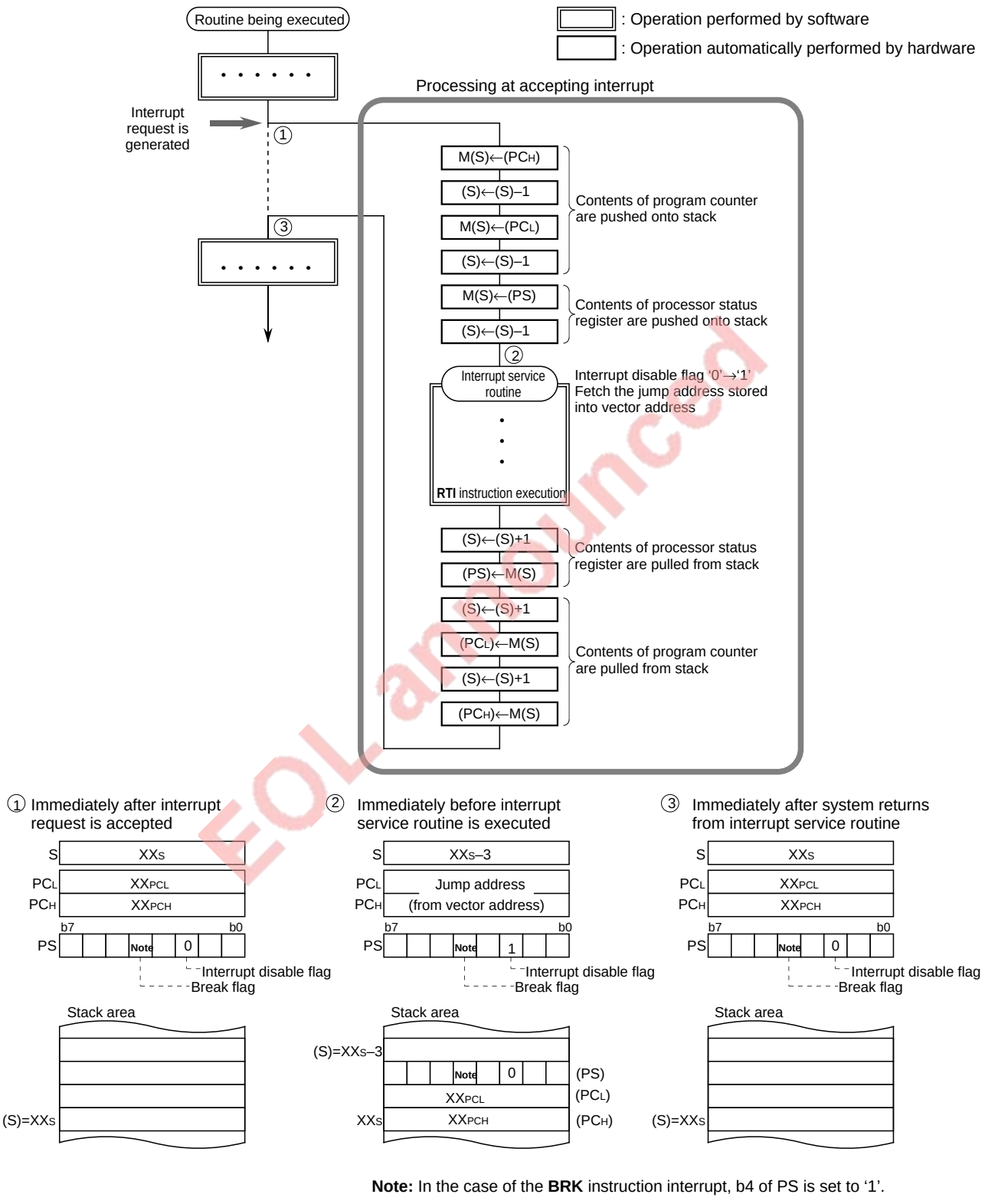
[4] When the **RTI** instruction, which is, the last instruction of the interrupt service routine, is executed, the contents of the program counter and the processor status register pushed onto the stack are pulled to the corresponding register in the following order:
① Processor status register
② Low-order 8 bits of program counter
③ High-order 8 bits of program counter

[5] The program temporarily suspended by the acceptance of the interrupt request is resumed at the address indicated by the program counter.

Note: When the **BRK** instruction is executed, 2 is added to the contents of program counter, and then the contents of the program counter are pushed onto the stack. As a result, upon return from the **BRK** instruction interrupt service routine, the one byte subsequent to the **BRK** instruction is not executed. Therefore, at programming, it is necessary to insert the **NOP** instruction immediately after the **BRK** instruction.

1.11 Interrupts

Figure 1.11.8 shows an operation when an interrupt request is accepted.



Note: In the case of the **BRK** instruction interrupt, b4 of PS is set to '1'.

Figure 1.11.8 Operation When Interrupt Request is Accepted

Processing Before Interrupt Service Routine

When an interrupt request is accepted, the interrupt service routine is started after the following are performed.

① the instruction being executed at the generation of the interrupt request is completed
② the pipeline postprocessing
③ the pushing onto the stack, and vector fetch

Figure 1.11.9 shows the processing time from the interrupt generation until the execution of an interrupt service routine, and Figure 1.11.10 shows a timing at interrupt acceptance.

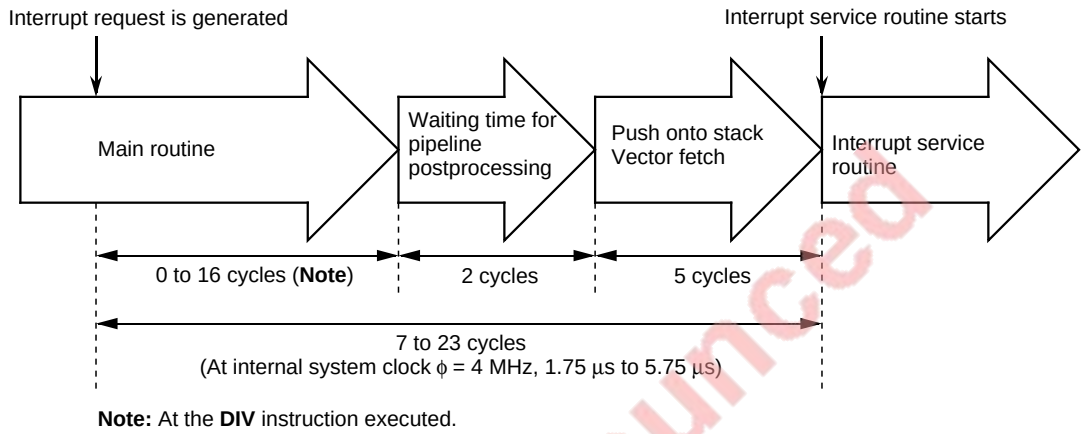


Figure 1.11.9 Processing Time from Interrupt Generation until Execution of Interrupt Service Routine

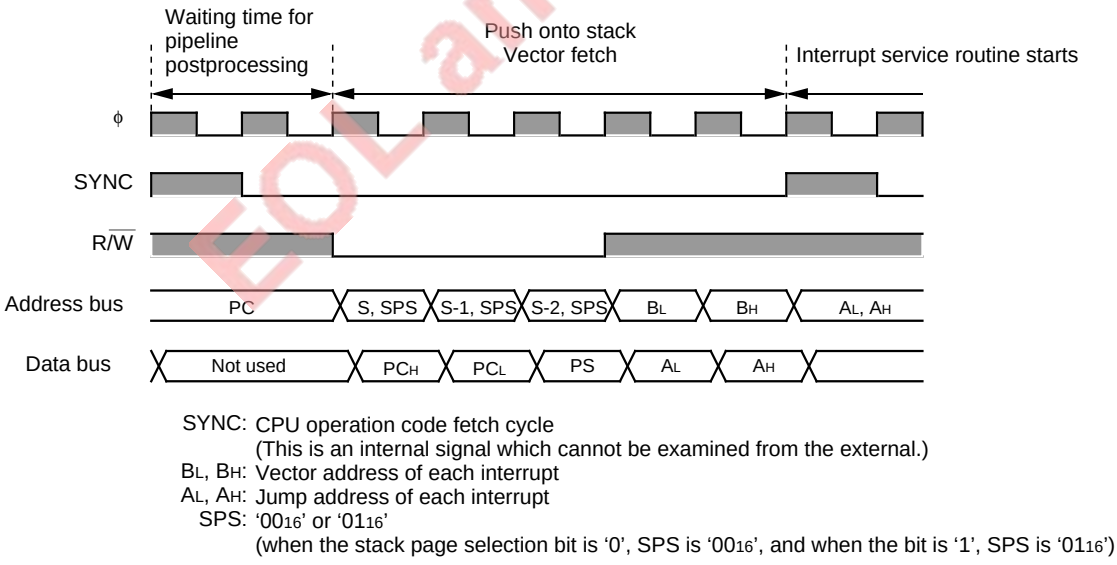


Figure 1.11.10 Timing at Interrupt Acceptance

1.11 Interrupts

Return from Stop/Wait Mode

When an interrupt request is accepted in the stop/wait mode, the CPU terminates these modes and returns to the normal mode.

Table 1.11.2 lists the interrupt sources available for CPU's return from the stop/wait mode.

Table 1.11.2 Interrupt Sources Available for CPU's Return from Stop/Wait Mode (O:Available, ×:Not available)

Interrupt Source	Return from Stop Mode	Return from Wait Mode
Reset (**Note 1**)	O	O
INT_0	O	O
INT_1	O	O
Key-on Wakeup	O	O
$CNTR_0$	O	O
$CNTR_1$	O	O
Timer X	O (**Note 2**)	O
Timer Y	O (**Note 2**)	O
Timer 1	×	O
Timer 2	×	O
Serial I/O Receive	O (**Note 3**)	O
Serial I/O Transmit	O (**Note 3**)	O
Bus Arbitration	O (**Note 3**)	O
A-D Conversion Complete	×	O
BRK Instruction	×	×

Notes 1: Reset is included in the above table, as well, because it performs the same operation as interrupts.
2: Available in the event count mode only.
3: Available only when the external clock input (or the clock divided by 16) is used as the synchronous clock.

For details, refer to **Section 1.19 Power Saving Function**.

1.11.5 Interrupt Control

Figure 1.11.11 shows an interrupt control diagram.

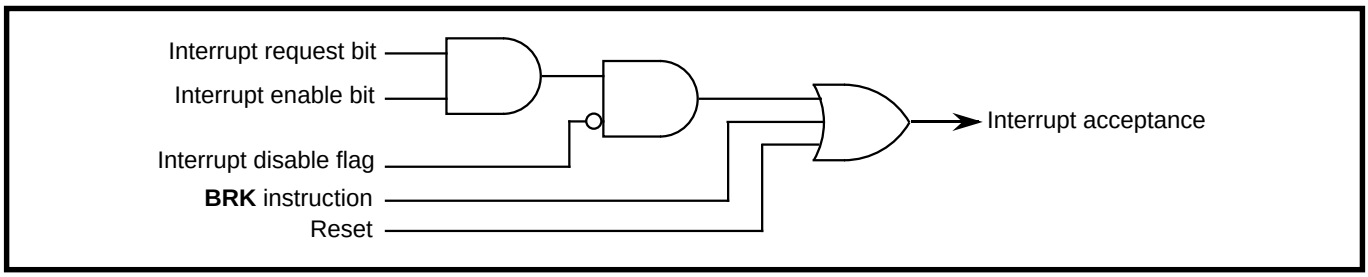


Figure 1.11.11 Interrupt Control Diagram

Only when all of the following conditions are satisfied, interrupts other than the **BRK** instruction interrupt are accepted:

- Corresponding interrupt request bit is '1' (interrupt requested).
- Corresponding interrupt enable bit is '1' (interrupt enabled).
- Interrupt disable flag is '0' (interrupt enabled).

The priority level of each interrupt is specified by hardware. However, processing of various priorities can be performed under software control by using the above bits and flag.
For the interrupt priority levels, refer to **Table 1.11.1 Interrupt Sources.**

Interrupt Request Bits

The interrupt request bits indicate whether or not there are interrupt requests. When an interrupt request is generated, an interrupt request bit is set to '1' and informs the external that the interrupt request is generated. After the interrupt is accepted, the interrupt request bit is automatically cleared to '0'.
The interrupt request bits can be cleared to '0' by software, but they cannot be set to '1'.

Interrupt Enable Bits

The interrupt enable bits control the acceptance of interrupt requests as follows:

- When an interrupt enable bit is '0', the acceptance of the corresponding interrupt request is disabled.
- When an interrupt enable bit is '1', the acceptance of the corresponding interrupt request is enabled.

Interrupt Disable Flag

This flag is located in the processor status register. The flag controls the acceptance of the interrupt requests other than the **BRK** instruction interrupt as follows:

- When the interrupt disable flag is '0', the acceptance of interrupt request is enabled.
- When the interrupt disable flag is '1', the acceptance of interrupt request is disabled.

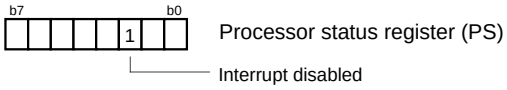
When the program branches to the interrupt service routine, this flag is automatically set to '1' and disables multiple interrupts. When multiple interrupts are used, clear this flag to '0' at the start of the interrupt service routine.

1.11 Interrupts

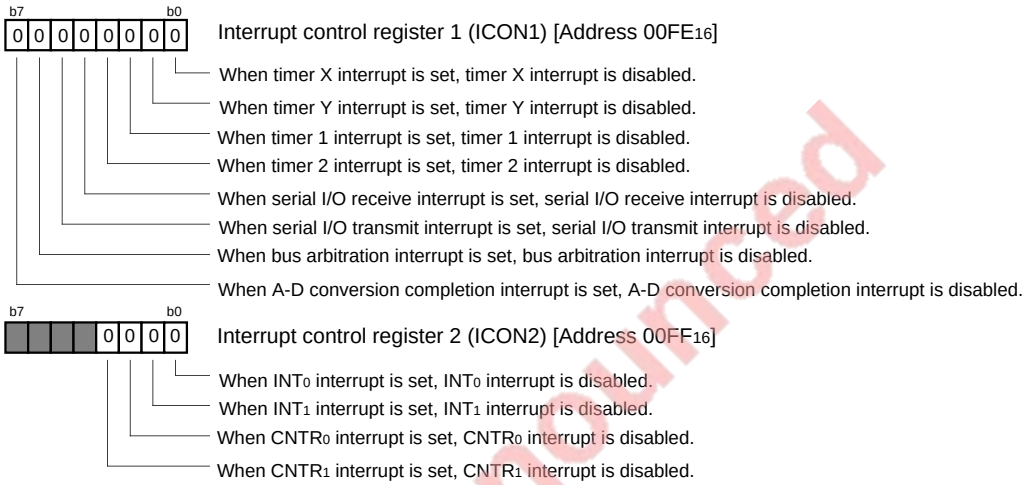
1.11.6 Setting of Interrupts

Figures 1.11.12 and 1.11.13 show the setting of interrupts.

Procedure 1 Setting interrupt disable flag to '1' to disable the acceptance of other interrupts during setting.



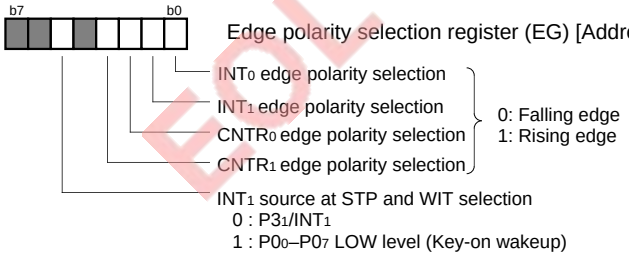
Procedure 2 Setting using interrupt enable bit to '0' (disabled)



Procedure 3 Setting each interrupt

• When INT interrupt, CNTR interrupt and key-on wakeup interrupt are used

1. Selection of edge polarity selection register



2. When CNTR interrupt and key-on wakeup interrupt are used, using port is set to input mode.
3. When key-on wakeup interrupt is used, using port pins are pulled high.

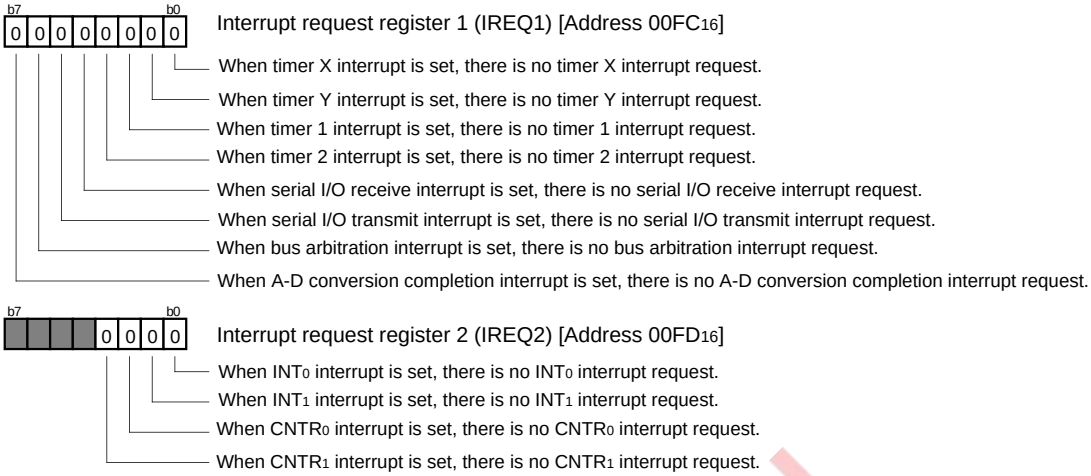
• When timer interrupt is used
1. Stop of timer count
2. Setting of each mode
3. Setting of timer (except pulse period measurement mode and pulse width measurement mode)

• When serial I/O receive interrupt, serial I/O transmit interrupt or bus arbitration interrupt are used
1. Setting of registers related to serial I/O
2. Setting of baud rate generator (only when internal clock is selected as synchronous clock)

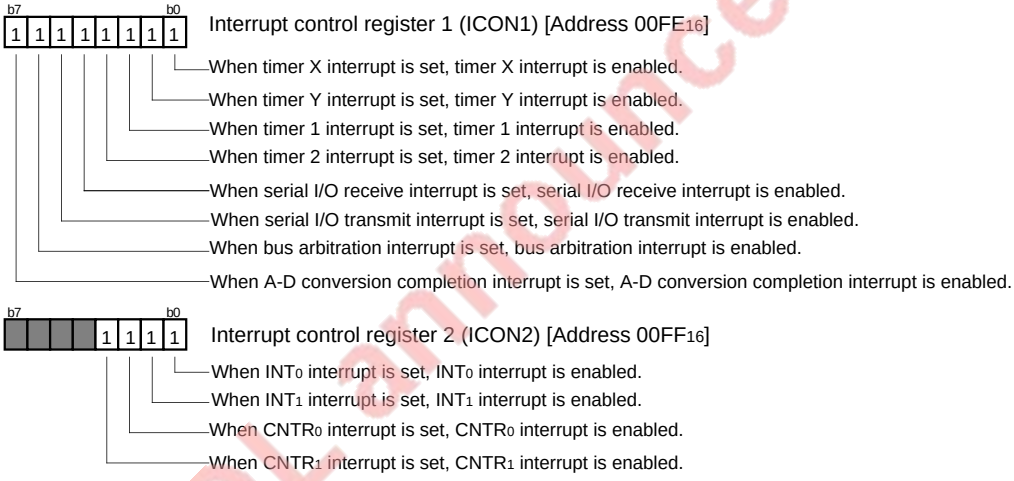
Note: For details, refer to setting of each function.

Figure 1.11.12 Setting of Interrupts (1)

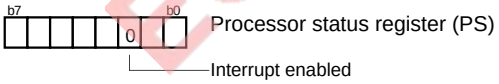
Procedure 4 Setting using interrupt request bit to '0' (no interrupt request)



Procedure 5 Using interrupt enable bit to '1' (enabled)



Procedure 6 Setting b2 of PS to '0' when the interrupt disable flag is set to '1' in procedure 1.



Procedure 7 Operate the function associated with each interrupt

- When key-on wakeup interrupt is used:
  System is set to enter the stop mode/wait mode with the **STP/WIT** instruction.
- When timer interrupt is used:
  Timer count start
- When serial I/O receive interrupt, serial I/O transmit interrupt and bus arbitration interrupt are used:
  Data is written to the transmit buffer register and transmit/receive start.
- When A-D conversion completion interrupt is used:
  Setting A-D control register (A-D conversion start)

Note: For details, refer to setting of each function.

Figure 1.11.13 Setting of Interrupts (2)

1.11 Interrupts

1.11.7 Notes on Usage

Pay attention to the following notes when an interrupt is used.

(1) For All Interrupts

- Before the execution of an interrupt, set the corresponding interrupt request bit and interrupt enable bit in the following order:
  ① Clear the interrupt request bit to '0' (no interrupt request).
  ② Set the corresponding interrupt enable bit to '1' (interrupt enabled).

- The interrupt request bits can be changed by software, but retain the values immediately after a rewrite instruction is executed. Therefore, the following operations must be performed after one or more instructions at the completion of a rewrite instruction:
  • Execute the **BBC** or **BBS** instruction after an interrupt request bit is changed.
  • Set an interrupt enable bit to '1' after an interrupt request bit is changed.

(2) For the INT and CNTR Interrupts

- When edge selection bits of the edge polarity selection register are set, interrupt request bits may become '1'. Therefore, set edge selection bits in the following sequence:
  ① Clear interrupt enable bit to '0' (interrupt disabled).
  ② Set edge selection bit.
  ③ Clear interrupt request bit to '0' (no interrupt request).
  ④ Execute one or more instructions (**NOP** etc.).
  ⑤ Set interrupt enable bit to '1' (interrupt enabled).

- The INT_0, INT_1, $CNTR_0$, and $CNTR_1$ pins have the alternative functions of input port pins $P3_0$, $P3_1$, I/O port pins $P4_0$, and $P4_1$, respectively. When these pins are used as input port pins, valid edges can still be detected because the 7480 Group and 7481 Group does not have the function to switch the INT and CNTR pins to input port pins. Therefore, when these pins are used as input port pins, clear all the corresponding interrupt enable bits of the INT and CNTR interrupts to '0' (disabled).

- Keep the trigger width input to the INT pins 250 ns or more.

(3) For the Key-On Wakeup Interrupt

- When the key-on wakeup interrupt is used, execute the **STP/WIT** instruction after all inputs to port P0 are held HIGH.

- In states other than the stop/wait mode, the key-on wakeup interrupt is invalid.

(4) For the BRK instruction interrupt

- When the **BRK** instruction is executed, 2 is added to the contents of program counter, and then the contents of the program counter are pushed onto the stack. As a result, upon return from the **BRK** instruction interrupt service routine, the one byte subsequent to the **BRK** instruction is not executed. Therefore, at programming, it is necessary to insert the **NOP** instruction immediately after the **BRK** instruction.

- When there are two or more interrupt sources of which interrupt request bits and interrupt enable bits are '1', but the interrupt disable flag is '1' (that is, in the interrupt disabled state), the execution of the **BRK** instruction starts execution of the interrupt service routine at the vector address with the highest priority level in these sources.

1.12 Timer X and Timer Y

The 7480 Group and 7481 Group have two 16-bit timers with 16-bit latches.
- Timer X
- Timer Y

Timer X or timer Y can select the following operation modes by the timer X or Y operation mode bits and the timer X or Y count source selection bits of the timer X mode register (address $00F6_{16}$) or the timer Y mode register (address $00F7_{16}$):
- Timer mode
- Event count mode
- Pulse output mode
- Pulse period measurement mode
- Pulse width measurement mode
- Programmable waveform generation mode
- Programmable one-shot output mode
- PWM mode

For details, refer to the section of each mode.

1.12.1 Block Diagram

Figure 1.12.1 shows the block diagram of timer X and timer Y.

1.12 Timer X and Timer Y

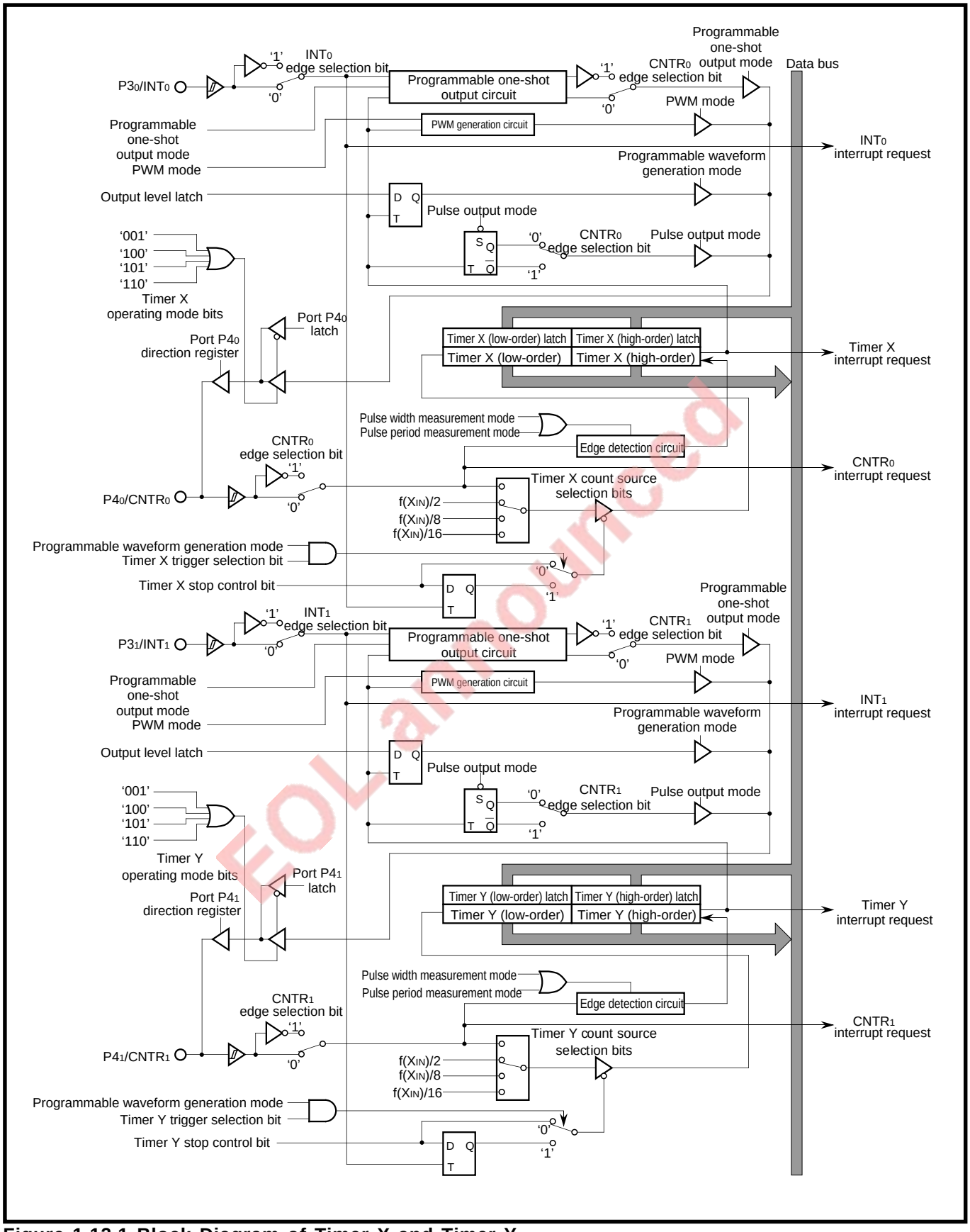


Figure 1.12.1 Block Diagram of Timer X and Timer Y

1.12.2 Registers Associated with Timer X and Timer Y

Figure 1.12.2 shows the memory map of the registers associated with timer X and timer Y.

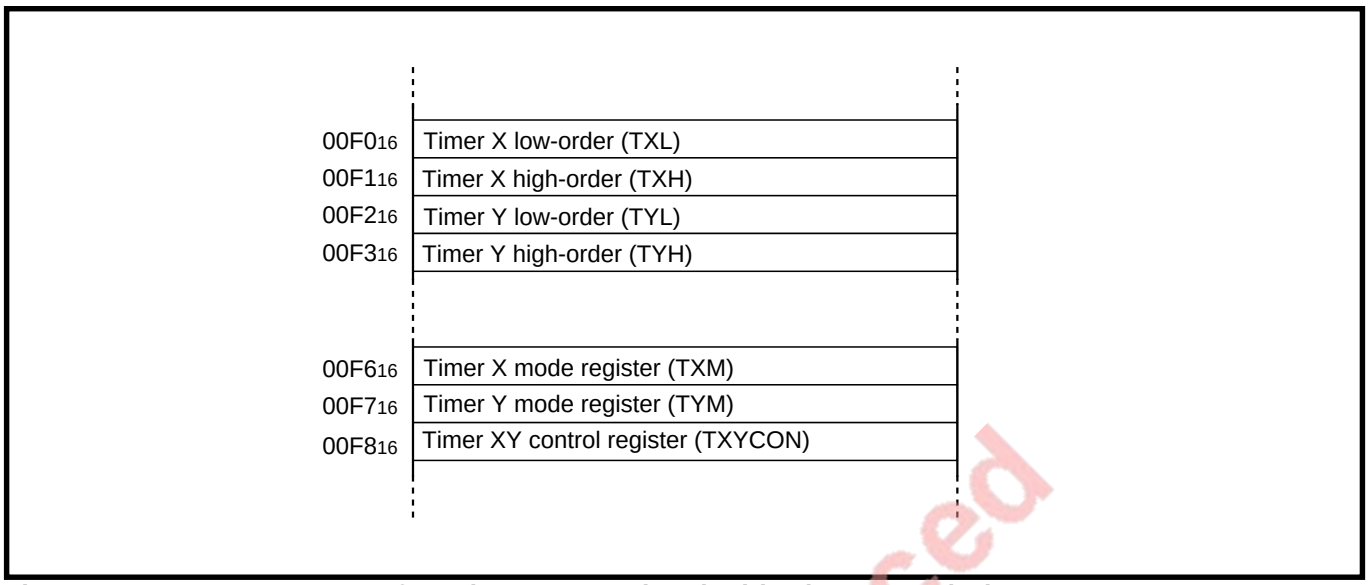


Figure 1.12.2 Memory Map of Registers Associated with Timer X and Timer Y

1.12 Timer X and Timer Y

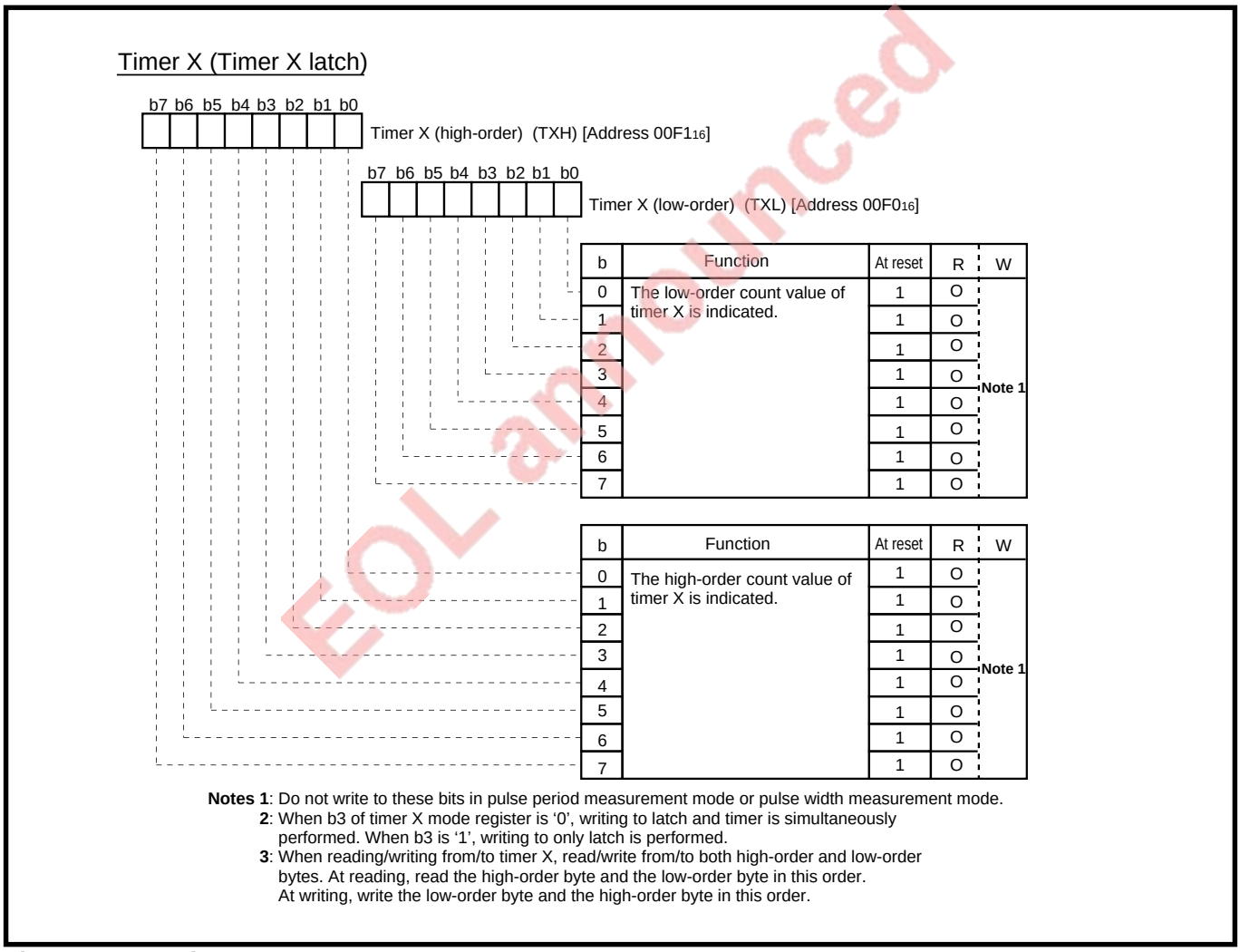
(1) Timer X and Timer Y

These are the 16-bit registers that count the count sources.

- When the timer X or Y write control bit of the timer X or Y mode register is;
  '0': data is written to the timer and the timer latch (**Note**), and
  '1': data is written to the timer latch only.
- In the pulse width measurement mode or the pulse period measurement mode, a read from the timer receives the contents of the timer latch.
  In the other modes, it receives the contents of the timer.

Note: The timer latches are the registers that hold the initial values automatically reloaded to the timers when they underflow, and they hold the measured values of pulse periods or widths. The timer latches cannot directly be read.

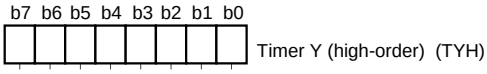
Figures 1.12.3 and 1.12.4 show the timer X and timer Y.



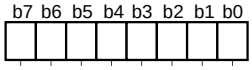
Notes 1: Do not write to these bits in pulse period measurement mode or pulse width measurement mode.
2: When b3 of timer X mode register is '0', writing to latch and timer is simultaneously performed. When b3 is '1', writing to only latch is performed.
3: When reading/writing from/to timer X, read/write from/to both high-order and low-order bytes. At reading, read the high-order byte and the low-order byte in this order. At writing, write the low-order byte and the high-order byte in this order.

Figure 1.12.3 Timer X

Timer Y (Timer Y latch)



Timer Y (high-order) (TYH) [Address $00F3_{16}$]



Timer Y (low-order) (TYL) [Address $00F2_{16}$]

b	Function	At reset	R	W
0	The low-order count value of timer Y is indicated.	1	O	Note 1
1		1	O	
2		1	O	
3		1	O	
4		1	O	
5		1	O	
6		1	O	
7		1	O	

b	Function	At reset	R	W
0	The high-order count value of timer Y is indicated.	1	O	Note 1
1		1	O	
2		1	O	
3		1	O	
4		1	O	
5		1	O	
6		1	O	
7		1	O	

Notes 1: Do not write to these bits in pulse period measurement mode or pulse width measurement mode.
2: When b3 of timer Y mode register is '0', writing to latch and timer is simultaneously performed. When b3 is '1', writing to only latch is performed.
3: When reading/writing from/to timer Y, read/write from/to both high-order and low-order bytes. At reading, read the high-order byte and the low-order byte in this order. At writing, write the low-order byte and the high-order byte in this order.

Figure 1.12.4 Timer Y

(2) Timer X Mode Register, Timer Y Mode Register, and Timer XY Control Register

These registers consist of the bits controlling the operations of timer X and timer Y. Figures 1.12.5, 1.12.6, and 1.12.7 show the timer X mode register, timer Y mode register, and timer XY control register respectively.

Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer X mode register (TXM) [Address $00F6_{16}$]

b	Name	Function	At reset	R	W
0	Timer X operating mode bits	b2 b1 b0 0 0 0 : Timer • event count mode 0 0 1 : Pulse output mode 0 1 0 : Pulse period measurement mode 0 1 1 : Pulse width measurement mode 1 0 0 : Programmable waveform generation mode 1 0 1 : Programmable one-shot output mode 1 1 0 : PWM mode 1 1 1 : Not available	0	O	O
1			0	O	O
2			0	O	O
3	Timer X write control bit	0 : Writing to both latch and timer 1 : Writing to latch only	0	O	O
4	Output level latch	0 : LOW output from $CNTR_0$ pin 1 : HIGH output from $CNTR_0$ pin	0	O	O
5	Timer X trigger selection bit	0 : Timer X free run in programmable waveform generation mode 1 : Trigger occurrence (input signal of INT_0 pin) and timer X start in programmable waveform generation mode	0	O	O
6	Timer X count source selection bits	b7 b6 0 0 : $f(X_{IN})/2$ 0 1 : $f(X_{IN})/8$ 1 0 : $f(X_{IN})/16$ 1 1 : Input from $CNTR_0$ pin	0	O	O
7			0	O	O

Figure 1.12.5 Timer X Mode Register

Timer Y mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer Y mode register (TYM) [Address $00F7_{16}$]

b	Name	Function	At reset	R	W
0	Timer Y operating mode bits	b2 b1 b0 0 0 0 : Timer • event count mode 0 0 1 : Pulse output mode 0 1 0 : Pulse period measurement mode 0 1 1 : Pulse width measurement mode 1 0 0 : Programmable waveform generation mode 1 0 1 : Programmable one-shot output mode 1 1 0 : PWM mode 1 1 1 : Not available	0	O	O
1			0	O	O
2			0	O	O
3	Timer Y write control bit	0 : Writing to both latch and timer 1 : Writing to latch only	0	O	O
4	Output level latch	0 : LOW output from $CNTR_1$ pin 1 : HIGH output from $CNTR_1$ pin	0	O	O
5	Timer Y trigger selection bit	0 : Timer Y free run in programmable waveform generation mode 1 : Trigger occurrence (input signal of INT_1 pin) and timer Y start in programmable waveform generation mode	0	O	O
6	Timer Y count source selection bits	b7 b6 0 0 : $f(X_{IN})/2$ 0 1 : $f(X_{IN})/8$ 1 0 : $f(X_{IN})/16$ 1 1 : Input from $CNTR_1$ pin	0	O	O
7			0	O	O

Figure 1.12.6 Timer Y Mode Register

Timer XY control register

b7 b6 b5 b4 b3 b2 b1 b0

0 0 0 0 0 0

Timer XY control register (TXYCON) [Address $00F8_{16}$]

b	Name	Function	At reset	R	W
0	Timer X stop control bit	0 : Count operation 1 : Count stop	1	O	O
1	Timer Y stop control bit	0 : Count operation 1 : Count stop	1	O	O
2	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
3			0	0	×
4			0	0	×
5			0	0	×
6			0	0	×
7			0	0	×

Figure 1.12.7 Timer XY Control Register

1.12 Timer X and Timer Y

1.12.3 Basic Operations of Timer X and Timer Y

Basic operations of timer X and timer Y are described below.
For details, refer to **(1) Operations** of each mode.

Count Sources

Timer X or timer Y can select the following count sources with the timer X or Y count source selection bits of the timer X or Y mode register:

- $f(X_{IN})/2$
- $f(X_{IN})/8$
- $f(X_{IN})/16$
- $CNTR_0$ or $CNTR_1$ pin input (in event count mode only).

Note: In the event count mode, the inverted signal of input to a CNTR pin is used as the count source when a CNTR edge selection bit of the edge polarity selection register is '1'.

Writes to and Reads from Timers

Write to and read from each timer two bytes together in the following order:

- Write: ① low-order byte ② high-order byte
- Read: ① high-order byte ② low-order byte

Note: When a read from and a write into the same timer are executed during an interrupt service routine etc., the normal operation cannot be performed.

■ **Writes to timers**

When 'TL (0000_{16} through $FFFF_{16}$)' is written to a timer, the following different operations are performed depending on the state of the timer X or Y write control bit of the timer X or Y mode register:

- In the '0' state of the timer X or Y write control bit, the 'TL' is set in both the timer latch and the timer.

Notes 1: A write to an operating timer causes the contents of the timer to be affected, so that the time from the last underflow until the next underflow is undefined.
2: A write to the low-order byte of an operating timer allows the timer to continue counting down until the next write to the high-order byte.

- In the '1' state of the timer X or Y write control bit, the 'TL' is set in the timer latch only.

Notes 1: A write to a stopped timer causes the contents of the timer not to be affected and allows the timer to count down from the value prior to the write. Therefore, the time from the start of count down until the first underflow is undefined.
2: If a write and an underflow occur at approximately the same time in an operating timer, the reloaded value may be undefined.

■ **Reads from timers**

The contents of a timer can be read by a read operation; however, the contents of the timer latch (measured value) are read in the pulse period measurement mode or the pulse width measurement mode.

Note: When the high-order byte of an operating timer is read, the low-order byte is set in the latch for reading. Therefore, the read value of the low-order byte retains the value at the time the high-order byte is being read.

Count Operation

The count operation (start/stop) of timer X or timer Y is controlled by the timer X or Y stop control bit of the timer XY control register as follows:

- When the timer X or Y stop control bit is set to '0', the timer starts counting.
- When the timer X or Y stop control bit is set to '1', the timer stops counting.

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source.

The timer X or Y stop control bit is recognized during the HIGH time of the count source. When the count has stopped, the count source cannot be accepted.

In the PWM mode, the high- and the low-order bytes of timer X or Y counts down each as an 8-bit timer.

Reloading Timers

When a timer reaches '0000$_{16}$' in the count operation, an underflow occurs at the subsequent rising edge of the count source, and the contents of the timer latch are reloaded to the timer.

In the pulse period measurement mode or the pulse width measurement mode, when a timer reaches '0000$_{16}$', an underflow occurs and a timer wraps around to 'FFFF$_{16}$' at the subsequent rising edge of the count source.

In the PWM mode, the high- and the low-order byte of a timer count down each as an 8-bit timer. When either the high- or the low-order byte of the timer becomes '01$_{16}$', an underflow occurs at the subsequent rising edge of the count source, and the contents of the timer latch are reloaded to the timer.

Timer Interrupt

At an underflow, the timer X or Y interrupt request bit of interrupt request register 1 is set to '1'; then a timer interrupt request is generated.

Table 1.12.1 lists the relation between timer count periods and values set to timer X and timer Y.

Table 1.12.1 Relation between Timer Count Periods and Values Set to Timer X and Timer Y

Clock Input Oscillation Frequency		f(XIN) = 8 MHz						f(XIN) = 4 MHz					
Count Source (Cycle Time)		f(XIN)/2 (0.25 μs)		f(XIN)/8 (1 μs)		f(XIN)/16 (2 μs)		f(XIN)/2 (0.5 μs)		f(XIN)/8 (2 μs)		f(XIN)/16 (4 μs)	
		High-order	Low-order	High-order	Low-order	High-order	Low-order	High-order	Low-order	High-order	Low-order	High-order	Low-order
Timer Period	1 ms	0F$_{16}$	9F$_{16}$	03$_{16}$	E7$_{16}$	01$_{16}$	F3$_{16}$	07$_{16}$	CF$_{16}$	01$_{16}$	F3$_{16}$	00$_{16}$	F9$_{16}$
	2 ms	1F$_{16}$	3F$_{16}$	07$_{16}$	CF$_{16}$	03$_{16}$	E7$_{16}$	0F$_{16}$	9F$_{16}$	03$_{16}$	E7$_{16}$	01$_{16}$	F3$_{16}$
	5 ms	4E$_{16}$	1F$_{16}$	13$_{16}$	87$_{16}$	09$_{16}$	C3$_{16}$	27$_{16}$	0F$_{16}$	09$_{16}$	C3$_{16}$	04$_{16}$	E1$_{16}$
	10 ms	9C$_{16}$	3F$_{16}$	27$_{16}$	0F$_{16}$	13$_{16}$	87$_{16}$	4E$_{16}$	1F$_{16}$	13$_{16}$	87$_{16}$	09$_{16}$	C3$_{16}$
	50 ms	–		C3$_{16}$	4F$_{16}$	61$_{16}$	A7$_{16}$	–		61$_{16}$	A7$_{16}$	30$_{16}$	D3$_{16}$
	100 ms	–		–		C3$_{16}$	4F$_{16}$	–		C3$_{16}$	4F$_{16}$	61$_{16}$	A7$_{16}$

1.12.4 Timer Mode and Event Count Mode

(1) Operations in Timer Mode and Event Count Mode

Operations in the timer mode and the event count mode are explained with Figure 1.12.8.

Count Sources

In the timer mode and the event count mode, timer X or timer Y can select the following count sources with the timer X or Y count source selection bits:

- $f(X_{IN})/2$
- $f(X_{IN})/8$
- $f(X_{IN})/16$

} timer mode

- $CNTR_0$ pin input (Timer X used)
- $CNTR_1$ pin input (Timer Y used)

} event count mode

Notes 1: In the event count mode, the inverted signal of input to a CNTR pin is used as the count source when a CNTR edge selection bit of the edge polarity selection register is '1'.

2: In the event count mode, keep the frequency of the CNTR pin input used as the count source $f(X_{IN})/4$ or less.

Writes to and Reads from Timers

When 'TL (0000_{16} through $FFFF_{16}$)' is written to a timer, the following different operations are performed depending on the state of the timer X or Y write control bit:

- In the '0' state of the timer X or Y write control bit, the 'TL' is set in both the timer latch and the timer (① in Figure 1.12.8).
- In the '1' state of the timer X or Y write control bit, the 'TL' is set in the timer latch only.

Also, the contents of the timer can be read by a read operation.

Count Operation

- When the timer X or Y stop control bit of the timer XY control register is cleared to '0', the timer starts counting (② in Figure 1.12.8).
- When the timer X or Y stop control bit is set to '1', the timer stops counting (③ in Figure 1.12.8).

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source (④ in Figure 1.12.8).

Reloading Timers

When a timer reaches '0000_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source, and the contents of the timer latch are reloaded to the timer (⑤ in Figure 1.12.8).

Timer Interrupt

At an underflow, the timer X or Y interrupt request bit is set to '1'; then a timer interrupt request is generated (⑥ in Figure 1.12.8).

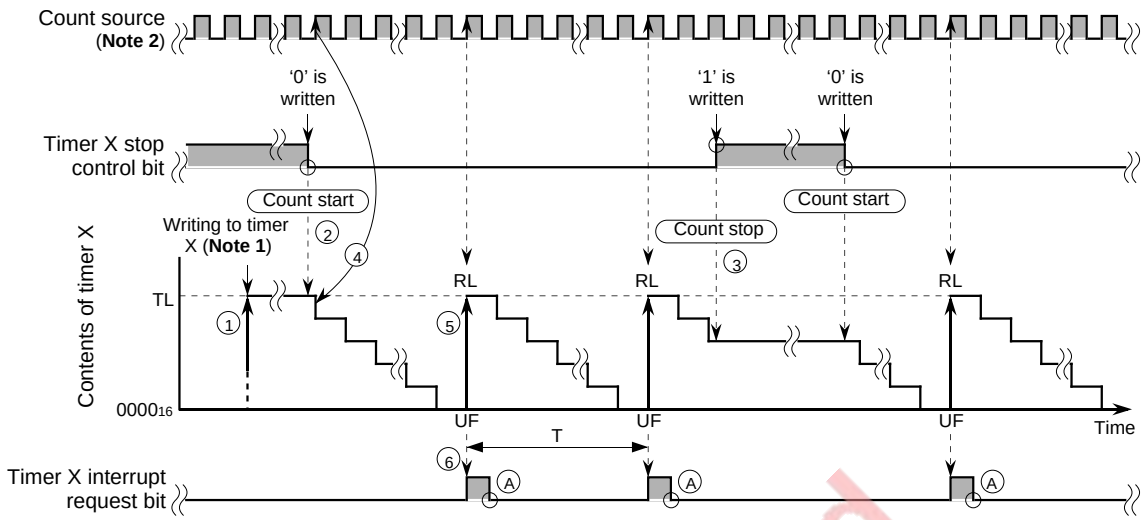


Figure 1.12.8 Operation Example in Timer Mode and Event Count Mode

(2) Setting of Timer Mode and Event Count Mode

Figures 1.12.9 and 1.12.10 show the setting of the timer mode and the event count mode.

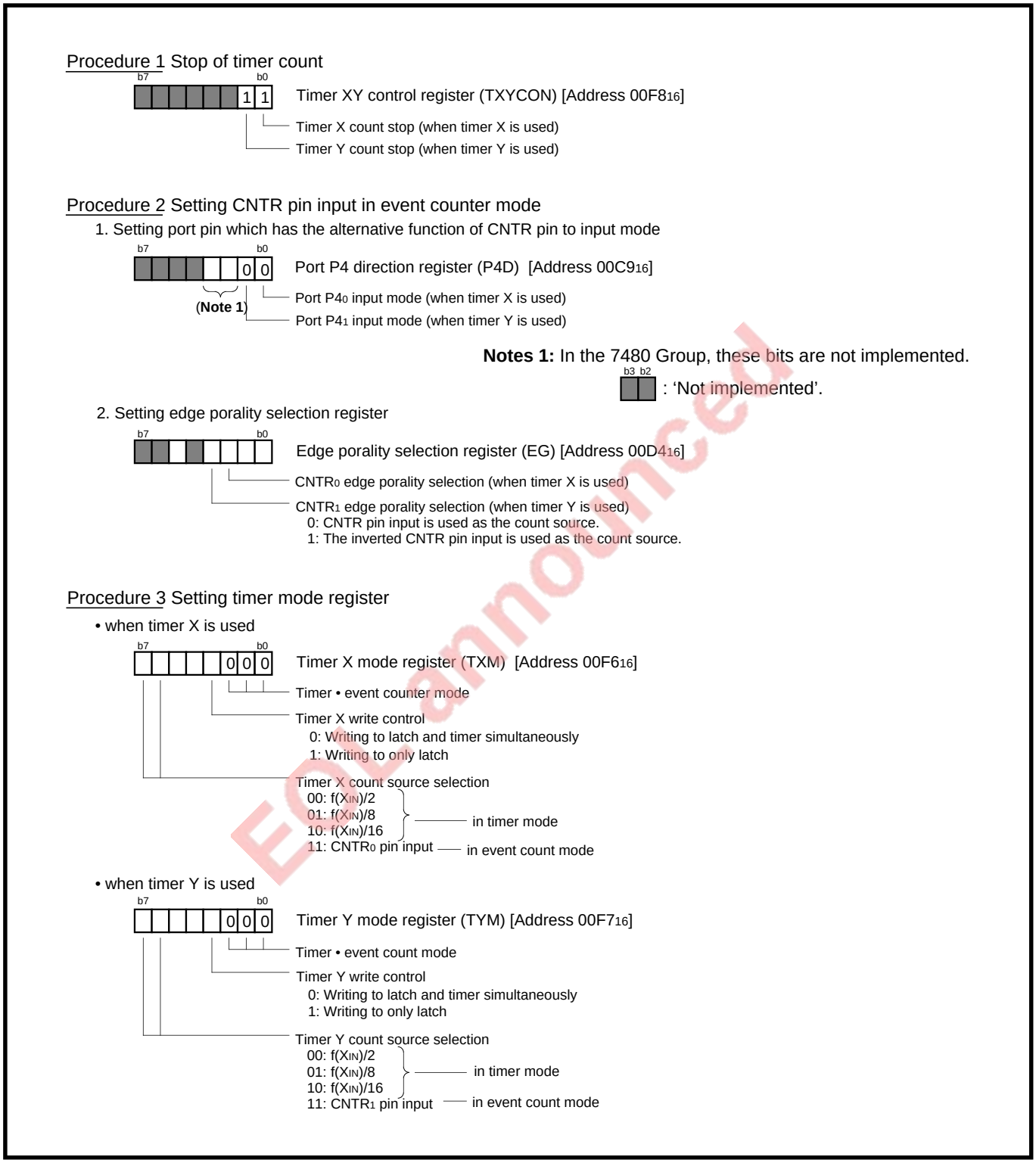


Figure 1.12.9 Setting of Timer Mode and Event Count Mode (1)

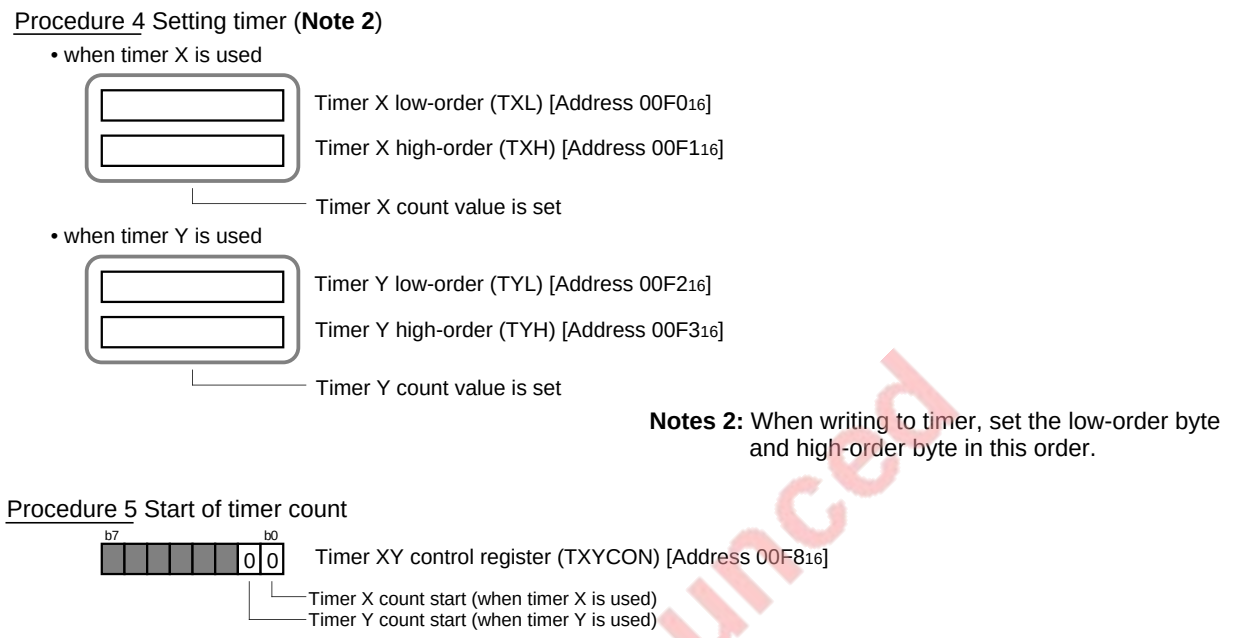


Figure 1.12.10 Setting of Timer Mode and Event Count Mode (2)

1.12.5 Pulse Output Mode

(1) Operations in Pulse Output Mode

Operations in the pulse output mode are explained with Figure 1.12.11.

Count Sources

In the pulse output mode, timer X or timer Y can select the following count sources with the timer X or Y count source selection bits:

- $f(X_{IN})/2$
- $f(X_{IN})/8$
- $f(X_{IN})/16$

Writes to and Reads from Timers

When 'TL (0000_{16} through $FFFF_{16}$)' is written to a timer, the following different operations are performed depending on the state of the timer X or Y write control bit:

- In the '0' state of the timer X or Y write control bit, the 'TL' is set in both the timer latch and the timer (① in Figure 1.12.11).
- In the '1' state of the timer X or Y write control bit, the 'TL' is set in the timer latch only.

Also, the contents of the timer can be read by a read operation.

Count Operation

- When the timer X or Y stop control bit is cleared to '0', the timer starts counting (② in Figure 1.12.11).
- When the timer X or Y stop control bit is set to '1', the timer stops counting (③ in Figure 1.12.11).

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source (④ in Figure 1.12.11).

Reloading Timers

When a timer reaches '0000_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source, and the contents of the timer latch are reloaded to the timer (⑤ in Figure 1.12.11).

Timer Interrupt

At an underflow, the timer X or Y interrupt request bit is set to '1'; then a timer interrupt request is generated (⑥ in Figure 1.12.11).

Pulse Output

At every underflows, polarity-inverted pulses are output from the following pins (⑦ in Figure 1.12.11):

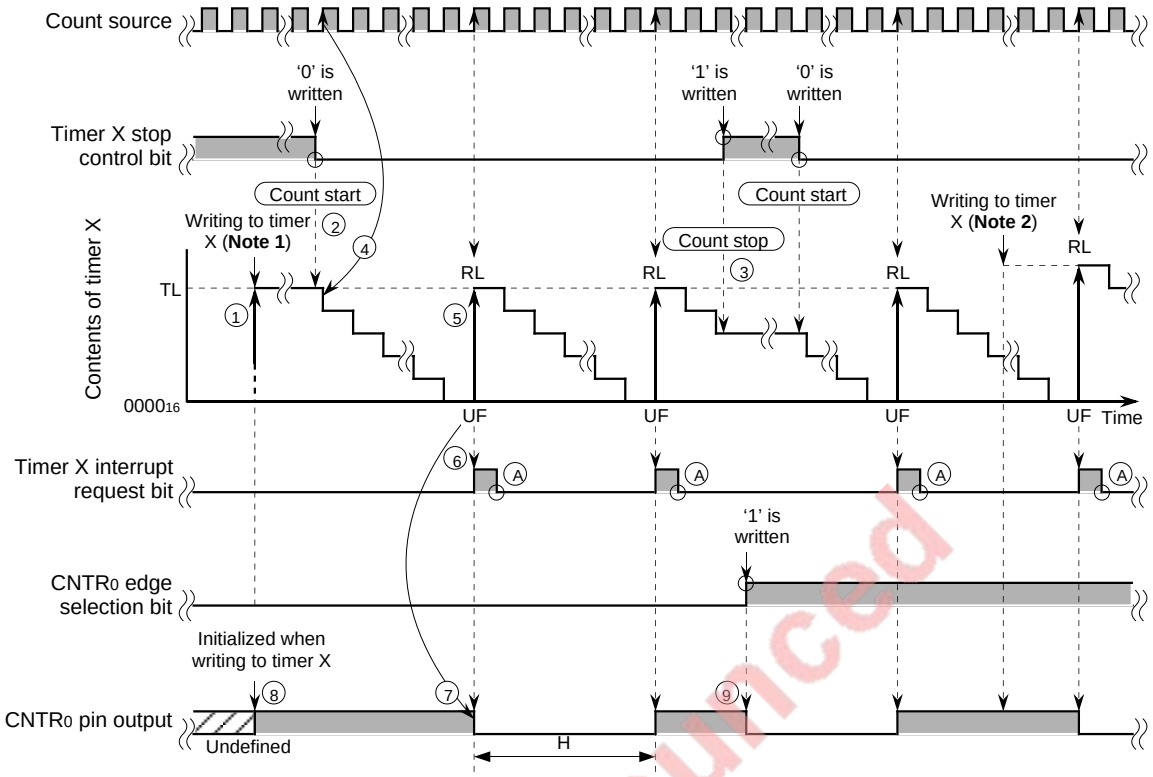
- $CNTR_0$ pin (Timer X used)
- $CNTR_1$ pin (Timer Y used)

When the timer X or Y write control bit is '0', the CNTR pin output is initialized to the following levels by a write to the timer:

- HIGH when the CNTR edge selection bit is '0' (⑧ in Figure 1.12.11).
- LOW when the CNTR edge selection bit is '1'.

Notes 1: When the timer X or Y write control bit is '1', the CNTR pin output level cannot be initialized by a write to the timer.

2: In the pulse output mode, the output level of a CNTR pin is inverted when the CNTR edge selection bit is switched (⑨ in Figure 1.12.11).



TL: Setting value to timer X
RL: Contents of timer latch is reloaded
UF: Underflow
H: Pulse width

$$H(s) = \frac{1}{\text{Count source frequency}} \times (\text{Setting value to timer X} + 1)$$

Ⓐ: Clearing by writing '0' to interrupt request bit or accepting interrupt request

Notes 1: In this case, timer X write control bit is '0' (writing to timer and timer latch simultaneously).
2: In this case, timer X write control bit is '1' (writing to only timer latch).

Figure 1.12.11 Operation Example in Pulse Output Mode

(2) Setting of Pulse Output Mode

Figures 1.12.12 and 1.12.13 show the setting of the pulse output mode.

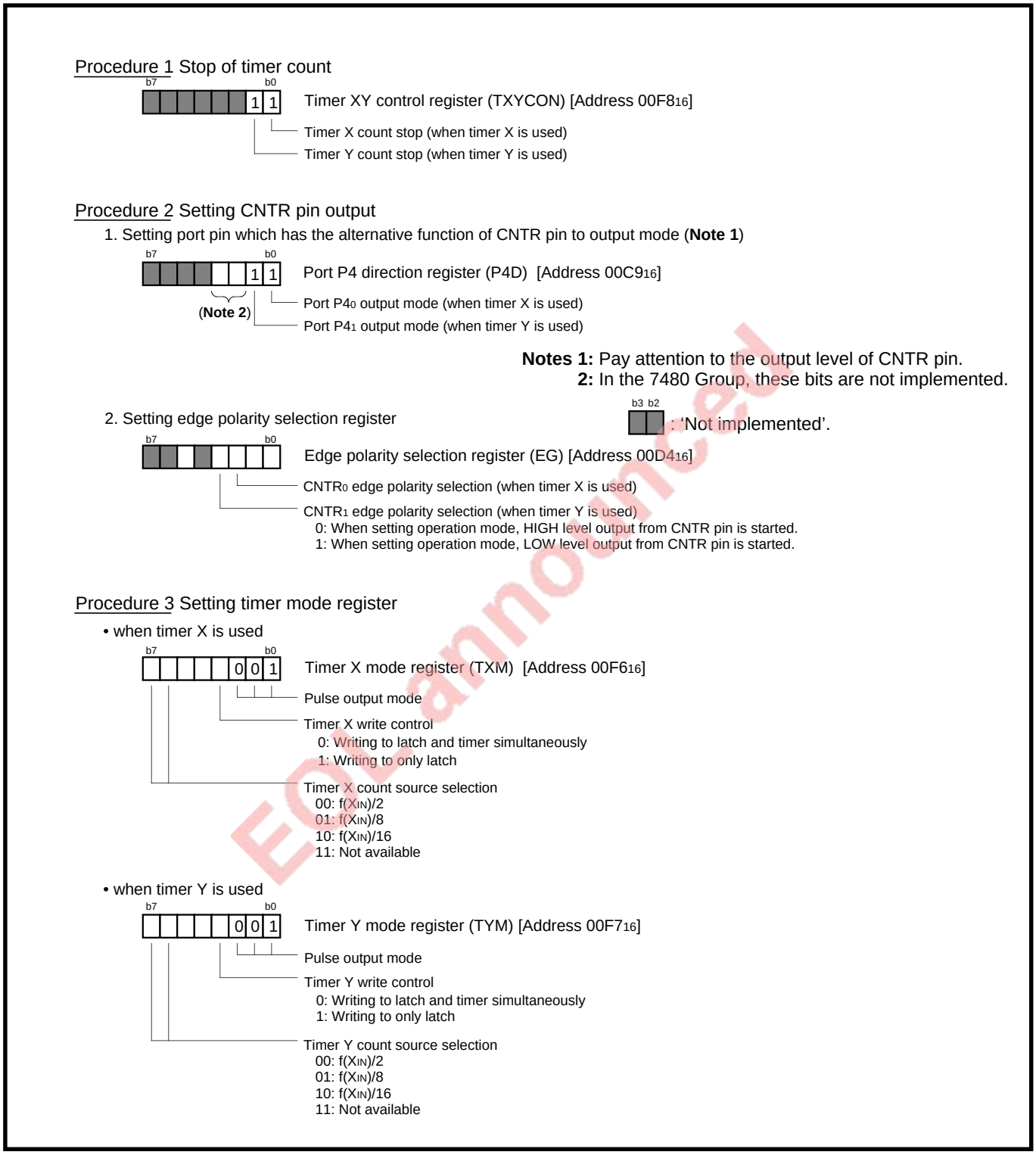


Figure 1.12.12 Setting of Pulse Output Mode (1)

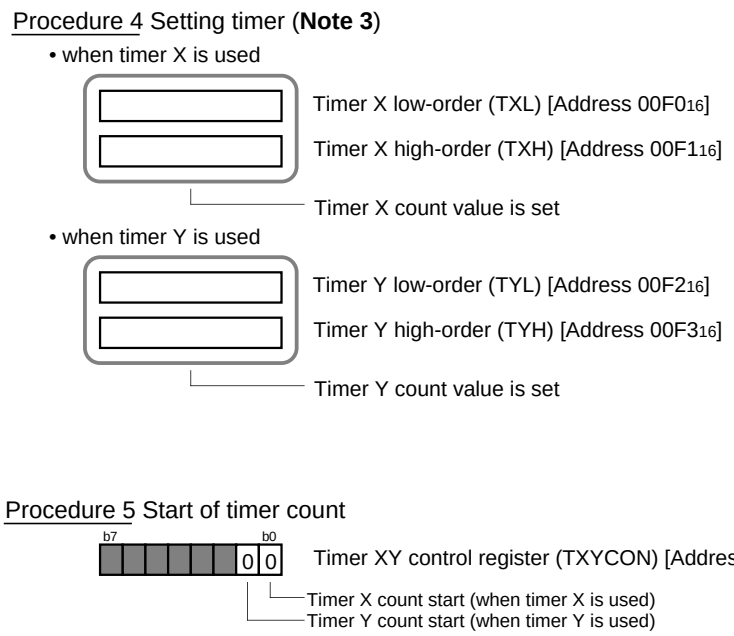


Figure 1.12.13 Setting of Pulse Output Mode (2)

1.12.6 Pulse Period Measurement Mode

(1) Operations in Pulse Period Measurement Mode

Operations in the pulse period measurement mode are explained with Figure 1.12.14.

Count Sources

In the pulse period measurement mode, timer X or timer Y can select the following count sources with the timer X or Y count source selection bits:

- $f(X_{IN})/2$
- $f(X_{IN})/8$
- $f(X_{IN})/16$

Writes to and Reads from Timers

In the pulse period measurement mode, do not write to timers.
When a timer is read, the read value is the contents of the timer latch (measured value of the last pulse period).

Count Operation

- When the timer X or Y stop control bit is cleared to '0', the timer starts counting (① in Figure 1.12.14).
- When the timer X or Y stop control bit is set to '1', the timer stops counting.

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source (② in Figure 1.12.14).

Reloading Timers

When a timer reaches '0000_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source and a timer wraps around to '$FFFF_{16}$' (③ in Figure 1.12.14).
When the valid edge of a CNTR pin input is detected in the count operation, the timer goes to '$FFFF_{16}$' (④ in Figure 1.12.14).

Timer Interrupt

At an underflow, the timer X or Y interrupt request bit of interrupt request register 1 is set to '1'; then a timer interrupt request is generated (⑤ in Figure 1.12.14).

CNTR Interrupt

When the valid edge of a CNTR pin input is detected, the CNTR interrupt request bit of interrupt request register 2 is set to '1', and the CNTR interrupt request is generated (⑥ in Figure 1.12.14). The measured value of the pulse period must be read at this time.

Pulse Period Measurement

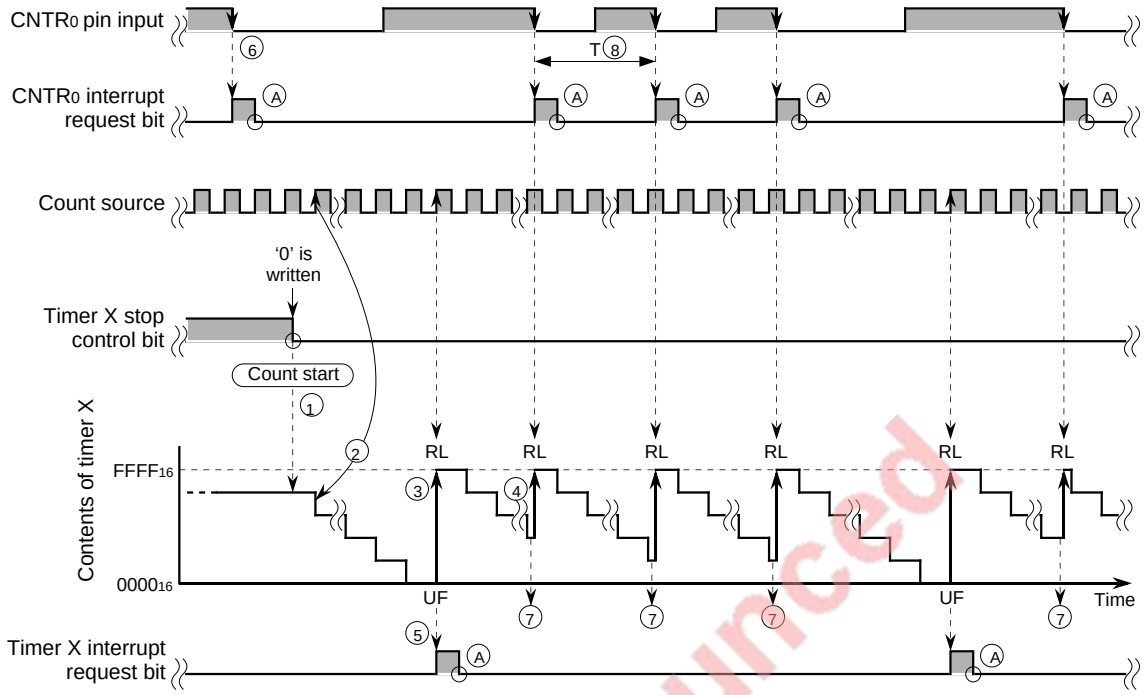
When any one of the following valid edges are detected, the complement on one of the contents of the timer is written to the timer latch (⑦ in Figure 1.12.14). The contents of the timer latch are retained until the measurement of the next pulse period is complete.

- Valid edge of a $CNTR_0$ pin input (Timer X used)
- Valid edge of a $CNTR_1$ pin input (Timer Y used)

The measurement type of pulse period is selected by a CNTR edge selection bit of the edge polarity selection register as follows:

- The period from a falling edge of a CNTR pin input until the next falling edge when the CNTR edge selection bit is '0' (⑧ in Figure 1.12.14).
- The period from a rising edge of a CNTR pin input until the next rising edge when the CNTR edge selection bit is '1'.

• When the period from falling edge until the next falling edge is measured (**Note**)



RL : 'FFFF16' is reloaded
UF : Underflow
T : Pulse period

$$T(s) = \frac{1}{\text{Count source frequency}} \times \{(\text{Contents of timer latch} + 1) + 65536 \times n\}$$

n: The number of timer interrupt request generation during measuring.

Ⓐ : Clearing by writing '0' to interrupt request bit or accepting interrupt request

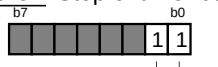
Note: In this case, CNTR0 edge selection bit is '0' (falling edge is valid).

Figure 1.12.14 Operation Example in Pulse Period Measurement Mode

(2) Setting of Pulse Period Measurement Mode

Figure 1.12.15 shows the setting of the pulse period measurement mode.

Procedure 1 Stop of timer count



Timer XY control register (TXYCON) [Address $00F8_{16}$]

Timer X count stop (when timer X is used)

Timer Y count stop (when timer Y is used)

Procedure 2 Setting CNTR pin input

1. Setting port pin which has the alternative function of CNTR pin to input mode

Port P4 direction register (P4D) [Address $00C9_{16}$]

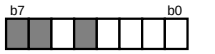
Port $P4_0$ input mode (when timer X is used)

Port $P4_1$ input mode (when timer Y is used)

Note: In the 7480 Group, these bits are not implemented.

: 'Not implemented'.

2. Setting edge polarity selection register



Edge polarity selection register (EG) [Address $00D4_{16}$]

$CNTR_0$ edge polarity selection (when timer X is used)

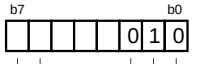
$CNTR_1$ edge polarity selection (when timer Y is used)

0: CNTR pin input from falling to falling is counted

1: CNTR pin input from rising to rising is counted

Procedure 3 Setting timer mode register

• when timer X is used



Timer X mode register (TXM) [Address $00F6_{16}$]

Pulse period measurement mode

Timer X count source selection

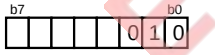
00: $f(X_{IN})/2$

01: $f(X_{IN})/8$

10: $f(X_{IN})/16$

11: Not available

• when timer Y is used



Timer Y mode register (TYM) [Address $00F7_{16}$]

Pulse period measurement mode

Timer Y count source selection

00: $f(X_{IN})/2$

01: $f(X_{IN})/8$

10: $f(X_{IN})/16$

11: Not available

Procedure 4 Start of timer count

Timer XY control register (TXYCON) [Address $00F8_{16}$]

Timer X count start (when timer X is used)

Timer Y count start (when timer Y is used)

Figure 1.12.15 Setting of Pulse Period Measurement Mode

1.12.7 Pulse Width Measurement Mode

(1) Operations in Pulse Width Measurement Mode

Operations in the pulse width measurement mode are explained with Figure 1.12.16.

Count Sources

In the pulse width measurement mode, timer X or timer Y can select the following count sources with the timer X or Y count source selection bits:

- f(X_{IN})/2
- f(X_{IN})/8
- f(X_{IN})/16

Writes to and Reads from timers

In the pulse width measurement mode, do not write to timers.
When a timer is read, the read value is the contents of the timer latch (measured value of the last pulse width).

Count Operation

- When the timer X or Y stop control bit is cleared to '0', the timer starts counting (① in Figure 1.12.16).
- When the timer X or Y stop control bit is set to '1', the timer stops counting.

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source (② in Figure 1.12.16).

Reloading Timers

When a timer reaches '0000_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source and a timer wraps around to '$FFFF_{16}$' (③ in Figure 1.12.16).
When the valid edge of a CNTR pin input is detected in the count operation, the timer goes to '$FFFF_{16}$' (④ in Figure 1.12.16).

Timer Interrupt

At an underflow, the timer X or Y interrupt request bit is set to '1'; then a timer interrupt request is generated (⑤ in Figure 1.12.16).

CNTR Interrupt

When the valid edge of a CNTR pin input is detected, the CNTR interrupt request bit of interrupt request register 2 is set to '1', and the CNTR interrupt request is generated (⑥ in Figure 1.12.16). The measured value of the pulse width must be read at this time.

Pulse Width Measurement

When any one of the following valid edges are detected, the complement on one of the contents of the timer is written to the timer latch (⑦ in Figure 1.12.16). The contents of the timer latch are retained until the measurement of the next pulse width is complete.

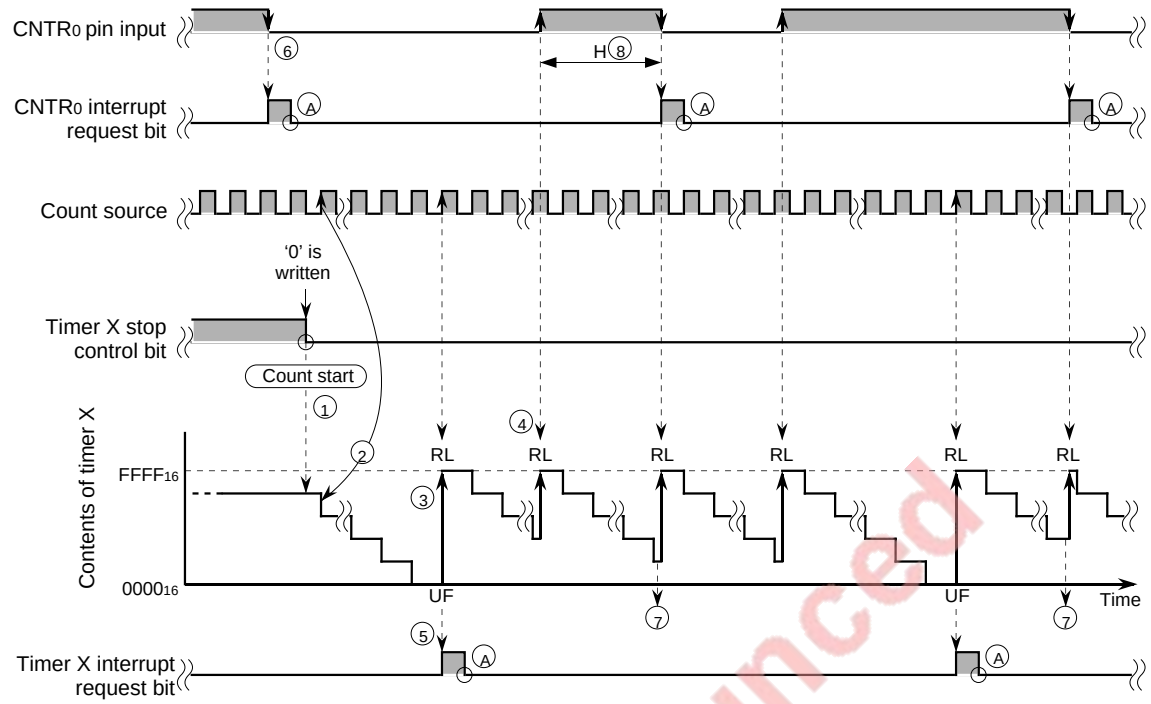
- Valid edge of a $CNTR_0$ pin input (Timer X used)
- Valid edge of a $CNTR_1$ pin input (Timer Y used)

The measurement type of pulse width is selected by a CNTR edge selection bit as follows:

- HIGH-level period from a rising edge of a CNTR pin input until the next falling edge when the CNTR edge selection bit is '0' (⑧ in Figure 1.12.16).
- LOW-level period from a falling edge of a CNTR pin input until the next rising edge when the CNTR edge selection bit is '1'.

1.12 Timer X and Timer Y

• When HIGH-level period is measured (**Note**)



RL : 'FFFF16' is reloaded
UF : Underflow
H : Pulse width

$$H(s) = \frac{1}{\text{Count source frequency}} \times \{(\text{Contents of timer latch}+1) + 65536 \times n\}$$

n: The number of timer interrupt request generation during measuring.

Ⓐ : Clearing by writing '0' to interrupt request bit or accepting interrupt request

Note: In this case, CNTR0 edge selection bit is '0' (falling edge is valid).

Figure 1.12.16 Operation Example in Pulse Width Measurement Mode

(2) Setting of Pulse Width Measurement Mode

Figure 1.12.17 shows the setting of the pulse width measurement mode.

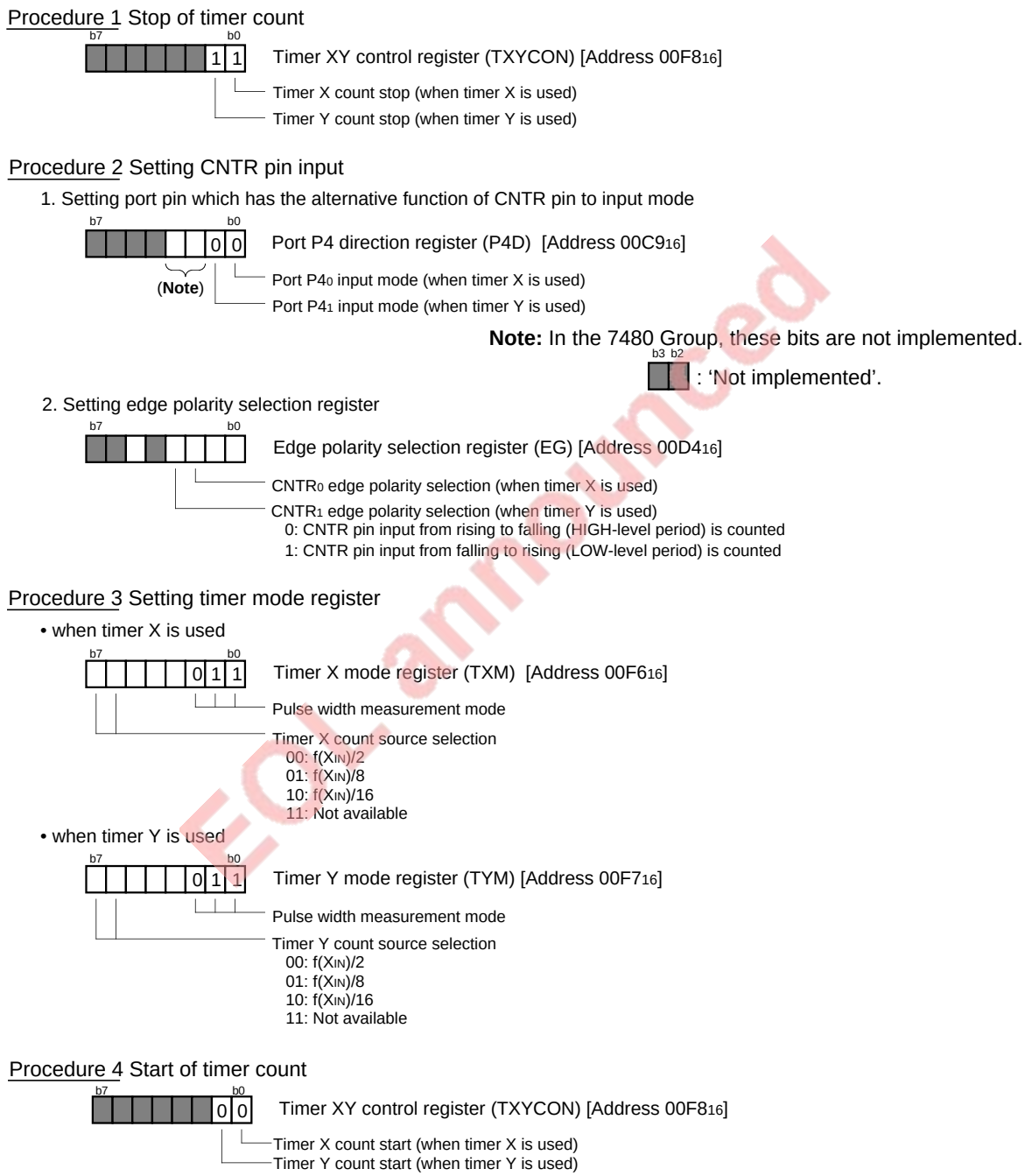


Figure 1.12.17 Setting of Pulse Width Measurement Mode

1.12.8 Programmable Waveform Generation Mode

(1) Operations in Programmable Waveform Generation Mode

Operations in the programmable waveform generation mode are explained with Figure 1.12.18.

Count Sources

In the programmable waveform generation mode, timer X or timer Y can select the following count sources with the timer X or Y count source selection bits:

- f(X_{IN})/2
- f(X_{IN})/8
- f(X_{IN})/16

Writes to and Reads from Timers

When 'TL (0000_{16} through $FFFF_{16}$)' is written to a timer, the following different operations are performed depending on the state of the timer X or Y write control bit:

- In the '0' state of the timer X or Y write control bit, the 'TL' is set in both the timer latch and the timer (① in Figure 1.12.18).
- In the '1' state of the timer X or Y write control bit, the 'TL' is set in the timer latch only (② in Figure 1.12.18).

Also, the contents of the timer can be read by a read operation.

Count Operation

In the programmable waveform generation mode, the following starting point of a timer can be selected with the timer X or Y trigger selection bit of the timer X or Y mode register:

- When a valid edge of an INT_0 pin input is detected (Timer X used).
- When a valid edge of an INT_1 pin input is detected (Timer Y used).

Clearing the timer X or Y stop control bit to '0' brings the following results:

■ • When the timer X or Y trigger selection bit is cleared to '0', the timer starts counting.
  • When the timer X or Y trigger selection bit is set to '1', the timer starts counting as soon as the valid edge of an INT pin input is detected (③ in Figure 1.12.18).

When the timer X or Y stop control bit is set to '1', the timer stops counting.

■ **Note:** Keep the trigger widths input to the INT pins 250 ns or more.

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source (④ in Figure 1.12.18).

Reloading Timers

When a timer reaches '0000_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source, and the contents of the timer latch are reloaded to the timer (⑤ in Figure 1.12.18).

Timer Interrupt

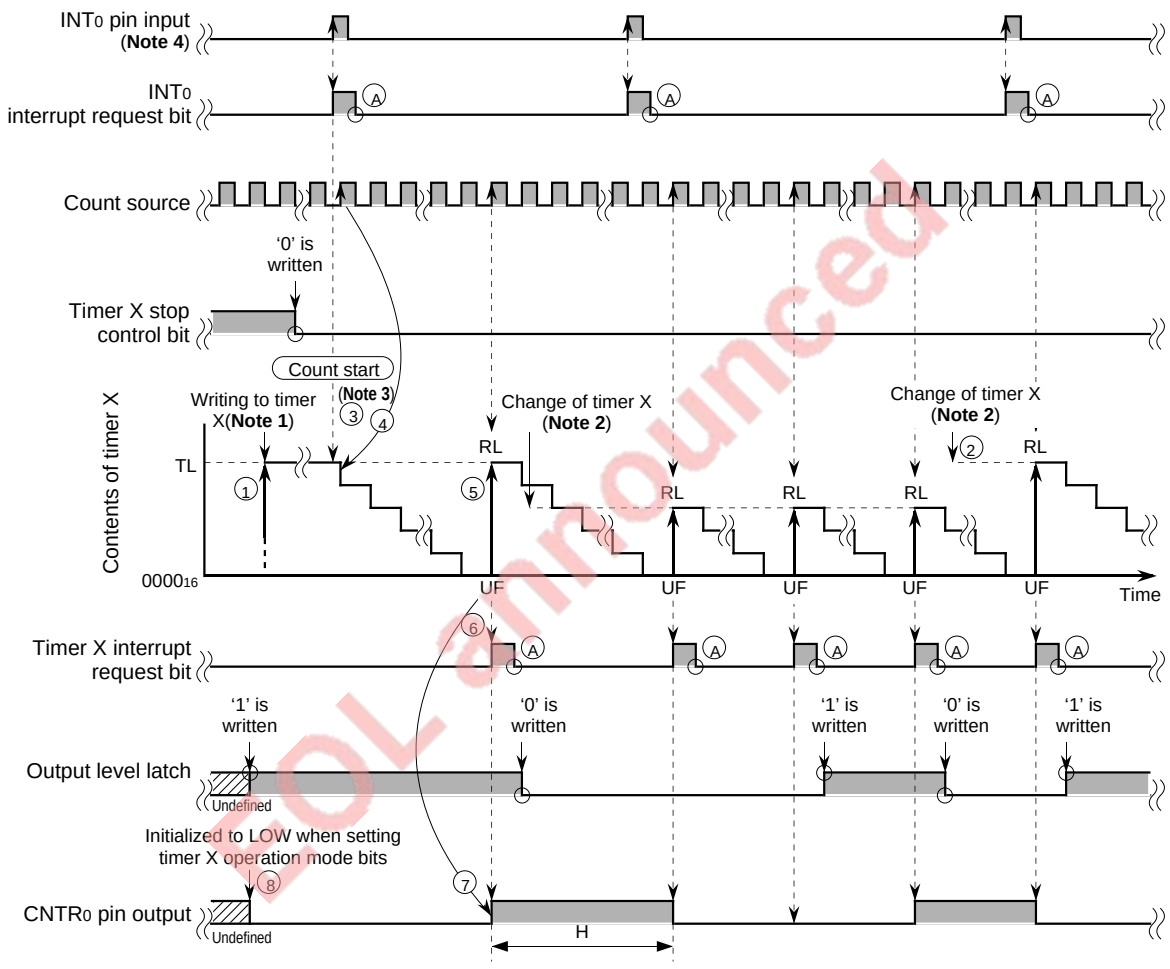
At an underflow, the timer X or Y interrupt request bit is set to '1'; then a timer interrupt request is generated (⑥ in Figure 1.12.18).

Generation of Programmable Waveform

When an underflow occurs in a timer, the contents of the output level latches of the timer X or Y mode register are output from the following pins (⑦ in Figure 1.12.18):

- $CNTR_0$ pin (Timer X used)
- $CNTR_1$ pin (Timer Y used)

When the timer X or Y operation mode bits of the timer X or Y mode register, which are set to other modes, are switched to the programmable waveform generation mode, the CNTR pin outputs are initialized to LOW (⑧ in Figure 1.12.18).



TL : Setting value to timer X
RL : Contents of timer latch is reloaded
UF : Underflow
H : Pulse width

$$H(s) = \frac{1}{\text{Count source frequency}} \times (\text{Setting value to timer X+1})$$

Ⓐ : Clearing by writing '0' to interrupt request bit or accepting interrupt request

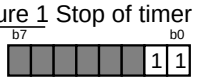
Notes 1: In this case, timer X write control bit is '0' (writing to timer and timer latch simultaneously).
2: In this case, timer X write control bit is '1' (writing to only timer latch).
3: In this case, timer X trigger selection bit is '1' (trigger occurs, at the same time, timer X is activated).
4: In this case, INT_0 edge selection bit is '1' (rising edge is valid).

Figure 1.12.18 Operation Example in Programmable Waveform Generation Mode

(2) Setting of Programmable Waveform Generation Mode

Figures 1.12.19 and 1.12.20 show the setting of the programmable waveform generation mode.

Procedure 1 Stop of timer count



Timer XY control register (TXYCON) [Address $00F8_{16}$]

Timer X count stop (when timer X is used)

Timer Y count stop (when timer Y is used)

Procedure 2 Setting INT pin input and CNTR pin output

1. Setting port pin which has the alternative function of the CNTR pin to output mode (**Note 1**)

(**Note 2**)

Port Register (P4D) [Address $00C9_{16}$]

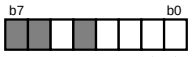
$P4_0$ output mode (when timer X is used)

$P4_1$ output mode (when timer Y is used)

Notes 1: Pay attention to the output level of CNTR pin.
2: In the 7480 Group, these bits are not implemented.

: 'Not implemented'.

2. Setting INT edge selection bit when timer is activated by the trigger of INT pin input



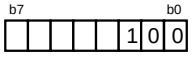
Edge polarity selection register (EG) [Address $00D4_{16}$]

INT_0 edge polarity selection (when timer X is used)

INT_1 edge polarity selection (when timer Y is used)
0: Timer is activated by detecting the falling edge of INT pin input
1: Timer is activated by detecting the rising edge of INT pin input

Procedure 3 Setting timer mode register

• when timer X is used



Timer X mode register (TXM) [Address $00F6_{16}$]

Programmable waveform generation mode (**Note 3**)

Timer X write control
0: Writing to latch and timer simultaneously
1: Writing to only latch

Output level latch

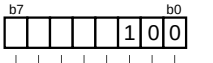
Timer X trigger selection
0: Timer X free run
1: Trigger occurrence (input signal of INT_0 pin) and timer X start

Timer X count source selection
00: $f(X_{IN})/2$
01: $f(X_{IN})/8$
10: $f(X_{IN})/16$
11: Not available

Notes 3: The CNTR pin output is initialized to LOW when the operation mode bits set to other modes are set to the programmable waveform generation mode.

Figure 1.12.19 Setting of Programmable Waveform Generation Mode (1)

• when timer Y is used



Timer Y mode register (TYM) [Address $00F7_{16}$]

Programmable waveform generation mode (**Note 3**)

Timer Y write control
0: Writing to latch and timer simultaneously
1: Writing to only latch

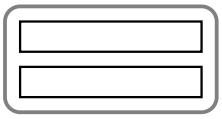
Output level latch

Timer Y trigger selection
0: Timer Y free run
1: Trigger occurrence (input signal of INT_1 pin) and timer Y start

Timer Y count source selection
00: $f(X_{IN})/2$
01: $f(X_{IN})/8$
10: $f(X_{IN})/16$
11: Not available

Procedure 4 Setting timer (**Note 4**)

• when timer X is used

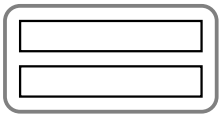


Timer X low-order (TXL) [Address $00F0_{16}$]

Timer X high-order (TXH) [Address $00F1_{16}$]

Timer X count value is set

• when timer Y is used



Timer Y low-order (TYL) [Address $00F2_{16}$]

Timer Y high-order (TYH) [Address $00F3_{16}$]

Timer Y count value is set

Notes 4: When writing to timer, set the low-order byte and high-order byte in this order.

Procedure 5 Start of timer count (**Note 5**)

Timer XY control register (TXYCON) [Address $00F8_{16}$]

Timer X count start (when timer X is used)
Timer Y count start (when timer Y is used)

Notes 5: When bit 5 of timer mode register is '1', timer count does not start at this time. Trigger (input signal of INT pin) occurs, at the same time, timer count starts.

Notes 6: Keep the trigger width input to INT pin 250 ns or more.

Figure 1.12.20 Setting of Programmable Waveform Generation Mode (2)

1.12.9 Programmable One-Shot Output Mode

(1) Operations in Programmable One-Shot Output Mode

Operations in the programmable one-shot output mode is explained with Figure 1.12.21.

Count Sources

In the programmable one-shot output mode, timer X or timer Y can select the following count sources with the timer X or Y count source selection bits:

- f(X_{IN})/2
- f(X_{IN})/8
- f(X_{IN})/16

Writes to and Reads from Timers

When 'TL (0000_{16} through $FFFF_{16}$)' is written to a timer, the following different operations are performed depending on the state of the timer X or Y write control bit:

- In the '0' state of the timer X or Y write control bit, the 'TL' is set in both the timer latch and the timer (① in Figure 1.12.21).
- In the '1' state of the timer X or Y write control bit, the 'TL' is set in the timer latch only.

Also, the contents of the timer can be read by a read operation.

Count Operation

- When the timer X or Y stop control bit is cleared to '0', the timer starts counting (② in Figure 1.12.21).
- When the timer X or Y stop control bit is set to '1', the timer stops counting.

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source (③ in Figure 1.12.21).

Reloading Timers

When a timer reaches '0000_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source, and the contents of the timer latch are reloaded to the timer (④ in Figure 1.12.21).

When the valid edge of an INT pin input is detected, the contents of the timer latch are also reloaded (⑤ in Figure 1.12.21).

Timer Interrupt

At an underflow, the timer X or Y interrupt request bit is set to '1'; then a timer interrupt request is generated (⑥ in Figure 1.12.21).

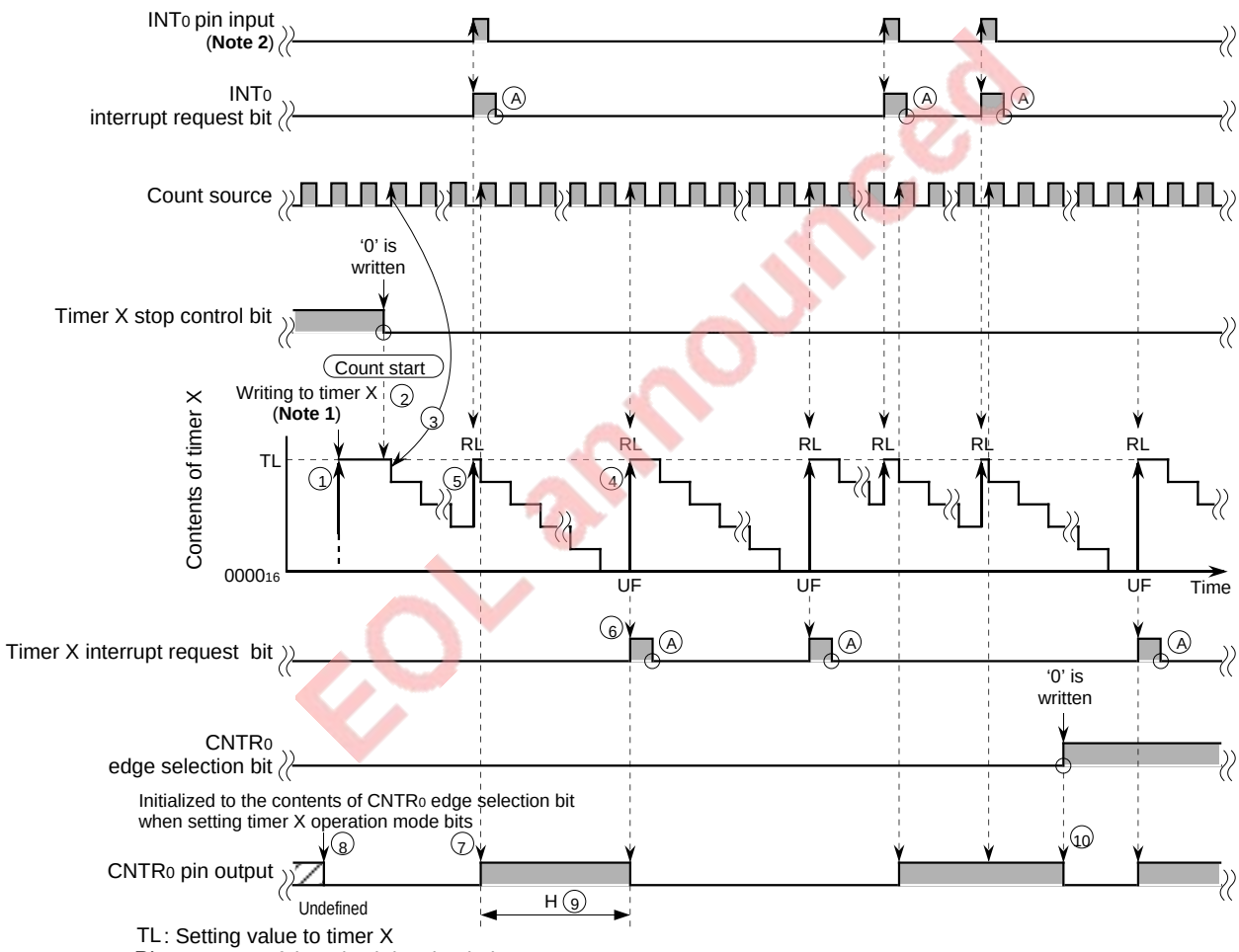
Generation of Programmable One-Shot Pulse

- When timer X is used, a one-shot pulse is output from the $CNTR_0$ pin when the valid edge of an INT_0 pin input is detected (⑦ in Figure 1.12.21).
- When timer Y is used, a one-shot pulse is output from the $CNTR_1$ pin when the valid edge of an INT_1 pin input is detected.

When the timer X or Y operation mode bits of the timer X or Y mode register are set to the programmable one-shot output mode, the CNTR pin outputs are initialized to the content of the CNTR edge selection bit (⑧ in Figure 1.12.21).
The CNTR pin output remains at the inverted level of the content of the CNTR edge selection bit for the period from the rising edge*1 of the count source immediately after the valid edge of an INT pin input is detected until the subsequent an underflow (⑨ in Figure 1.12.21).

Notes 1: Keep the trigger widths input to the INT pins 250 ns or more.
2: In the programmable one-shot output mode, the output level of a CNTR pin is inverted when the CNTR edge selection bit is switched (⑩ in Figure 1.12.21).

∗1: One cycle or less of the rising edge of the count source, after the valid edge of an INT pin input is detected.



TL: Setting value to timer X
RL: Contents of timer latch is reloaded
UF: Underflow
H: Pulse width

$$H(s) = \frac{1}{\text{Count source frequency}} \times (\text{Setting value to timer X} + 1)$$

Ⓐ : Clearing by writing '0' to interrupt request bit or accepting interrupt request

Notes 1: In this case, timer X write control bit is '0' (writing to timer and timer latch simultaneously).
2: In this case, INT0 edge selection bit is '1' (rising edge is valid).

Figure 1.12.21 Operation Example in Programmable One-Shot Output Mode

(2) Setting of Programmable One-Shot Output Mode

Figures 1.12.22 and 1.12.23 show the setting of the programmable one-shot output mode.

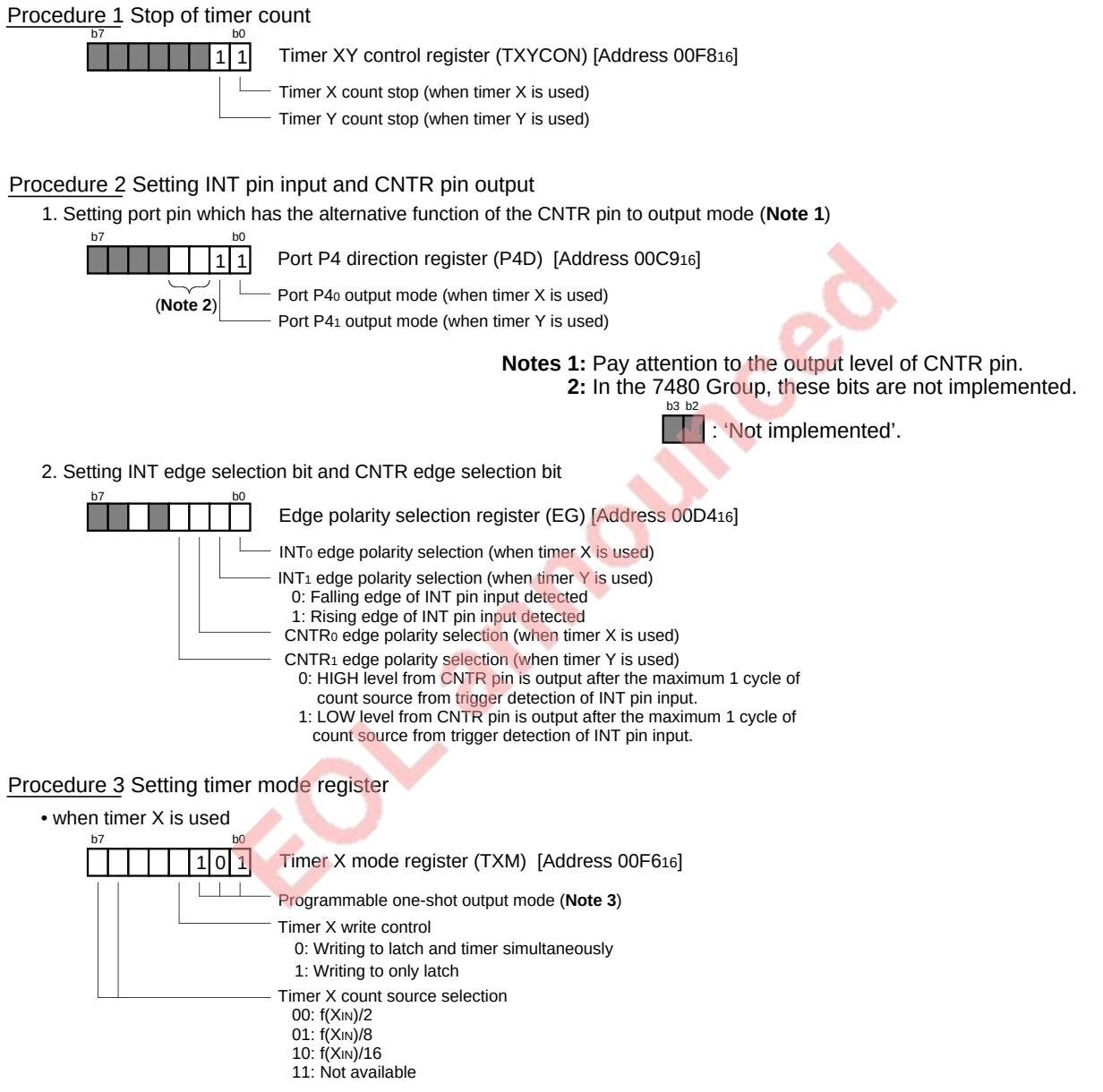
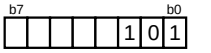


Figure 1.12.22 Setting of Programmable One-Shot Output Mode (1)

• when timer Y is used



Timer Y mode register (TYM) [Address $00F7_{16}$]

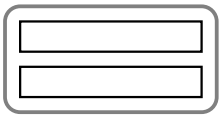
Programmable one-shot output mode (**Note 3**)

Timer Y write control
0: Writing to latch and timer simultaneously
1: Writing to only latch

Timer Y count source selection
00: $f(X_{IN})/2$
01: $f(X_{IN})/8$
10: $f(X_{IN})/16$
11: Not available

Procedure 4 Setting timer (**Note 4**)

• when timer X is used

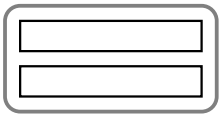


Timer X low-order (TXL) [Address $00F0_{16}$]

Timer X high-order (TXH) [Address $00F1_{16}$]

Timer X count value is set

• when timer Y is used



Timer Y low-order (TYL) [Address $00F2_{16}$]

Timer Y high-order (TYH) [Address $00F3_{16}$]

Timer Y count value is set

Notes 4: When writing to timer, set the low-order byte and high-order byte in this order.

Procedure 5 Start of timer count

Timer XY control register (TXYCON) [Address $00F8_{16}$]

Timer X count start (when timer X is used)
Timer Y count start (when timer Y is used)

Notes 5: Keep the trigger width input to INT pin 250 ns or more.

Figure 1.12.23 Setting of Programmable One-Shot Output Mode (2)

1.12.10 PWM Mode

(1) Operations in PWM Mode

Operations in the PWM mode are explained with Figure 1.12.24.

Count Sources

In the PWM mode, timer X or timer Y can select the following count sources with the timer X or Y count source selection bits:

- f(XIN)/2
- f(XIN)/8
- f(XIN)/16

Writes to and Reads from Timers

When 'TL (0000_{16} through $FFFF_{16}$)' is written to a timer, the following different operations are performed depending on the state of the timer X or Y write control bit:

- In the '0' state of the timer X or Y write control bit, the 'TL' is set in both the timer latch and the timer (① in Figure 1.12.24).
- In the '1' state of the timer X or Y write control bit, the 'TL' is set in the timer latch only.

Also, the contents of the timer can be read by a read operation.

Count Operation

In the PWM mode, the high- and the low-order byte of a timer counts down each as an 8-bit timer.

- When the timer X or Y stop control bit is cleared to '0' in the HIGH state of the PWM output, only the high-order byte of the timer starts counting down (② in Figure1.12.24), while in the LOW state of the PWM output, only the low-order byte of the timer starts counting down (③ in Figure 1.12.24).
- When the stop control bit is set to '1', both the high- and the low-order byte stop counting down (④ in Figure 1.12.24).

In the count operation, the contents of the high- or the low-order byte of the timer are decremented by 1 at every rising edge of the count source (⑤ in Figure 1.12.24).

When either the high- or the low-order byte of an operating timer becomes '01_{16}', it stops counting. At the same time, the other starts counting (⑥ in Figure 1.12.24).

Reloading Timers

When either the high- or the low-order byte of an operating timer becomes '01_{16}', an underflow occurs at the subsequent rising edge of the count source, and the contents of the timer latch is reloaded to the timer (⑥ in Figure 1.12.24).

Timer Interrupt

At an rising edge of the PWM output waveform, the timer X or Y interrupt request bit is set to '1'; then a timer interrupt request is generated (⑦ in Figure 1.12.24).

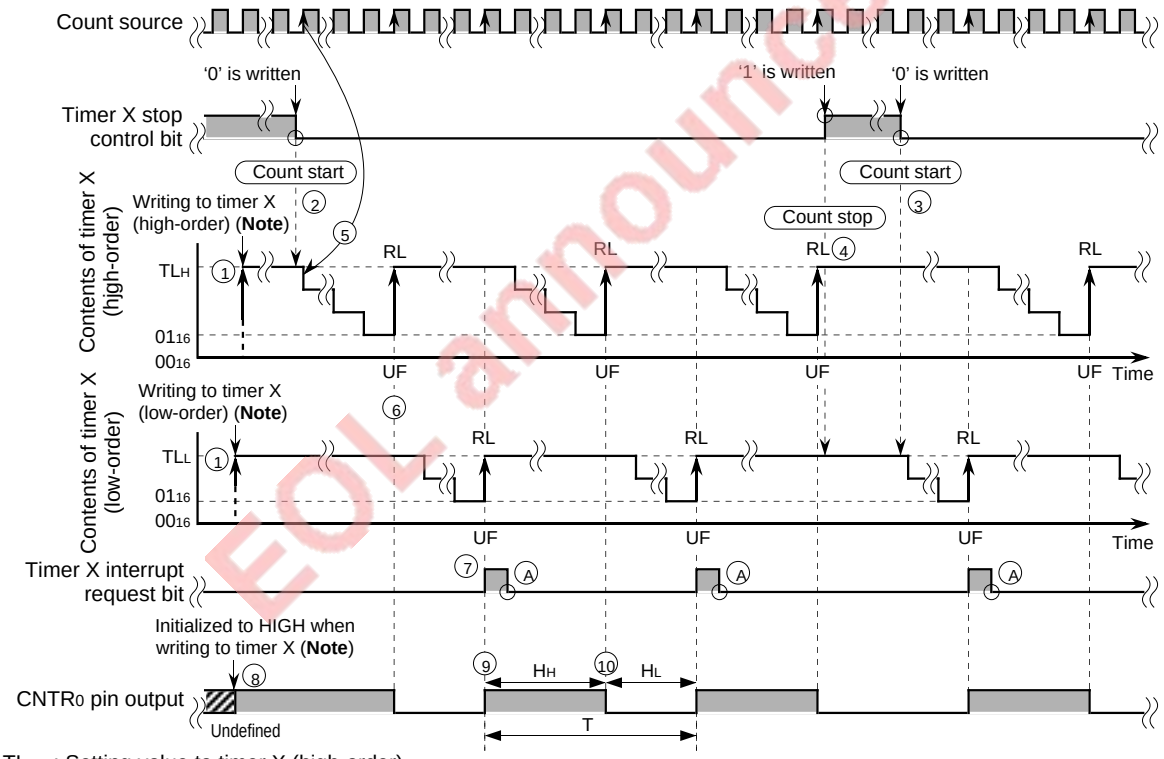
PWM Output

- When timer X is used, the PWM waveform is output from the CNTR0 pin.
- When timer Y is used, the PWM waveform is output from the CNTR1 pin.

When the timer X or Y write control bit is '0', the CNTR pin output is initialized to HIGH by a write to the timer (⑧ in Figure 1.12.24). When it is '1', however, the CNTR pin output level cannot be initialized by a write to the timer. In the PWM mode, when the low-order byte of the timer becomes '01$_{16}$' in the LOW level of the PWM output (⑨ in Figure 1.12.24), or when the high-order byte becomes '01$_{16}$' in the HIGH level of the PWM output (⑩ in Figure 1.12.24), an underflow occurs in each timer at the subsequent rising edge of the count source and the output level of the CNTR pin is inverted.

When 'TL_L (00$_{16}$ through FF$_{16}$)' is written to the low-order byte of the timer and 'TL_H (00$_{16}$ through FF$_{16}$)' to the high-order byte, the duty cycle of the PWM waveform output from the CNTR pin is expressed by '$TL_H/(TL_H + TL_L)$'.

Notes 1: • All of the PWM outputs are HIGH when TL_L = 00$_{16}$ and $TL_H \neq$ 00$_{16}$.
• All of the PWM outputs are LOW when TL_H = 00$_{16}$.
2: When at least one of TL_L and TL_H is '00$_{16}$', no timer interrupt request can be generated.
3: Even when value '00$_{16}$' is written to a timer, the timer continues counting down. Therefore, the contents of the timer are undefined.



TL_H : Setting value to timer X (high-order)
TL_L : Setting value to timer X (low-order)
T : PWM period T(s)=$H_H + H_L$
H_H : Pulse width (HIGH-period)
H_L : Pulse width (LOW-period)

RL: Contents of timer latch is reloaded
UF: Underflow

$$H_H(s) = \frac{1}{\text{Count source frequency}} \times \text{Setting value to timer X (high-order)}$$

$$H_L(s) = \frac{1}{\text{Count source frequency}} \times \text{Setting value to timer X (low-order)}$$

Ⓐ : Clearing by writing '0' to interrupt request bit or accepting interrupt request

Note: In this case, timer X write control bit is '0' (writing to timer and timer latch simultaneously).

Figure 1.12.24 Operation Example in PWM Mode

(2) Setting of PWM Mode

Figures 1.12.25 and 1.12.26 show the setting of the PWM mode.

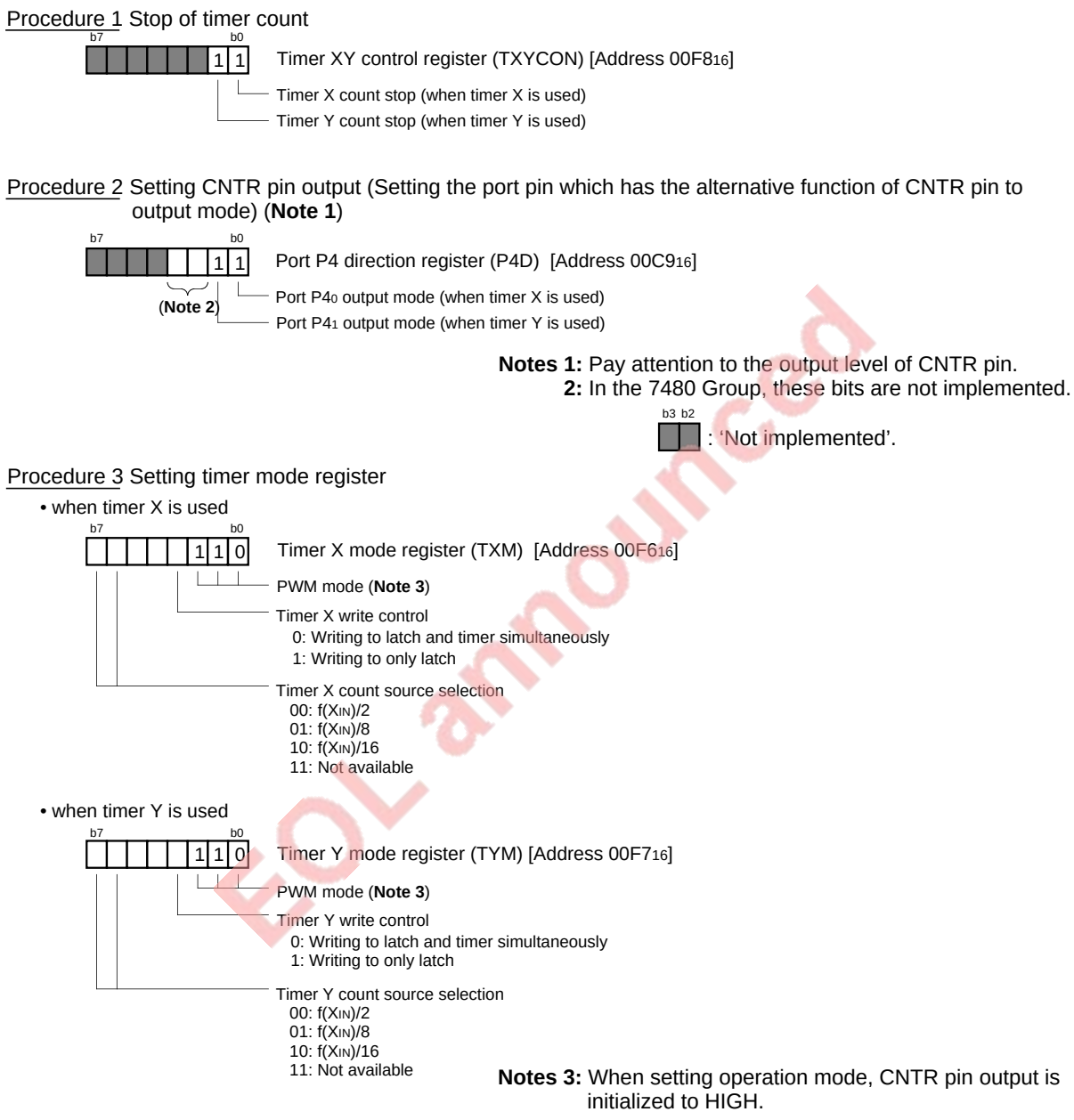


Figure 1.12.25 Setting of PWM Mode (1)

Procedure 4 Setting timer (**Note 4)**

• when timer X is used

Timer X low-order (TXL) [Address $00F0_{16}$]

Timer X low-order count value (LOW level output interval) is set

Timer X high-order (TXH) [Address $00F1_{16}$]

Timer X high-order count value (HIGH level output interval) is set

• when timer Y is used

Timer Y low-order (TYL) [Address $00F2_{16}$]

Timer Y low-order count value (LOW level output interval) is set

Timer Y high-order (TYH) [Address $00F3_{16}$]

Timer Y high-order count value (HIGH level output interval) is set

Notes 4: When writing to timer, set the low-order byte and high-order byte in this order.

Procedure 5 Start of timer count

Timer XY control register (TXYCON) [Address $00F8_{16}$]

Timer X count start (when timer X is used)

Timer Y count start (when timer Y is used)

Figure 1.12.26 Setting of PWM Mode (2)

1.12 Timer X and Timer Y

1.12.11 Notes on Usage

Pay attention to the following notes when timer X or Y is used.

(1) In All Modes

■ Write to and Read from Timers

Write to and read from each timer two bytes together in the following order:

- Write: ① low-order byte → ② high-order byte
- Read: ① high-order byte → ② low-order byte

When a read from and a write into the same timer are executed during an interrupt service routine etc., the normal operation cannot be performed.

In the pulse period measurement mode and the pulse width measurement mode, do not write to timers.

■ Writes to Timers

When the timer X or Y write control bit is '0':

- A write to an operating timer causes the contents of the timer to be affected, so that the time from the last underflow until the next underflow is undefined in this case.
- A write to the low-order byte of an operating timer allows the timer to continue counting down until the next write to the high-order byte. Therefore, the time until the subsequent underflow may be undefined.

Figure 1.12.27 shows an operation in timer X or timer Y at writes.

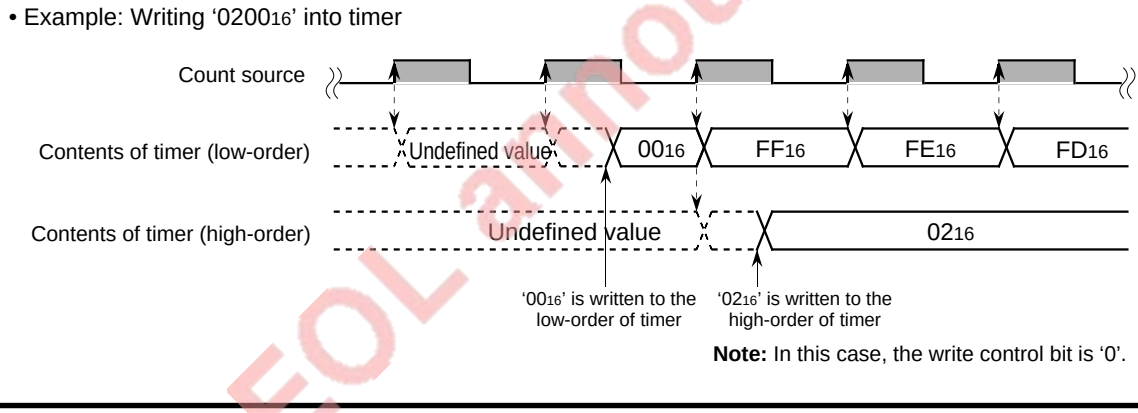


Figure 1.12.27 Operation in Timer X or Timer Y at Writes

When the timer X or Y write control bit is '1':

- A write to a stopped timer causes the contents of the timer not to be affected and allows the timer to count down from the value prior to this write. Therefore, the time from the start of count down until the first underflow is undefined.
- If a write and an underflow occur at approximately the same time in an operating timer, the reloaded value may be undefined.

■ Reads from Timers
- When the high-order byte of an operating timer is read, the low-order byte is set in the latch for reading. Therefore, the read value of the low-order byte retains the value at the time the high-order byte is being read.
- In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source, while the contents of each timer are transferred to the latch for reading by falling edge, so that the read value of the timer may be different from its real value.

Figure 1.12.28 shows an operation in timer X or timer Y at reads.

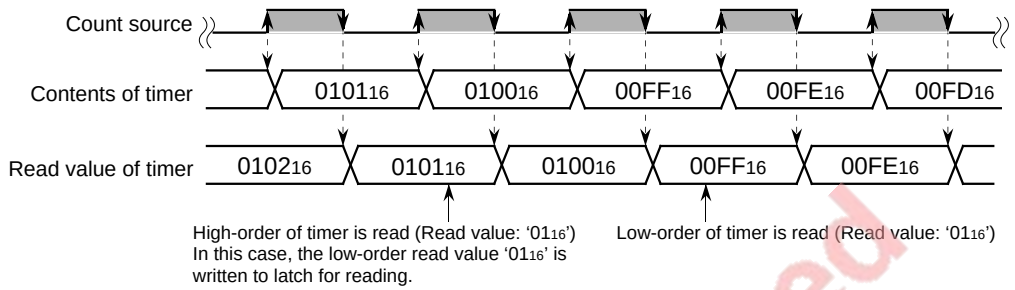


Figure 1.12.28 Operation in Timer X or Timer Y at Reads

(2) In Event Count Mode

■ The inverted signal of input to a CNTR pin is used as the count source when a CNTR edge selection bit of the edge polarity selection register is '1'.

■ Keep the frequency of the CNTR pin input used as the count source f(XIN)/4 or less.

(3) In Pulse Output Mode

■ When the timer X or Y write control bit is '0', the CNTR pin output is initialized to the following levels by a write to the timer:
- HIGH when the CNTR edge selection bit is '0'.
- LOW when the CNTR edge selection bit is '1'.

When the timer X or Y write control bit, however, is '1', the CNTR pin output level cannot be initialized by a write to the timer.

■ The output level of a CNTR pin is inverted when the CNTR edge polarity selection bit is switched.

(4) In Pulse Period Measurement Mode and Pulse Width Measurement Mode

■ Do not write to timers in these modes; otherwise the last measured value in the timer latch will be changed by this write.

■ When a timer is read, the read value is the contents of the timer latch (the last measured value).

(5) In Programmable Waveform Generation Mode

■ When the timer X or Y operation mode bits, which are set to other modes, are switched to the programmable waveform generation mode, the CNTR pin outputs are initialized to LOW.

■ When the timer X or Y trigger selection bit is '1', keep the trigger widths input to the INT pins 250ns or more.

(6) In Programmable One-Shot Output Mode

- When the timer X or Y operation mode bits are set to the programmable one-shot output mode, the CNTR pin outputs are initialized to the content of the CNTR edge selection bit.
- The output level of a CNTR pin is inverted when the CNTR edge selection bit is switched.
- Keep the trigger widths input to the INT pins 250 ns or more.

(7) In PWM Mode

- When the timer X or Y write control bit is '0', the CNTR pin output is initialized to HIGH by a write to the timer. When the write control bit is '1', the CNTR pin output level cannot be initialized by a write to the timer.
- All of the PWM outputs are HIGH when $TL_L = 00_{16}$ and $TL_H \neq 00_{16}$.
- All of the PWM outputs are LOW when $TL_H = 00_{16}$.
- When at least one of TL_L and TL_H is '00_{16}', no timer interrupt request can be generated.
- Even when value '00_{16}' is written to a timer, the timer continues counting down. Therefore, the contents of the timer are undefined.

(8) I/O Port Pins $P4_0$ and $P4_1$ with the Alternative Functions of Timer I/O Pins $CNTR_0$ and $CNTR_1$

Port pins $P4_0$ and $P4_1$ have the alternative functions of 16-bit timer I/O pins $CNTR_0$ and $CNTR_1$ respectively. If the timer X or Y operation mode bit of the corresponding timer is set to any mode related to output (**Note**), these pins cannot perform the normal function as output port pins. Refer to **Figure 1.10.3 Block Diagrams of Port Pins $P2_i$ to $P5_i$** in **Section 1.10. Input/Output Pins.**

Note: Modes related to output:
- Pulse output mode
- Programmable waveform generation mode
- Programmable one-shot output mode
- PWM mode

(9) Edge Polarity Selection Register

When the edge polarity selection bit of edge polarity selection register is set, the interrupt request bit may be set to '1'.
Refer to **Section 1.11.7 (2)** in **1.11 Interrupts**.

1.13 Timer 1 and Timer 2

The 7480 Group and 7481 Group have two 8-bit timers with 8-bit latches:
- Timer 1
- Timer 2

Timer 1 or timer 2 can select the following operation modes by the timer 1 or 2 operation mode bit of the timer 1 mode register (address 00F9$_{16}$) or the timer 2 mode register (address 00FA$_{16}$):
- Timer mode
- Programmable waveform generation mode

For details, refer to the section of each mode.

1.13.1 Block Diagram

Figure 1.13.1 shows the block diagram of timer 1 and timer 2.

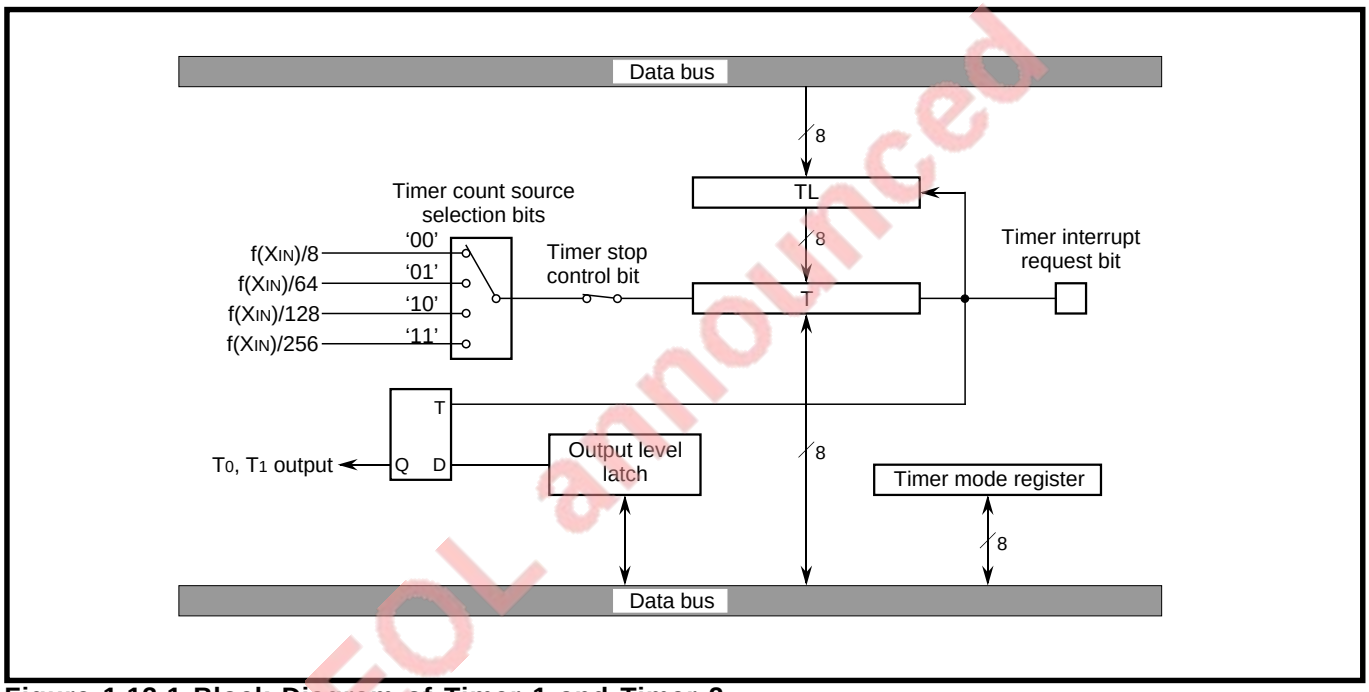


Figure 1.13.1 Block Diagram of Timer 1 and Timer 2

1.13.2 Registers Associated with Timer 1 and Timer 2

Figure 1.13.2 shows the memory map of the registers associated with timer 1 and timer 2.

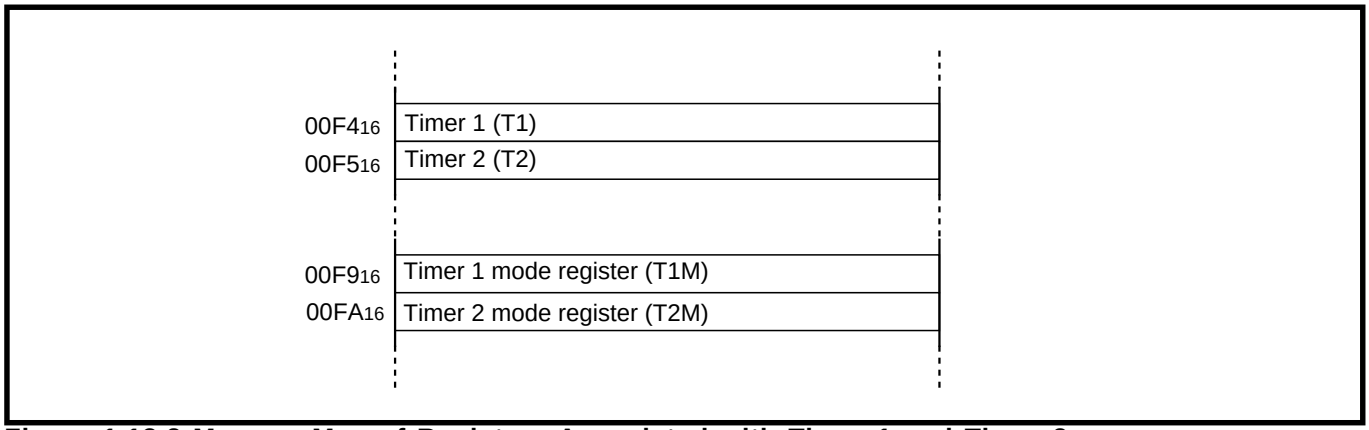


Figure 1.13.2 Memory Map of Registers Associated with Timer 1 and Timer 2

1.13 Timer 1 and Timer 2

(1) Timer 1 and Timer 2

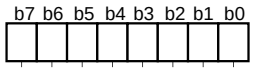
These are the 8-bit registers that count the pulses of count sources.

- When a timer is written, the written data is set to the timer and the timer latch (**Note**).
- When a timer is read, the read value is the contents of the timer.

Note: The timer latches are the registers that hold the initial values automatically reloaded to the timers when they underflow. Actually, the value decremented by 1 from the contents of the timer latch is reloaded to the timer.

Figures 1.13.3 and 1.13.4 show the timer 1 and timer 2.

Timer 1 (Timer 1 latch)

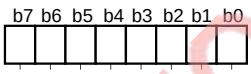


Timer1 (T1) [Address $00F4_{16}$]

b	Function	At reset	R	W
0	The timer 1 count value is indicated.	1	O	O
1		1	O	O
2		1	O	O
3		1	O	O
4		1	O	O
5		1	O	O
6		1	O	O
7		1	O	O

Figure 1.13.3 Timer 1

Timer 2 (Timer 2 latch)



Timer 2 (T2) [Address $00F5_{16}$]

b	Function	At reset	R	W
0	The timer 2 count value is indicated.	Undefined	O	O
1		Undefined	O	O
2		Undefined	O	O
3		Undefined	O	O
4		Undefined	O	O
5		Undefined	O	O
6		Undefined	O	O
7		Undefined	O	O

Figure 1.13.4 Timer 2

(2) Timer 1 Mode Register and Timer 2 Mode Register

These registers consist of the bits controlling the operation of timer 1 and timer 2. Figures 1.13.5 and 1.13.6 show the timer 1 and 2 mode registers.

Timer 1 mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0		0	0		

Timer 1 mode register (T1M) [Address $00F9_{16}$]

b	Name	Function	At reset	R	W
0	Timer 1 stop control bit	0 : Count operation 1 : Count stop	0	O	O
1	Timer 1 operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode	0	O	O
2	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
3			0	0	×
4	Output level latch	0 : LOW output from T_0 pin 1 : HIGH output from T_0 pin	0	O	O
5	Not implemented. Writing to this bit is disabled. This bit is '0' at reading.		0	0	×
6	Timer 1 count source selection bits	b7 b6 0 0 : $f(X_{IN})/8$ 0 1 : $f(X_{IN})/64$ 1 0 : $f(X_{IN})/128$ 1 1 : $f(X_{IN})/256$	0	O	O
7			0	O	O

Figure 1.13.5 Timer 1 Mode Register

Timer 2 mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0		0	0		

Timer 2 mode register (T2M) [Address $00FA_{16}$]

b	Name	Function	At reset	R	W
0	Timer 2 stop control bit	0 : Count operation 1 : Count stop	0	O	O
1	Timer 2 operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode	0	O	O
2	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
3			0	0	×
4	Output level latch	0 : LOW output from T_1 pin 1 : HIGH output from T_1 pin	0	O	O
5	Not implemented. Writing to this bit is disabled. This bit is '0' at reading.		0	0	×
6	Timer 2 count source selection bits	b7 b6 0 0 : $f(X_{IN})/8$ 0 1 : $f(X_{IN})/64$ 1 0 : $f(X_{IN})/128$ 1 1 : $f(X_{IN})/256$	0	O	O
7			0	O	O

Figure 1.13.6 Timer 2 Mode Register

1.13 Timer 1 and Timer 2

1.13.3 Basic Operations of Timer 1 and Timer 2

Basic operations of timer 1 and timer 2 are described below.
For details, refer to **(1) Operations** of each mode.

Count Sources

Timer 1 and timer 2 can select the following count sources with the timer 1 or 2 count source select bits of the timer 1 or 2 mode register:

- f(XIN)/8
- f(XIN)/64
- f(XIN)/128
- f(XIN)/256

Writes to Timers

When 'TL(00_{16} through FF_{16})' is written to a timer, 'TL' is set in both the timer and the timer latch.

Note: A write to an operating timer causes the contents of the timer to be affected, so that the period from the last underflow until the next underflow is undefined.

Reads from Timers

The contents of the timer can be read by a read operation.

Count Operations

The count operation (start/stop) of timer 1 or timer 2 is controlled by the timer 1 or 2 stop control bit of the timer 1 or 2 mode register as follows:

- When the timer 1 or 2 stop control bit is set to '0', the timer starts counting.
- When the timer 1 or 2 stop control bit is set to '1', the timer stops counting.

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source.
The timer 1 or 2 stop control bit is recognized during the HIGH time of the count source. When the count has stopped, the count source cannot be accepted.

Reloading Timers

When a timer reaches 'FF_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source, and the value decremented by 1 from the contents of the timer latch is reloaded to the timer.

Timer Interrupt

At an underflow, the timer 1 or 2 interrupt request bit of interrupt request register 1 is set to '1'; then a timer interrupt request is generated.

Table 1.13.1 lists the relation between timer count periods and values set to timer 1 and timer 2.

Table 1.13.1 Relation between Timer Count Periods and Values Set to Timer 1 and Timer 2

Clock Input Oscillation Frequency		f(XIN) = 8 MHz				f(XIN) = 4 MHz		
Count Source (Count Period)		f(XIN)/8 (1μs)	f(XIN)/64 (8μs)	f(XIN)/128 (16μs)	f(XIN)/256 (32μs)	f(XIN)/8 (2μs)	f(XIN)/64 (16μs)	f(XIN)/128 (32μs)
Timer Period	100 μs	63_{16}	—	—	—	31_{16}	—	—
	200 μs	$C7_{16}$	18_{16}	—	—	63_{16}	—	—
	500 μs	—	—	—	—	$F9_{16}$	—	—
	1 ms	—	$7C_{16}$	—	—	—	—	—
	2 ms	—	$F9_{16}$	$7C_{16}$	—	—	$7C_{16}$	—
	4 ms	—	—	$F9_{16}$	$7C_{16}$	—	$F9_{16}$	$7C_{16}$

1.13.4 Timer Mode

(1) Operations in Timer Mode

The operations in the timer mode is explained with Figure 1.13.7.

Count Sources

In the timer mode, timer 1 or timer 2 can select the following count sources with the timer 1 or 2 count source selection bits of the timer 1 or 2 mode register:

- f(XIN)/8
- f(XIN)/64
- f(XIN)/128
- f(XIN)/256

Writes to and Reads from Timer

When 'TL (00_{16} through FF_{16})' is written to a timer, 'TL' is set in both the timer and the timer latch (① in Figure 1.13.7).
Also, the contents of the timer can be read by a read operation.

Count Operation

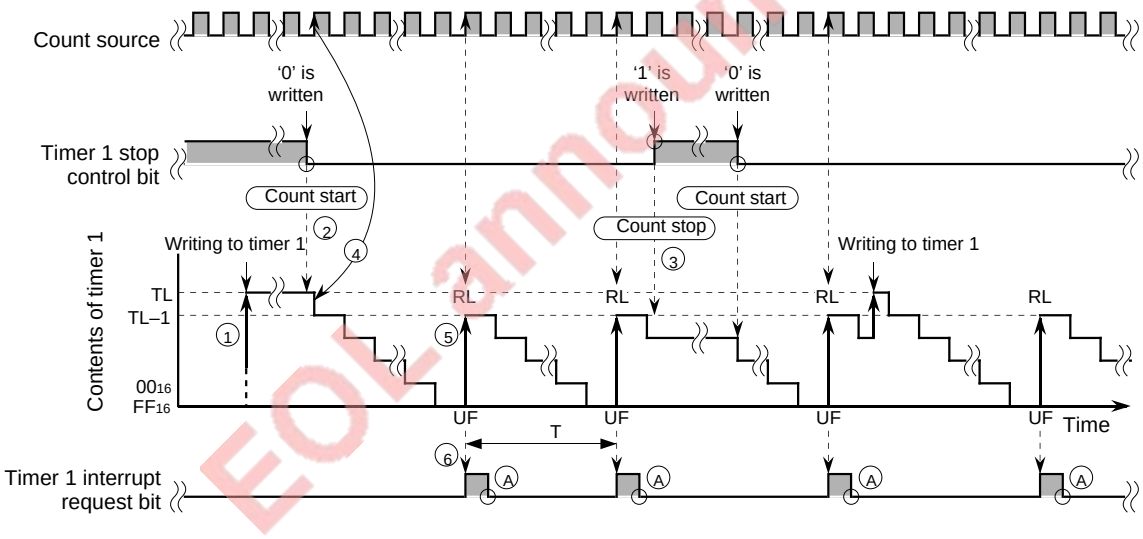
• When the timer 1 or 2 stop control bit is cleared to '0', the timer starts counting (② in Figure 1.13.7).
• When the timer 1 or 2 stop control bit is set to '1', the timer stops counting (③ in Figure 1.13.7).
In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source (④ in Figure 1.13.7).

Reloading Timers

When a timer reaches 'FF_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source, and the value decremented by 1 from the contents of the timer latch is reloaded to the timer. (⑤ in Figure 1.13.7).

Timer Interrupt

At an underflow, the timer 1 or 2 interrupt request bit is set to '1', then a timer interrupt request is generated (⑥ in Figure 1.13.7).



TL : Setting value to timer 1
RL : The value which is the timer latch decremented by 1 is reloaded
UF : Underflow
T : Count period $T(s) = \frac{1}{\text{Count source frequency}} \times$ (Setting value to timer 1 +1)

Ⓐ : Clearing by writing '0' to interrupt request bit or accepting interrupt request

Figure 1.13.7 Operations in Timer Mode

(2) Setting of Timer Mode

Figure 1.13.8 shows the setting of the timer mode.

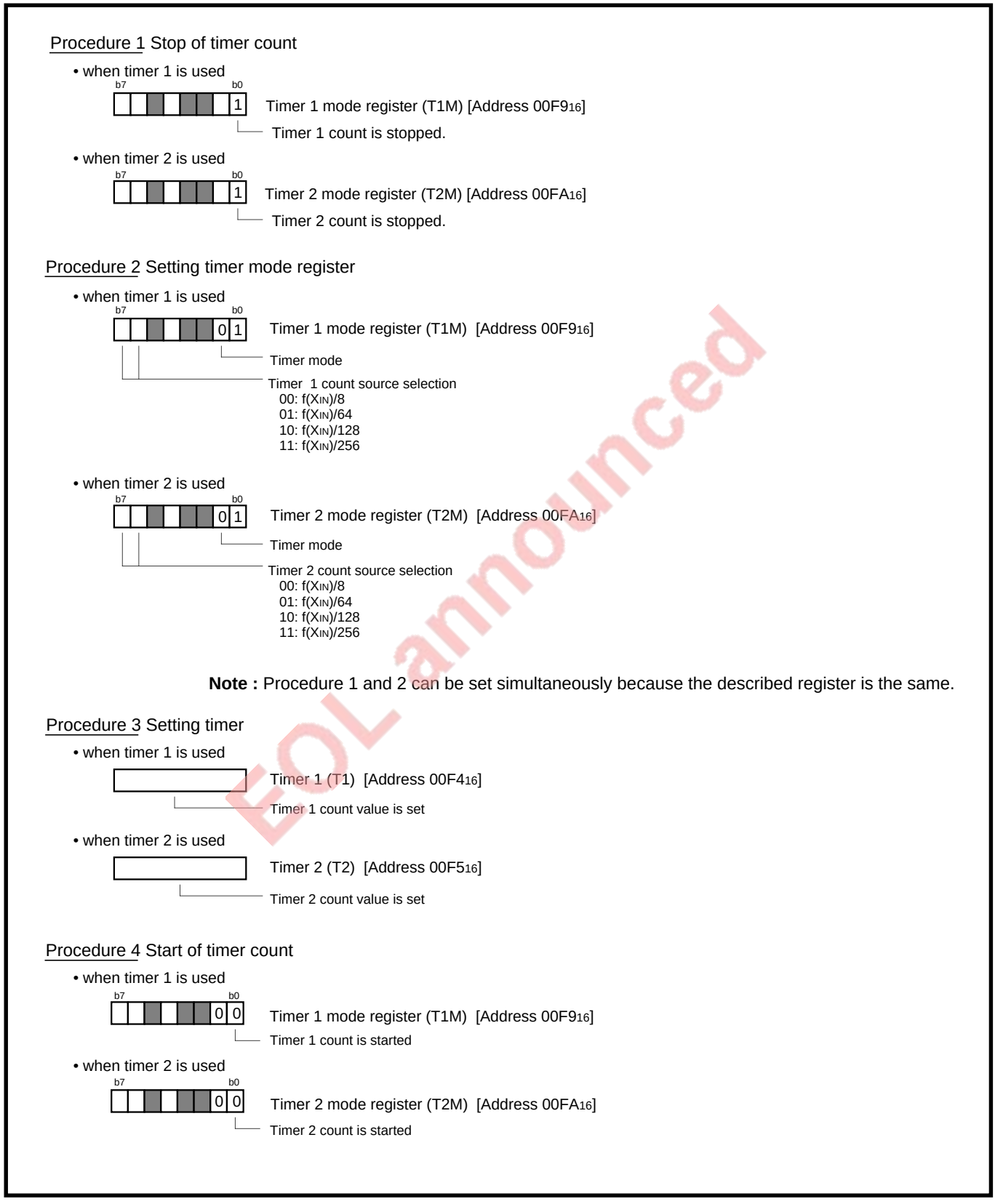


Figure 1.13.8 Setting of Timer Mode

1.13.5 Programmable Waveform Generation Mode

(1) Operations in Programmable Waveform Generation Mode

Operations in the programmable waveform generation mode is explained with Figure 1.13.9.

Count Sources

In the programmable waveform generation mode, timer 1 or timer 2 can select the following count sources with the timer 1 or 2 count source selection bits:

- f(XIN)/8
- f(XIN)/64
- f(XIN)/128
- f(XIN)/256

Writes to and Reads from Timers

When 'TL(00_{16} through FF_{16})' is written to a timer, 'TL' is set in both the timer and the timer latch (① in Figure 1.13.9).

Also, the contents of the timer can be read by a read operation.

Count Operation

- When the timer 1 or 2 stop control bit is cleared to '0', the timer starts counting (② in Figure 1.13.9).
- When the timer 1 or 2 stop control bit is set to '1', the timer stops counting (③ in Figure 1.13.9).

In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source (④ in Figure 1.13.9).

Reloading timers

When a timer reaches 'FF_{16}' in the count operation, an underflow occurs at the subsequent rising edge of the count source, and the value decremented by 1 from the contents of the timer latch is reloaded to the timer. (⑤ in Figure 1.13.9).

Timer interrupt

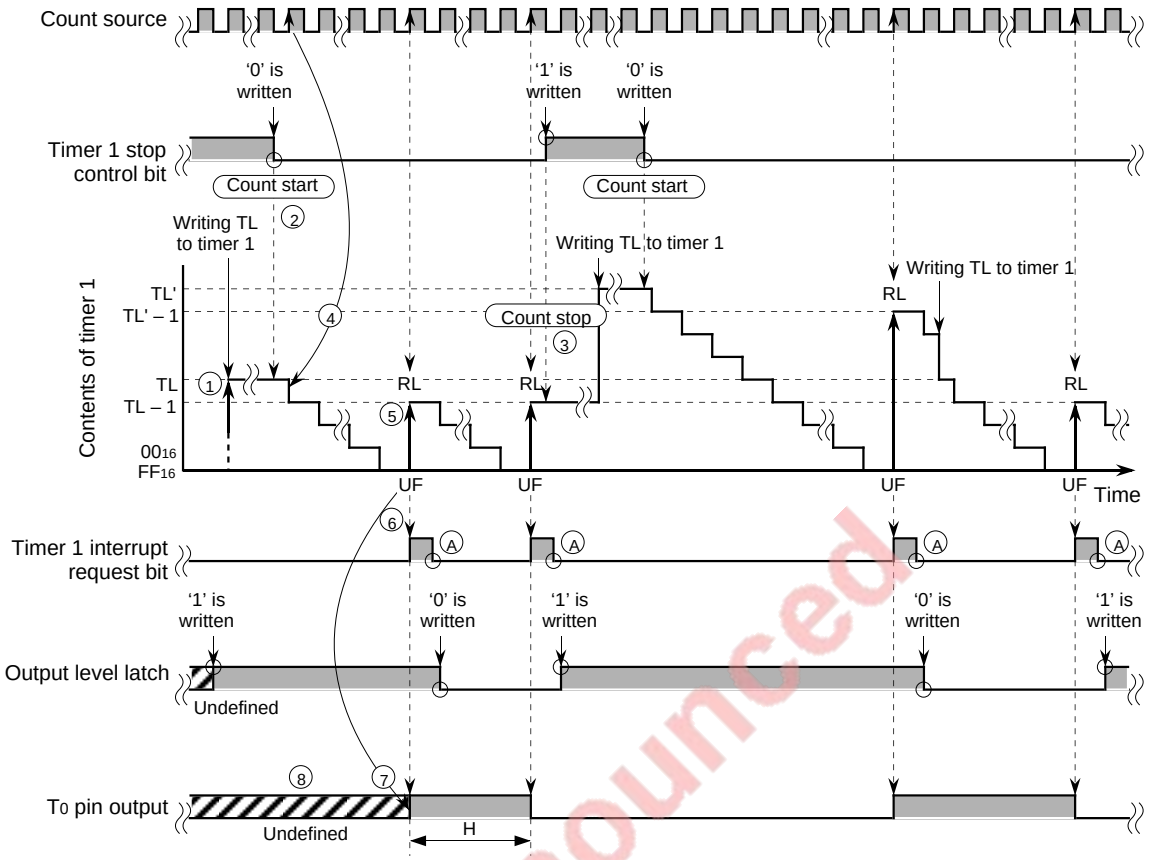
At an underflow, the timer 1 or 2 interrupt request bit is set to '1'; then a timer interrupt request is generated (⑥ in Figure 1.13.9).

Generation of Programmable Waveform

When an underflow occurs in a timer, the contents of the output level latch are output from the following pins (⑦ in Figure 1.13.9):

- T0 pin (Timer 1 used)
- T1 pin (Timer 2 used)

The output level of the T0 or T1 pin remains undefined until the first underflow occurs in this mode (⑧ in Figure 1.13.9).



Ⓐ : Clearing by writing '0' to interrupt request bit or accepting interrupt request

TL : Setting value to timer 1
RL : The value which is the timer latch decremented by 1 is reloaded
UF : Underflow
H : Pulse width $H(s) = \frac{1}{\text{Count source frequency}} \times (\text{Setting value to timer 1} + 1)$

Figure 1.13.9 Operations in Programmable Waveform Generation Mode

(2) Setting of Programmable Waveform Generation Mode

Figures 1.13.10 and 1.13.11 show the setting of the programmable waveform generation mode.

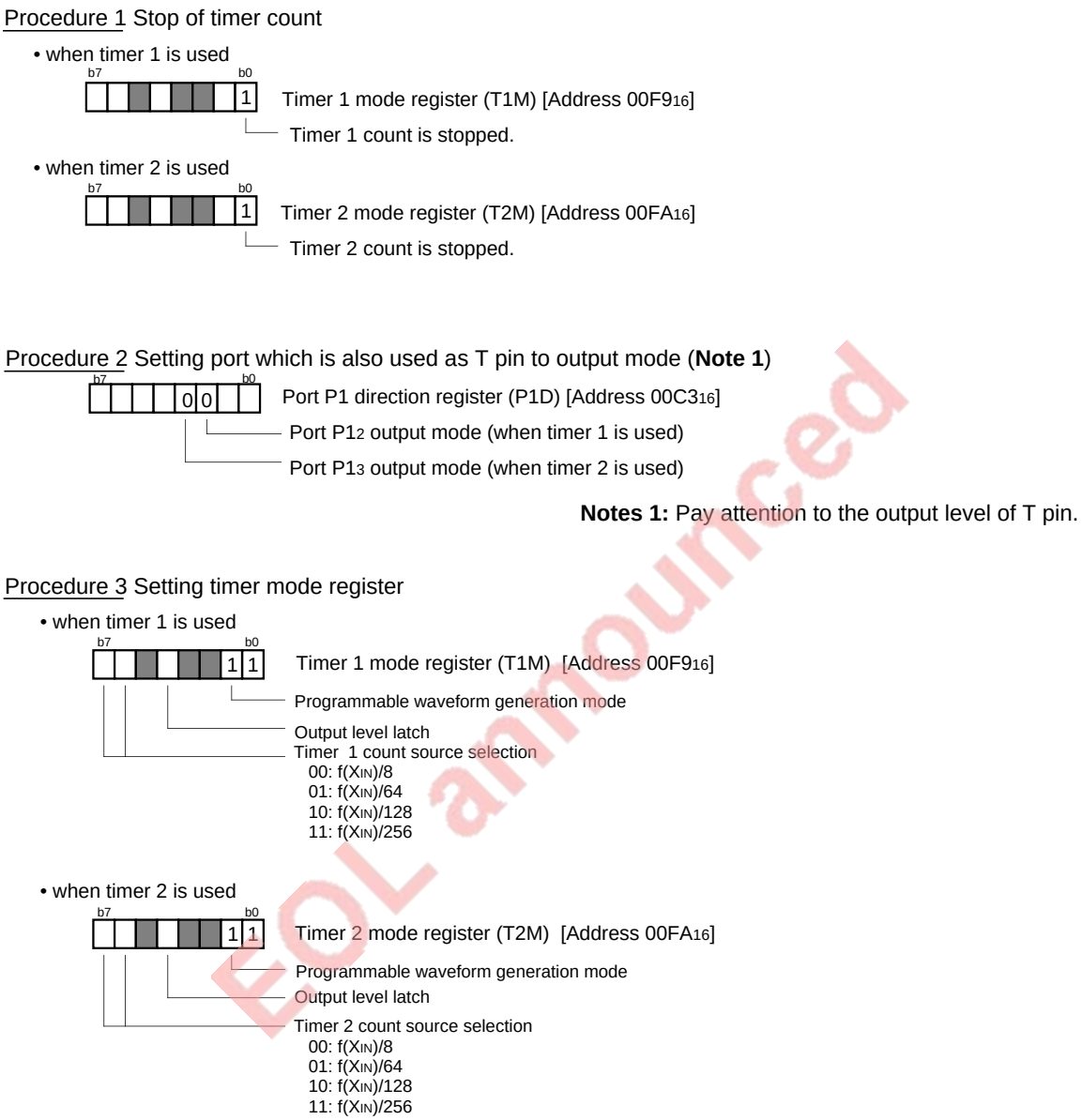


Figure 1.13.10 Setting of Programmable Waveform Generation Mode (1)

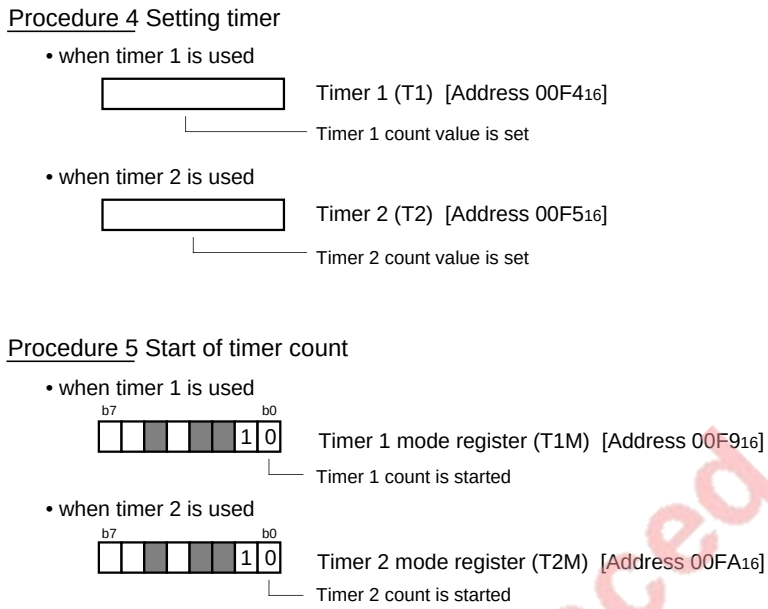


Figure 1.13.11 Setting of Programmable Waveform Generation Mode (2)

1.13.6 Notes on Usage

Pay attention to the following notes when timer 1 or timer 2 is used.

(1) In All Modes

■ A write to an operating timer causes the contents of the timer to be affected, so that the period from the last underflow until the next underflow is undefined.

■ In the count operation, the contents of each timer are decremented by 1 at every rising edge of the count source, while the contents of each timer are transferred to the latch for reading at the falling edge, so that the read value of the timer may be different from its real value by +1.

Figure 1.13.12 shows an operation in timer 1 and timer 2 at reads.

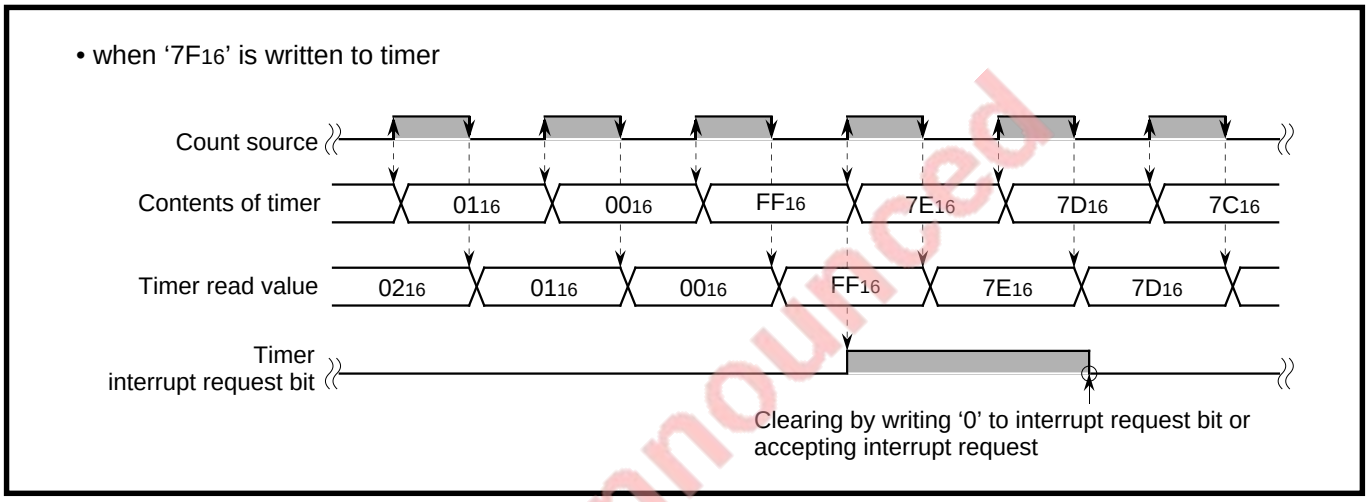


Figure 1.13.12 Operations in Timer 1 and Timer 2 at Reads

(2) I/O Port Pins P12 and P13 with the Alternative Functions of Timer Output Pins T0 and T1

Port pins P12 and P13 have the alternative functions of timer output pins T0 and T1 respectively. If the timer operation mode bit of the corresponding timer (1 or 2) is set to the programmable waveform generation mode, these pins cannot perform the normal function as output port pins. Refer to **Figure 1.10.1 Block Diagrams of Port Pins P0i and P10–P13** in **Section 1.10 Input/Output pins**. Therefore, set the corresponding timer (1 or 2) operation mode bit to the timer mode when these pins are used as normal I/O port pins.

1.14 Serial I/O

Serial I/O transmits or receives 8-bit data serially, between two microcomputers.
The serial I/O of the 7480 Group and 7481 Group can operate with a transmission format of either synchronous or asynchronous (UART) type.
If data is not sent on the transmission line because of collision. The microcomputer informs external the collision in the contention bus system communications by generating a bus arbitration interrupt request.

1.14.1 Registers Associated with Serial I/O

Figure 1.14.1 shows the memory map of the registers associated with serial I/O.

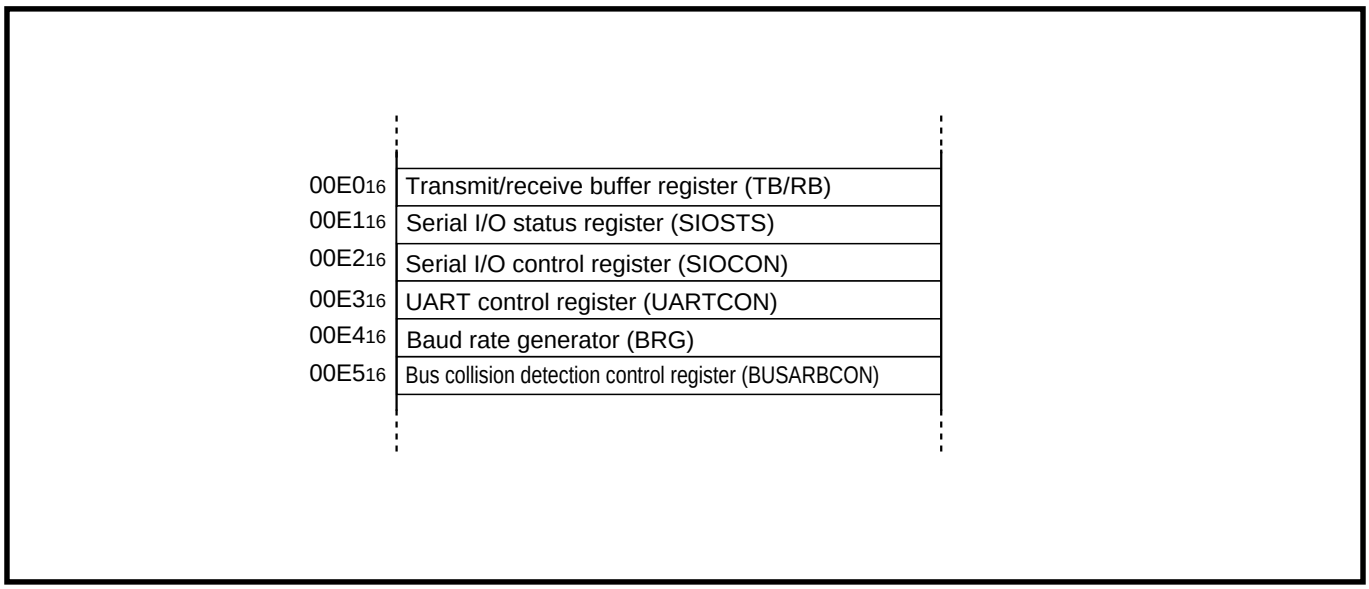


Figure 1.14.1 Memory Map of Registers Associated with Serial I/O

1.14 Serial I/O

(1) Transmit Buffer Register and Receive Buffer Register

The transmit buffer register and the receive buffer register are located at the same address. These registers are written transmit data and read receive data when clock synchronous or clock asynchronous serial I/O is used.

Clock Synchronous Serial I/O

A write to the transmit buffer register (**Note**) starts the following operations:

- When the BRG output/4 is selected as the synchronous clock, communication is started.
- When an external clock is selected as the synchronous clock and the $\overline{SRDY}$ output is in the enable state, the level of the $\overline{SRDY}$ signal changes from HIGH to LOW, and the completion of the communication preparation is signaled to the external.

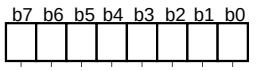
Clock Asynchronous Serial I/O (UART)

A write to the transmit buffer register (**Note**) starts data transmission.

Note: During transmission, data is written to the transmit buffer register.
During reception, dummy data is written to the transmit buffer register when the clock synchronous serial I/O is selected.

Figure 1.14.2 shows the transmit/receive buffer register.

Transmit/Receive buffer register



Transmit/Receive buffer register (TB/RB) [Address $00E0_{16}$]

b	Function	At reset	R W
0	In transmission: Transmit data is transferred to the transmit shift register by writing transmit data. In reception: When data is stored completely in the receive shift register, the receive data is transferred to this register.	Undefined	Note
1		Undefined	
2		Undefined	
3		Undefined	
4		Undefined	
5		Undefined	
6		Undefined	
7		Undefined	

Note: In transmission, this register is a write-only register.
In reception, this register is a read-only register.

Figure 1.14.2 Transmit/Receive Buffer Register

(2) Serial I/O Status Register

This register consists of the flags that indicate the serial I/O transmit/receive status. Figure 1.14.3 shows the serial I/O status register.

Serial I/O status register

b7 b6 b5 b4 b3 b2 b1 b0

1 | | | | | | | |

Serial I/O status register (SIOSTS) [Address $00E1_{16}$]

b	Name	Function	At reset	R	W
0	Transmit buffer empty flag (TBE)	0 : Buffer full 1 : Buffer empty	0	O	×
1	Receive buffer full flag (RBF)	0 : Buffer empty 1 : Buffer full	0	O	×
2	Transmit shift completion flag (TSC)	0 : Transmit shift in progress 1 : Transmit shift completed	0	O	×
3	Overrun error flag (OE)	0 : No error 1 : Overrun error	0	O	×
4	Parity error flag (PE)	0 : No error 1 : Parity error	0	O	×
5	Framing error flag (FE)	0 : No error 1 : Framing error	0	O	×
6	Summing error flag (SE)	0 : OE U PE U FE=0 1 : OE U PE U FE=1	0	O	×
7	This bit is fixed to '1'.		1	1	×

Note: b4 and b5 are valid only in UART

Figure 1.14.3 Serial I/O Status Register

Each flag of the serial I/O status register is described below.

Transmit Buffer Empty Flag (TBE: bit 0)

This flag indicates the status of the transmit buffer register.

- When the data written to the transmit buffer register is transferred to the transmit shift register, this flag is set to '1'.
- When transmit data is written to the transmit buffer register, this flag is cleared to '0'.

This flag is valid in both clock synchronous serial I/O and UART.

Receive Buffer Full Flag (RBF: bit 1)

This flag indicates the status of the receive buffer register.

- When receive data is stored completely in the receive shift register and transferred to the receive buffer register, this flag is set to '1'.
- When the transferred data is read from the receive buffer register, this flag is cleared to '0'.

This flag is valid in both clock synchronous serial I/O and UART.

Transmit Shift Completion Flag (TSC: bit 2)

This flag indicates the status of the transmit shift operation.

- When the data in the transmit buffer register is transferred to the transmit shift register, and shift operation is started by the synchronous clock (the start bit of the transmit data is transmitted), this flag is cleared to '0'.
- When the shift operation is completed (the transmission of the last bit of the transmit data is completed), this flag is set to '1'.

This flag is valid in both clock synchronous serial I/O and UART.

Overrun Error Flag (OE: bit 3)

This flag indicates the status of reading receive data.

- When the next receive data is stored completely in the receive shift register before the receive data stored in the receive buffer register is read through, this flag is set to '1'.
- This flag is cleared to '0' by any operations listed in Table 1.14.1.

This flag is valid in both clock synchronous serial I/O and UART.

Parity Error Flag (PE: bit 4)

This flag indicates the result of checking even or odd parity by hardware in UART.

- This flag is set to '1' when the parity of the receive data differs from the predetermined parity.
- This flag is cleared to '0' by any operation listed in Table 1.14.1.

This flag is valid only at parity enable in UART.

Framing Error Flag (FE: bit 5)

This flag indicates faults of frame synchronization in UART.

- When the stop bit of receive data is not received at the specified timing, this flag is set to '1'. Only the first stop bit is tested and the second stop bit is not tested.
- This flag is cleared to '0' by any operation listed in Table 1.14.1.

This flag is valid only in UART.

Summing Error Flag (SE: bit 6)

This flag indicates faults of serial I/O.

- When the overrun error, parity error or framing error occurs, this flag is set to '1'.
- This flag is cleared to '0' by any operation listed in Table 1.14.1.

This flag is valid in both clock synchronous serial I/O and UART.

[Clearing Error Flag]

Error flags (bits 3 to 6) of the serial I/O status register are cleared to '0' by any operation listed in Table 1.14.1.

Table 1.14.1 Clearing Error Flags

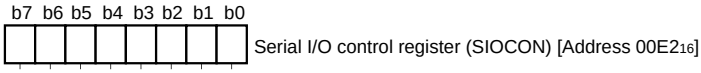
Clearing Method / Error Flag	Set Serial I/O Enable Bit to '0'	Set Receive Enable Bit to '0'	Dummy Data is Written to Serial I/O Status Register
Overrun Error Flag	O	O	O
Parity Error Flag	×	O	O
Framing Error Flag	×	O	O
Summing Error Flag	×	O	O

(3) Serial I/O Control Register

This register controls the selection of a transmit/receive mode, a synchronous clock, serial I/O pin functions, etc. of serial I/O.

Figure 1.14.4 shows the serial I/O control register.

Serial I/O control register



b	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : f(XIN)/4 1 : f(XIN)/16	0	O	O
1	Serial I/O synchronous clock selection bit (SCS)	when clock synchronous serial I/O is selected 0 : BRG output/4 1 : External clock input when UART is selected 0 : BRG output/16 1 : External clock input/16	0	O	O
2	$\overline{SRDY}$ output enable bit (SRDY)	0 : P17 pin 1 : $\overline{SRDY}$ output pin	0	O	O
3	Transmit interrupt source selection bit (TIC)	0 : Interrupt occurs when transmit buffer is empty. 1 : Interrupt occurs when transmit shift operation is completed.	0	O	O
4	Transmit enable bit (TE)	0 : Transmit disabled 1 : Transmit enabled	0	O	O
5	Receive enable bit (RE)	0 : Receive disabled 1 : Receive enabled	0	O	O
6	Serial I/O mode selection bit (SIOM)	0 : Asynchronous serial I/O (UART) 1 : Clock synchronous serial I/O	0	O	O
7	Serial I/O enable bit (SIOE)	0 : Serial I/O disabled 1 : Serial I/O enabled	0	O	O

Figure 1.14.4 Serial I/O Control Register

(4) UART Control Register

This register controls the data transmission formats in clock asynchronous serial I/O (UART). This register is valid only when UART is selected.

Figure 1.14.5 shows the UART control register.

UART control register

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

UART control register (UARTCON) [Address $00E3_{16}$]

b	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	O	O
1	Parity enable bit (PARE)	0 : Parity disabled 1 : Parity enabled	0	O	O
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	O	O
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	O	O
4	These bits are fixed to '1'.		1	1	×
5			1	1	×
6			1	1	×
7			1	1	×

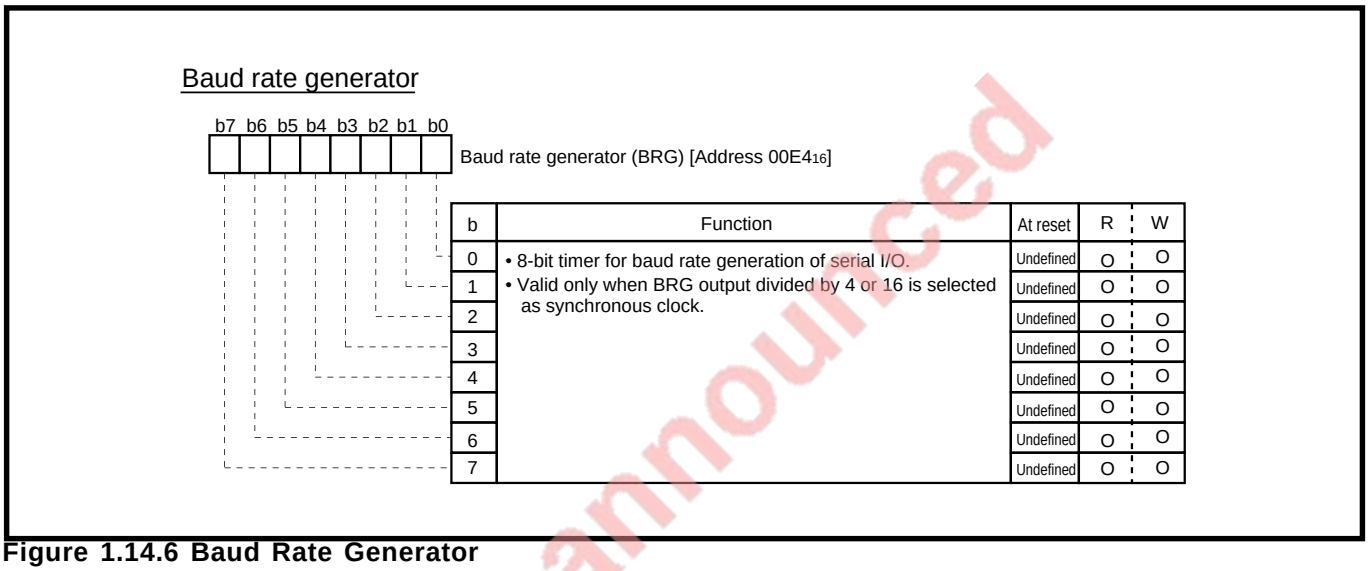
Figure 1.14.5 UART Control Register

(5) Baud Rate Generator (BRG)

The baud rate generator is an 8-bit counter with an auto-reload register, used only for serial I/O. When the serial I/O synchronous clock selection bit of the serial I/O control register is '0', setting value 'n' (any number of 00_{16} to FF_{16}) to the baud rate generator outputs a signal of the BRG count source (**Note 1**) divided by 'n + 1' as the BRG output (**Note 2**) .

Notes 1: • f(XIN)/4: when the BRG count source selection bit of the serial I/O control register is '0'.
• f(XIN)/16: when the BRG count source selection bit is '1'.
2: • BRG output/4 is used for the synchronous clock in clock synchronous serial I/O.
• BRG output/16 is used for the synchronous clock in clock asynchronous serial I/O.

Figure 1.14.6 shows the baud rate generator.

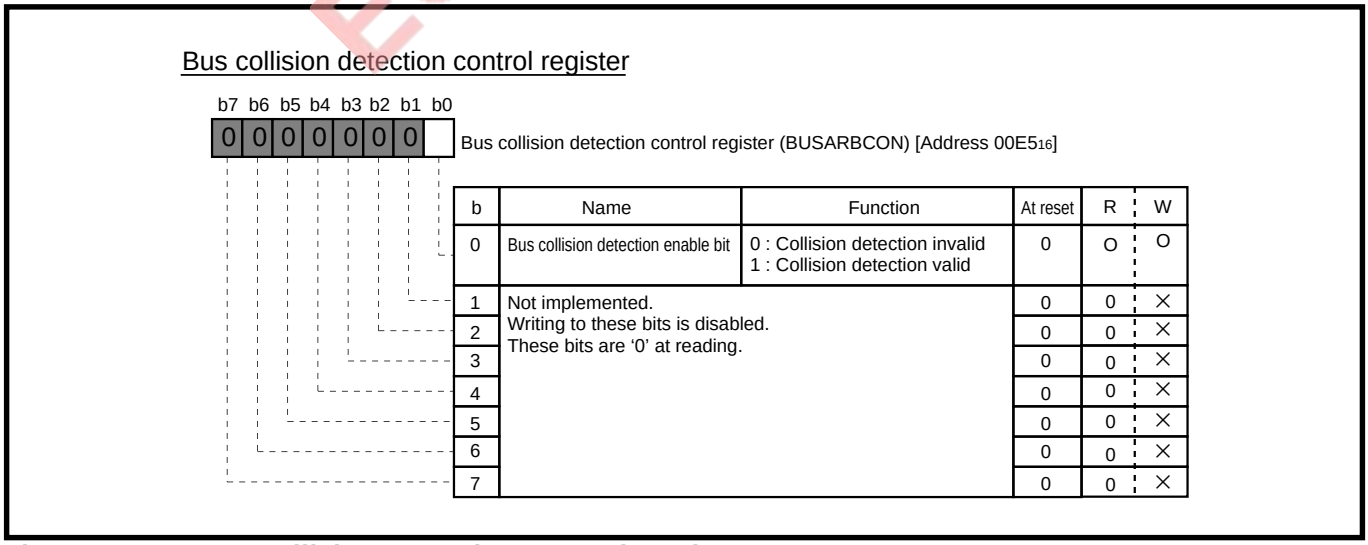


b	Function	At reset	R	W
0	• 8-bit timer for baud rate generation of serial I/O. • Valid only when BRG output divided by 4 or 16 is selected as synchronous clock.	Undefined	O	O
1		Undefined	O	O
2		Undefined	O	O
3		Undefined	O	O
4		Undefined	O	O
5		Undefined	O	O
6		Undefined	O	O
7		Undefined	O	O

Figure 1.14.6 Baud Rate Generator

(6) Bus Collision Detection Control Register

This register consists of the bits controlling the valid/invalid of the bus collision detection. Figure 1.14.7 shows the bus collision detection control register.



b	Name	Function	At reset	R	W
0	Bus collision detection enable bit	0 : Collision detection invalid 1 : Collision detection valid	0	O	O
1	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
2			0	0	×
3			0	0	×
4			0	0	×
5			0	0	×
6			0	0	×
7			0	0	×

Figure 1.14.7 Bus Collision Detection Control Register

1.14.2 Clock Synchronous Serial I/O

In clock synchronous serial I/O, the transmit operation of the transmitter (**Note 1**) and the receive operation of the receiver (**Note 2**) are performed simultaneously, synchronizing with the synchronous clock used for transferring, which is generated by the clock control circuit.

Notes 1: Synchronized with falling edges of the synchronous clock, data is transmitted from the TxD pin of the transmitter by the bit.
2: Synchronized with rising edges of the synchronous clock, data is received from the RxD pin of the receiver by the bit.

Clock synchronous serial I/O is selected by setting the serial I/O mode selection bit of the serial I/O control register to '1'.

Data Communication

- Half-duplex communication: one of the two communicating microcomputers operates only as a transmitter and the other only as a receiver at a time or vice versa.
- Full-duplex communication: both of the two communicating microcomputers operate simultaneously as transmitter and receiver.

Synchronous Clock

A synchronous clock is selected by the serial I/O synchronous clock selection bit of the serial I/O control register as follows:

- 0: BRG output/4
- 1: External clock input to the SCLK pin

For the BRG output, refer to **(5) Baud Rate Generator (BRG)** in **Section 1.14.1.**

When a clock synchronous serial I/O communication is carried out between two microcomputers, the synchronous clock is normally selected as follows:

- Microcomputer 1 clears the serial I/O synchronous clock selection bit to '0', and 8 synchronous clock pulses, generated by writing to the transmit buffer register, are output from the SCLK pin.
- Microcomputer 2 selects the external clock and inputs the pulses outputted from microcomputer 1 to the SCLK pin. This is the synchronous clock.

Note: When an external clock is selected as the synchronous clock:

- Perform the following operations while the SCLK pin input is HIGH during data transmission:
  - Write '1' to the transmit enable bit
  - Write transmit data to the transmit buffer register
- The shift operations of the transmit shift register and the receive shift register are performed while the synchronous clock is being input to the serial I/O circuit. Stop the synchronous clock with 8 cycles when an external clock is selected as the synchronous clock. The synchronous clock automatically stops after 8 synchronous clock pulses generated when the BRG output/4 is selected as the synchronous clock.

Data Transfer Rate (Baud Rate)

In clock synchronous serial I/O, the data transfer rate (baud rate), which is the frequency of the synchronous clock, is calculated by the following formulas:

- When the serial I/O synchronous clock selection bit is '0'.
  (BRG output/4 is selected as the synchronous clock)

$$\text{Baud rate [bps]} = \frac{f(X_{IN})}{\text{Division ratio (\textbf{Note 1}) } \times \text{ (BRG setting value (\textbf{Note 2}) + 1) } \times 4}$$

Notes 1: BRG count source selection bit of the serial I/O control register is as follows:
- '0': Division ratio is 4
- '1': Division ratio is 16.

2: The value written to the baud rate generator (00_{16} to FF_{16}).

- When the serial I/O synchronous clock selection bit is '1'
  (an external clock input is selected as the synchronous clock):

  Baud rate [bps] = the external clock input frequency from the SCLK pin

Output of $\overline{\text{SRDY}}$ Signal

In clock synchronous serial I/O, the output level of the $\overline{\text{SRDY}}$ pin changes from HIGH to LOW by writing to the transmit buffer register when the $\overline{\text{SRDY}}$ output enable bit of serial I/O control register is '1'. The completion of the serial I/O communication preparation is signaled to the external by the $\overline{\text{SRDY}}$ output. Also, the $\overline{\text{SRDY}}$ pin returns to the HIGH state at the first falling edge of the synchronous clock.

Note: Set the transmit enable bit to '1' as well as the receive enable bit and the $\overline{\text{SRDY}}$ output enable bit of the serial I/O control register when the receiver outputs the $\overline{\text{SRDY}}$ signal while the external clock is selected as the synchronous clock.

Starting of Transmission and Reception

- When the BRG output/4 is selected as the synchronous clock:
  Transmitting and receiving starts when a write to the transmit buffer register occurs.
  Normally, communication is started after the completion of communication preparation of the target unit is recognized with the $\overline{\text{SRDY}}$ signal.
- When the external clock is selected as the synchronous clock:
  Transmitting and receiving starts when input to the external clock starts.
  When data is written to the transmit buffer register, the output level of the $\overline{\text{SRDY}}$ pin changes from HIGH to LOW and informs the target unit of the completion of communication preparation.

1.14 Serial I/O

(1) Block Diagram of Clock Synchronous Serial I/O

Figure 1.14.8 shows the block diagram of a clock synchronous serial I/O.

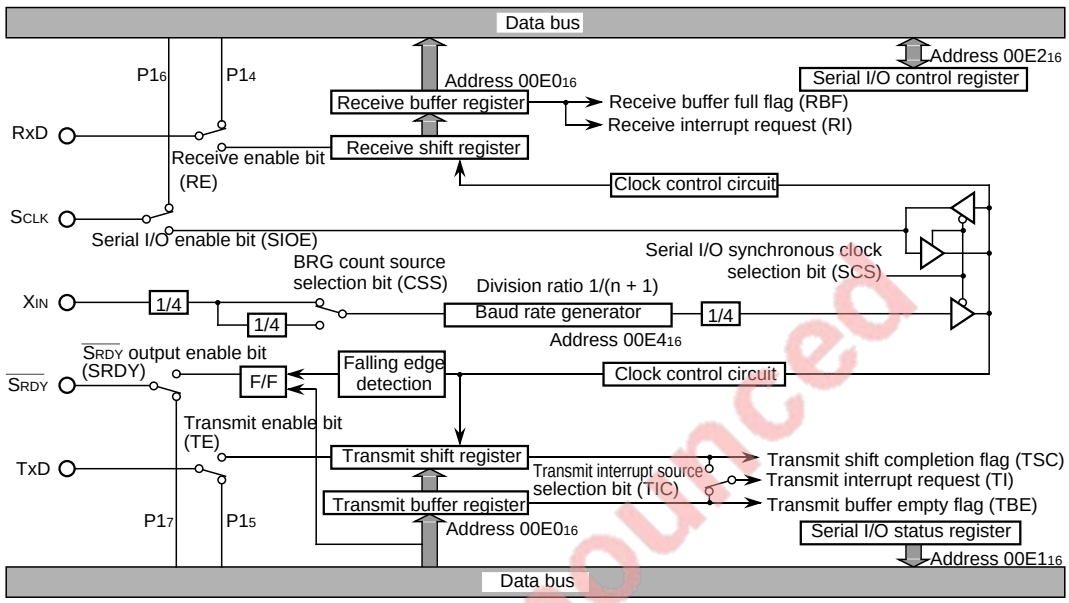


Figure 1.14.8 Block Diagram of Clock Synchronous Serial I/O

(2) Operations of Clock Synchronous Serial I/O Transmission

Transmit Operation

① When transmit data is written to the transmit buffer register (**Note 1**), the transmit buffer empty flag of the serial I/O status register is cleared to '0'.

② The transmit data written to the transmit buffer register is transferred to the transmit shift register. When the data transfer to the transmit shift register is completed, the transmit buffer empty flag goes to '1' (**Note 2**).
In this instance, when the BRG output/4 is selected as the synchronous clock, 8 synchronous clock pulses are generated.

③ Synchronized with a falling edge of the synchronous clock, the least significant bit (LSB) of the transmit data transferred to the transmit shift register is output from the TxD pin. At this time, the contents of the transmit shift register are shifted to the low-order direction by one bit, and the transmit shift completion flag is cleared to '0'.

④ By repeating the shift operation of 'Transmit Operation ③' 8 times, 8-bit transmit data is output from the TxD pin by the bit from the LSB.

⑤ When 8 bits of the transmit data are output by the 8 shift operations, the transmit shift completion flag is set to '1' (**Note 3**).

Notes 1: When the external clock is selected as the synchronous clock, write the transmit data to the transmit buffer register during the HIGH state of the synchronous clock.

2: When the transmit buffer empty flag is '1', the next transmit data can be written to the transmit buffer register.

3: The supply of the synchronous clock pulse to the transmit shift register stops automatically upon transmit completion when the BRG output/4 is selected as the synchronous clock. However, when the next transmit data is written to the transmit buffer register during the '0' state of the transmit shift completion flag, the supply of the synchronous clock pulse continues, and data is successively transmitted.
When the external clock is selected as the synchronous clock, shift operation continues as long as the external clock is being input. Therefore, it is necessary to stop the external clock after transmission is completed.

Serial I/O Transmit Interrupt

In the following cases, the serial I/O transmit interrupt request bit of interrupt request register 1 is set to '1': then the interrupt request is generated.

- When the transmit interrupt source selection bit is '0', and the data written to the transmit buffer register is transferred to the transmit shift register ('Transmit Operation ②').
- When the transmit interrupt source selection bit is '1', and the shift operation of the transmit shift register is completed ('Transmit Operation ⑤').

Figure 1.14.9 shows the transmit operation of clock synchronous serial I/O. The numbers in the figure corresponds to those of the above-mentioned 'Transmit Operation'.
Figure 1.14.10 shows a transmit timing of clock synchronous serial I/O.

1.14 Serial I/O

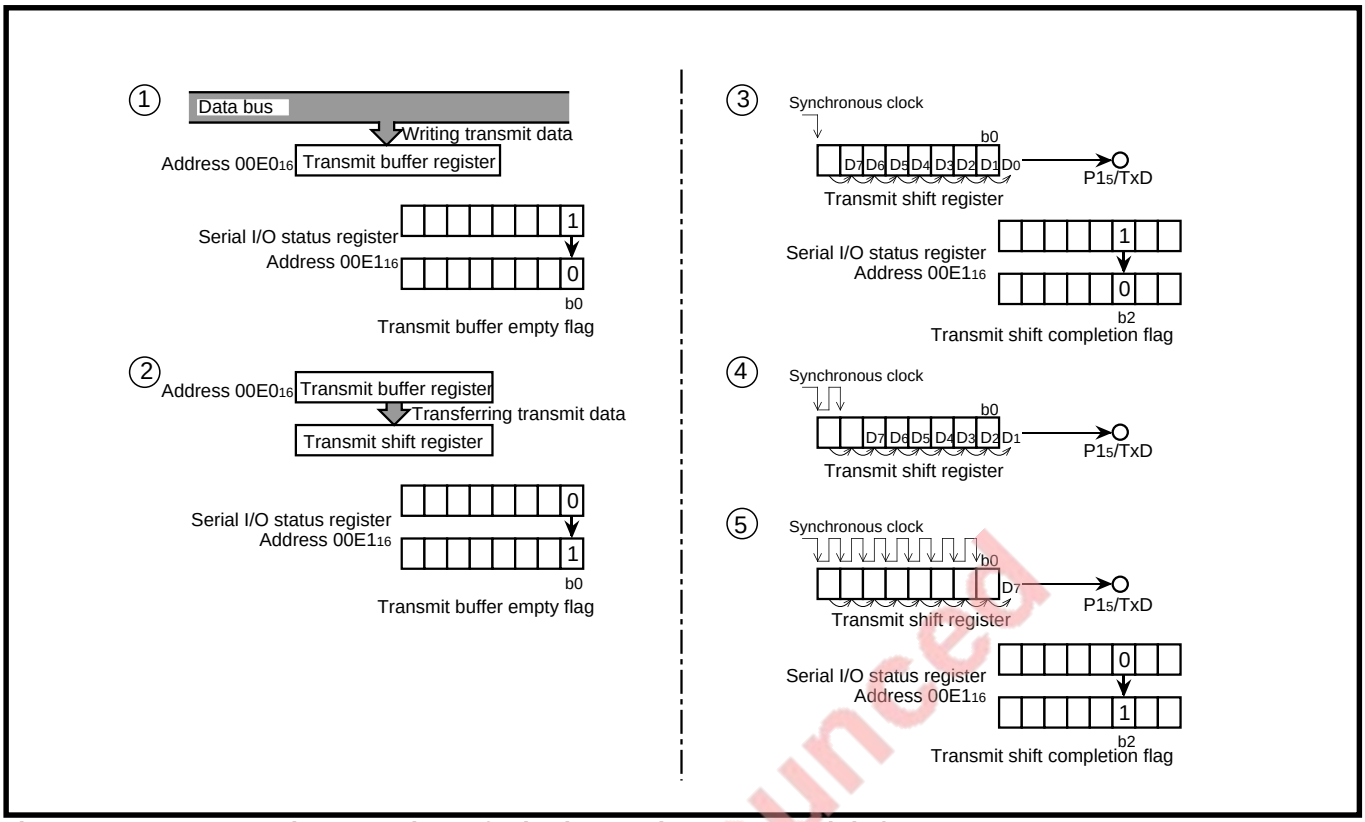
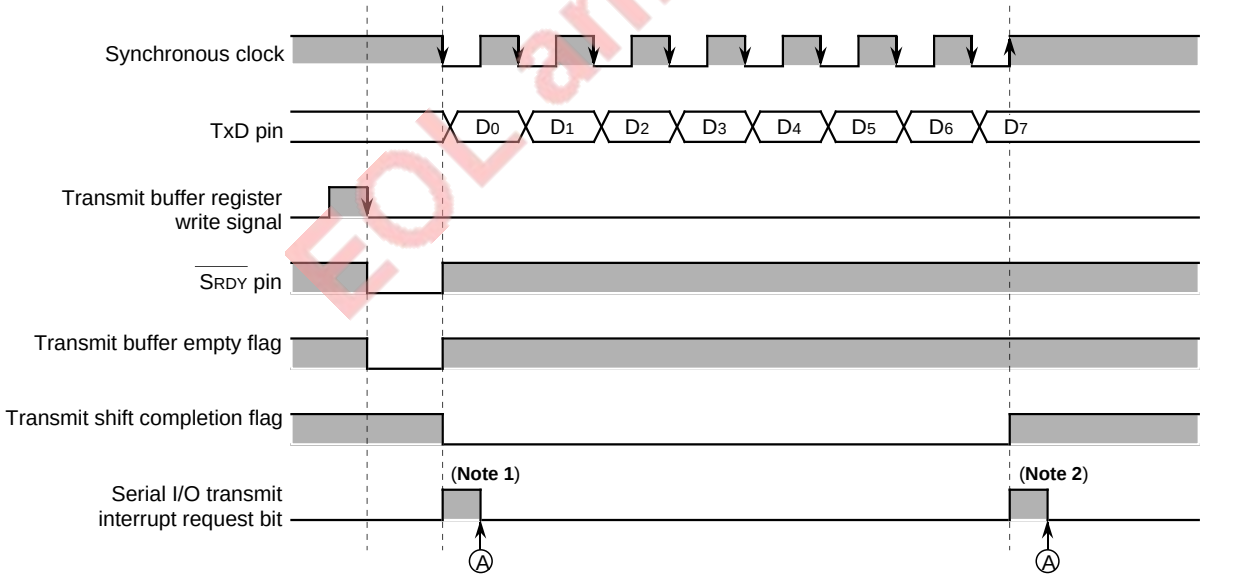


Figure 1.14.9 Transmit Operation of Clock Synchronous Serial I/O



Ⓐ : Clearing by writing '0' to the serial I/O transmit interrupt request bit or accepting a serial I/O transmit interrupt request.

Notes 1: When the transmit interrupt source selection bit is '0' (transmit buffer is emptied).
2: When the transmit interrupt source selection bit is '1' (transmit shift operation is completed).

Figure 1.14.10 Transmit Timing of Clock Synchronous Serial I/O

(3) Operations of Clock Synchronous Serial I/O Reception

Receive Operation

① Synchronized with a rising edge of the synchronous clock, a transmitted bit data is received on the RxD pin, which is stored in the most significant bit (MSB) of the receive shift register.

② The contents of the receive shift register are shifted to the low-order direction by one bit every time a bit data is received, and the next bit data is stored in the MSB. 8-bit data is fully stored in the receive shift register by repeating this shift operation 8 times.

③ The completely received 8-bit data stored in the receive shift register is transferred to the receive buffer register. When the data transfer to the receive buffer register is completed, the receive buffer full flag of the serial I/O status register is set to '1' (**Note**).

Note: If the next data is stored completely word in the receive shift register before the data transferred from the receive shift register to the receive buffer register is read through, the overrun error is generated. At this time, the overrun error flag and the summing error flag of the serial I/O status register are set to '1'. For the handling in this case, refer to **'■Handling when overrun error is generated'** in **(5) Notes on Usage of Clock Synchronous Serial I/O**.
When the receive buffer register is read, the receive buffer full flag is cleared to '0'.

Serial I/O Receive Interrupt

When the data stored completely in the receive shift register is transferred to the receive buffer register ('Receive Operation ③'), the serial I/O receive interrupt request bit of interrupt request register 1 is set to '1'; then the interrupt request is generated.

Figure 1.14.11 shows the receive operation of clock synchronous serial I/O. The numbers in the figure corresponds to those of the above-mentioned 'Receive Operation'.
Figure 1.14.12 shows a receive timing of clock synchronous serial I/O.

1.14 Serial I/O

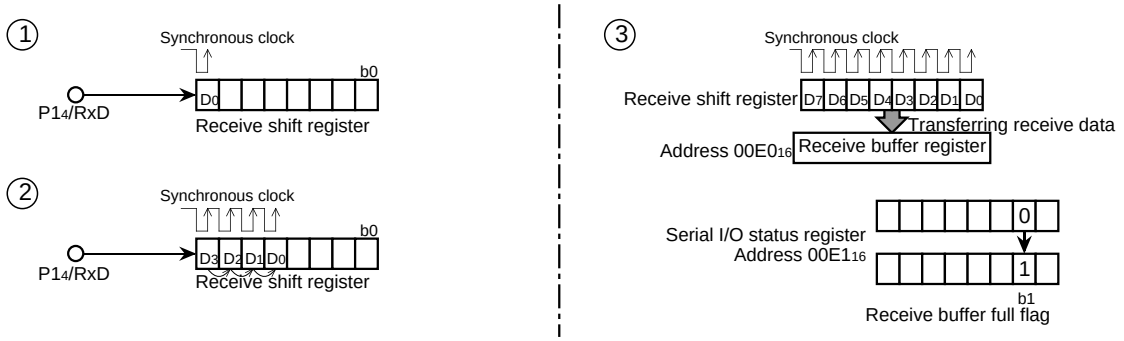


Figure 1.14.11 Receive Operation of Clock Synchronous Serial I/O

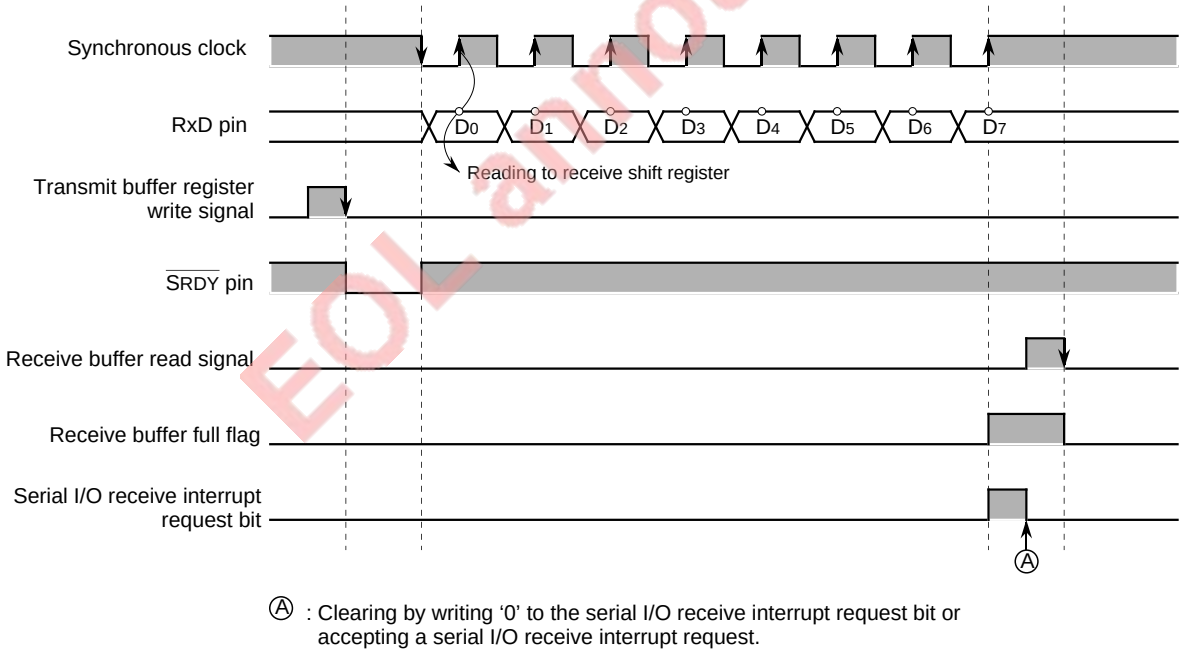
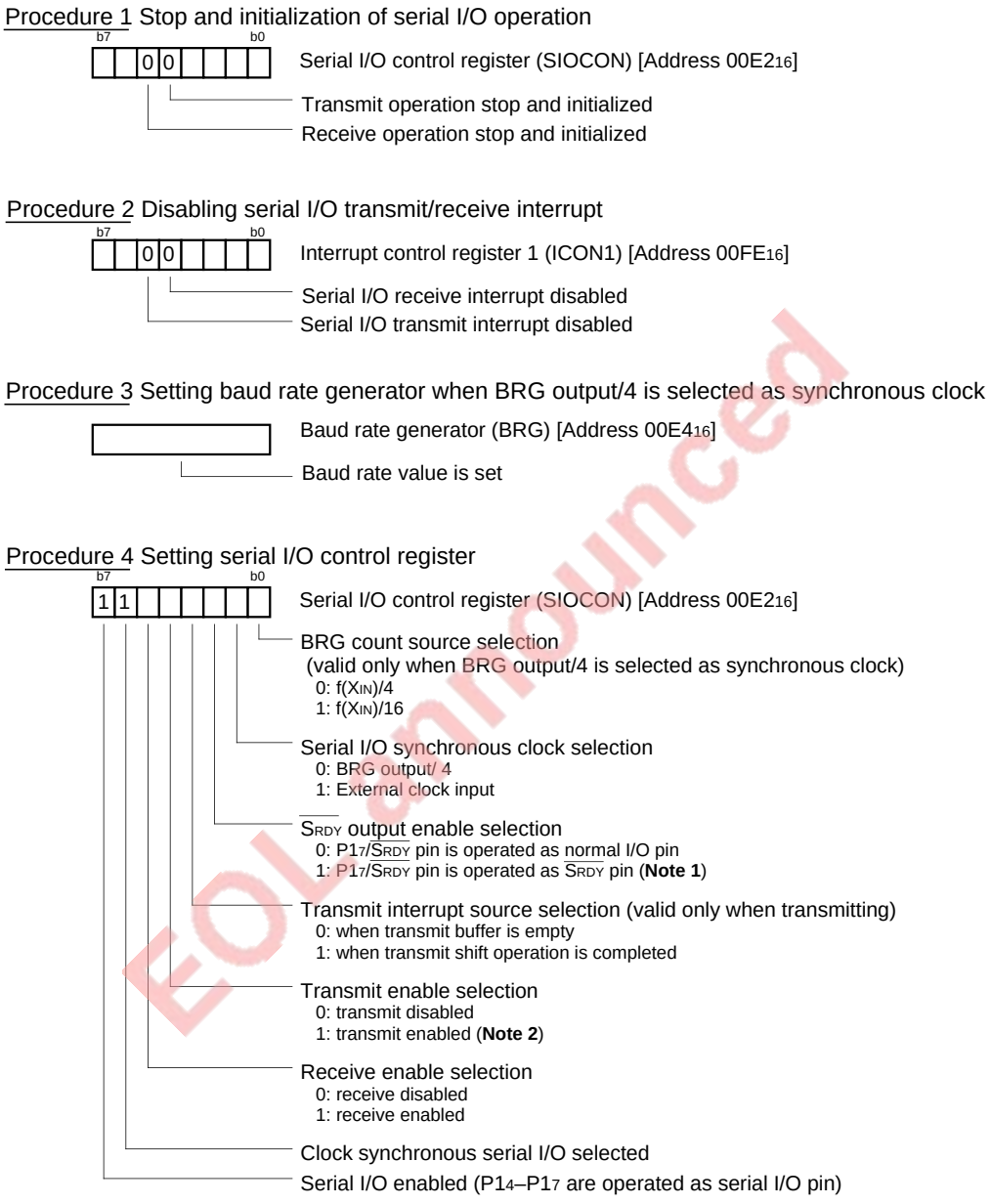


Figure 1.14.12 Receive Timing of Clock Synchronous Serial I/O

(4) Setting of Clock Synchronous Serial I/O

Figures 1.14.13 and 1.14.14 show the setting of clock synchronous serial I/O.



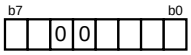
Notes 1: When the following conditions are satisfied, set the transmit enable bit in addition to the receive enable bit and SRDY output enable bit to '1'.
- Half-duplex communication is performed
- External clock is selected as synchronous clock for receive side
- SRDY output is performed.

2: Keep SCLK pin input HIGH when writing transmit enable bit to '1' to select the external clock input as the synchronous clock.

Figure 1.14.13 Setting of Clock Synchronous Serial I/O (1)

Procedure 5 Setting interrupt when serial I/O transmit/receive interrupt is used (**Note 3**)

1. '0' is set to serial I/O transmit/receive interrupt request bit

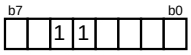


Interrupt request register 1 (IREQ1) [Address 00FC16]

Serial I/O receive interrupt request bit (when receiving)

Serial I/O transmit interrupt request bit (when transmitting)

2. Serial I/O transmit/receive interrupt is enabled



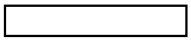
Interrupt control register 1 (ICON1) [Address 00FE16]

Serial I/O receive interrupt enabled (when receiving)

Serial I/O transmit interrupt enabled (when transmitting)

Notes 3: Refer to■ **Handling when overrun error is generated** in **(5) Notes on Usage of Clock Synchronous Serial I/O.**

Procedure 6 Start of data transmit/receive



Transmit buffer register (TB) [Address 00E016]

Writing transmit data when transmitting (**Note 4**)

Writing dummy data when receiving half-duplex communication

Notes 4: Keep SCLK pin input HIGH when writing transmit data to transmit buffer register to select the external clock input as the synchronous clock.

Figure 1.14.14 Setting of Clock Synchronous Serial I/O (2)

(5) Notes on Usage of Clock Synchronous Serial I/O

Pay attention to the following notes when clock synchronous serial I/O is selected.

Selecting External Clock as Synchronous Clock

■ Perform the following operations while the SCLK pin input is HIGH during transmission:
- Write '1' to the transmit enable bit
- Write transmit data to the transmit buffer register

■ The shift operations of the transmit shift register and the receive shift register are performed while the synchronous clock is being input to the serial I/O circuit. Stop the synchronous clock with 8 cycles.

■ Keep the HIGH- and the LOW- width (TWH and TWL) of the pulses used as the external clock source TWH, TWL [s] ≥ (8/f(XIN) [Hz]). For example, use a frequency of 500 kHz or less (50% duty cycle) as the external clock source when f(XIN) = 8 MHz.

■ Set the transmit enable bit to '1' as well as the receive enable bit and the $\overline{\text{SRDY}}$ output enable bit of the serial I/O control register when the receiver outputs the $\overline{\text{SRDY}}$ signal.

Handling Recovering from Errors Generated

■ Handling when overrun error is generated

If the next data is stored completely word in the receive shift register before the data transferred from the receive shift register to the receive buffer register is read through, the overrun error is generated. At this time, the overrun error flag and the summing error flag of the serial I/O status register are set to '1'. The contents of the receive shift register are not transferred to the receive buffer register, so that the contents of the receive buffer register remain unaffected. As a result, if the contents of the receive buffer register are read, the data of the receive shift register is not transferred to the receive buffer register and becomes invalid.

When the overrun error occurs, clear the overrun error flag to '0' by any of the following operations, and perform receive preparation again.
- Clear the serial I/O enable bit of the serial I/O control register to '0'. (In this case, only the overrun error flag returns to '0'.)
- Clear the receive enable bit of the serial I/O control register to '0'.
- Write dummy data into the serial I/O status register.

Referring to Transmit Shift Completion Flag

The transmit shift completion flag changes from '1' to '0' with a delay of 0.5 to 1.5 clocks of the synchronous clock. Therefore, pay attention to this delay when data transmission is controlled, by referring to the transmit shift completion flag after the transmit data is written to the transmit buffer register.

Stopping Transmission/Reception of Clock Synchronous Serial I/O

■ In order to stop the transmit operation in half-duplex transmission, clear the transmit enable bit of the serial I/O control register to '0'. As a result, the following stop and initialization of transmit operation are performed:
- To stop and initialize the clock supplied to the transmit shift register
- To clear the transmit shift register (Only when '0' is written to the transmit enable bit while the SCLK pin input is HIGH, selecting an external clock as the synchronous clock.)
- To clear the transmit buffer empty flag and transmit shift completion flag

REASON: Neither stopping transmit operation nor initializing the transmitter circuit is performed even when the serial I/O enable bit is cleared to '0' (serial I/O disabled), and internal transmit operation continues. (Because serial I/O pins TxD, RxD, SCLK, and $\overline{\text{SRDY}}$ function as I/O port pins, transmit data cannot be output to the external.)

■ In order to stop the receive operation in half-duplex transmission, clear the receive enable bit or the serial I/O enable bit of the serial I/O control register to '0'. As a result, the following stop and initialization of the receive operation are performed:
- To stop and initialize the clock supplied to the receive shift register
- To clear the receive shift register
- To clear every error flag
- To clear the receive buffer full flag

■ In order to stop the transmit and receive operations in full-duplex transmission, clear both the transmit enable bit and the receive enable bit of the serial I/O control register to '0' at the same time. (To stop only one of the transmit or receive operation cannot be done in the full-duplex communication of clock synchronous serial I/O.)

REASON: In clock synchronous serial I/O, the same clock is used for transmission and reception. Therefore, transmission and reception cannot be synchronized when either transmit or receive operation is disabled, causing displacement of bit positions.

Re-setting Serial I/O Control Register

Re-set the serial I/O control register according to the following sequence:

① Clear both the transmit and receive enable bits of the serial I/O control register to '0' to stop and initialize transmit and receive operations.

② Set bits 0 to 3 and 6 of the serial I/O control register.

③ Set the transmit enable bit or receive enable bit to '1'.

(Procedures ② and ③ can be performed simultaneously with the **LDM** instruction.)

Using Serial I/O Transmit Interrupt and Serial I/O Receive Interrupt

■ Set the associated registers in the following sequence to use serial I/O transmit interrupt.

① Clear the serial I/O transmit interrupt enable bit of interrupt control register 1 to '0'.
② Set the serial I/O control register.
③ Execute one or more instructions such as **NOP**.
④ Clear the serial I/O transmit interrupt request bit of interrupt request register 1 to '0'.
⑤ Set the serial I/O transmit interrupt enable bit of interrupt control register 1 to '1'.

REASONS 1: If normal port pins are switched to serial I/O pins with the serial I/O control register, the serial I/O transmit interrupt request bit may become '1'.

2: If the transmit enable bit of the serial I/O control register is set to '1', the transmit buffer empty flag and the transmit shift completion flag are '1'. As a result, the serial I/O transmit interrupt request bit becomes '1' regardless of the state of the transmit interrupt source selection bit of the serial I/O control register, and the interrupt request is generated.

■ Set the associated registers in the following sequence to use serial I/O receive interrupt.

① Clear the serial I/O receive interrupt enable bit of interrupt control register 1 to '0'.
② Set the serial I/O control register.
③ Execute one or more instructions, such as **NOP**.
④ Clear the serial I/O receive interrupt request bit of interrupt request register 1 to '0'.
⑤ Set the serial I/O receive interrupt enable bit of interrupt control register 1 to '1'.

REASON: If normal port pins are switched to serial I/O pins with the serial I/O control register, the serial I/O receive interrupt request bit may become '1'.

1.14.3 Clock Asynchronous Serial I/O (UART)

In clock asynchronous serial I/O (UART), the transmit operation of the transmitter and the receive operation of the receiver are performed simultaneously, synchronizing with the synchronous clock used for transferring, which is generated by the clock control circuit.
In UART, the transmitter and the receiver have the same transmit/receive baud rate and the same data transfer format.

UART is selected by clearing the serial I/O mode selection bit of the serial I/O control register to '0'.

Data Communication

- Half-duplex communication: one of the two communicating microcomputers operates only as a transmitter and the other only as a receiver at a time or vice versa.
- Full-duplex communication: both of the two communicating microcomputers operate simultaneously as transmitter and receiver.

Synchronous Clock

A synchronous clock is selected by the serial I/O synchronous clock selection bit of the serial I/O control register as follows:

- 0: BRG output /16
- 1: External clock/16 input to the SCLK pin

For the BRG output, refer to **(5) Baud Rate Generator** in **Section 1.14.1**.

Notes 1: In UART, the P16/SCLK pin can be used as port pin P16 when the BRG output/16 is selected as the synchronous clock, since the SCLK pin is not used to output the synchronous clock to the external.

2: When the external clock/16 is selected as a synchronous clock, keep the HIGH- and the LOW-width (TWH and TWL) of the pulses used as the external clock source TWH, TWL [s] ≥ (2/f(XIN) [Hz]). For example, use a frequency of 2 MHz or less (50% duty cycle) as the external clock source when f(XIN) = 8 MHz.

Data Transfer Rate (Baud Rate)

In UART, the baud rate, which is the frequency of the synchronous clock, is calculated by the following formulas.

- When the serial I/O synchronous clock selection bit is '0':.
  (BRG output/16 is selected as the synchronous clock.)

$$\text{Baud rate [bps]} = \frac{f(X_{IN})}{\text{Division ratio (\textbf{Note 1})} \times (\text{BRG setting value (\textbf{Note 2})} + 1) \times 16}$$

Notes 1: BRG count source selection bit of the serial I/O control register is as follows:
- '0': Division ratio is 4
- '1': Division ratio is 16.

2: The value written to the baud rate generator (00_{16} to FF_{16}).

- When the serial I/O synchronous clock selection bit is '1'
  (the external clock/16 input is selected as the synchronous clock):

$$\text{Baud rate [bps]} = \frac{\text{External clock input frequency from the } S_{CLK} \text{ pin}}{16}$$

Table 1.14.2 lists an example of baud rates.

Table 1.14.2 Example of Baud Rates

Baud Rate	$f(X_{IN})$ = 7.9872 MHz		$f(X_{IN})$ = 3.9936 MHz	
[bps]	Count Source	BRG Setting Value	Count Source	BRG Setting Value
300	$f(X_{IN})/16$	103 (67_{16})	$f(X_{IN})/16$	51 (33_{16})
600	$f(X_{IN})/16$	51 (33_{16})	$f(X_{IN})/16$	25 (19_{16})
1200	$f(X_{IN})/16$	25 (19_{16})	$f(X_{IN})/16$	12 ($0C_{16}$)
2400	$f(X_{IN})/16$	12 ($0C_{16}$)	$f(X_{IN})/4$	25 (19_{16})
4800	$f(X_{IN})/4$	25 (19_{16})	$f(X_{IN})/4$	12 ($0C_{16}$)
9600	$f(X_{IN})/4$	12 ($0C_{16}$)		
15600	$f(X_{IN})/4$	7 (07_{16})		
31200	$f(X_{IN})/4$	3 (03_{16})		
41600	$f(X_{IN})/4$	2 (02_{16})		

1.14 Serial I/O

Data Transfer Formats

In UART, the data transfer formats shown in Figure 1.14.15 can be selected with the UART control register.

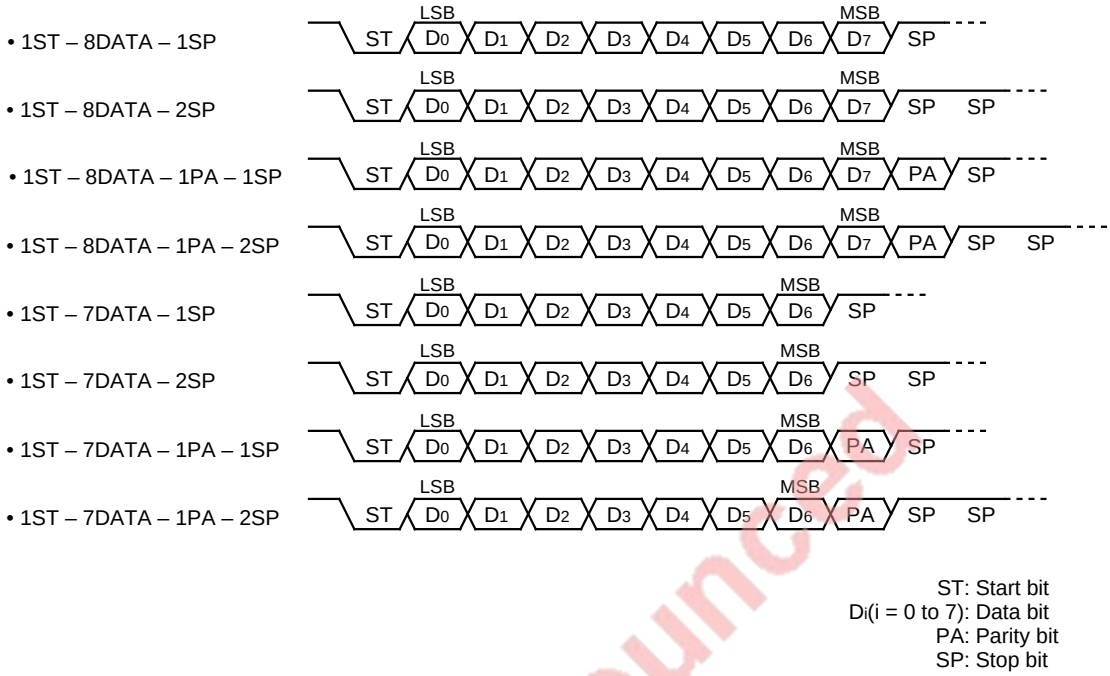


Figure 1.14.15 Data Transfer Formats in UART

Table 1.14.3 lists the setting of the UART control register, and Table 1.14.4 lists the function of the UART data transfer bits.

Table 1.14.3 Setting of UART Control Register

Transfer Data Format	UART Control Register			
	b3 **(Note 1)**	b2 **(Note 2)**	b1 **(Note 3)**	b0 **(Note 4)**
1ST—8DATA—1SP	0	—	0	0
1ST—7DATA—1SP	0		0	1
1ST—8DATA—1PA—1SP	0	0: Even parity	1	0
1ST—7DATA—1PA—1SP	0	1: Odd parity	1	1
1ST—8DATA—2SP	1	—	0	0
1ST—7DATA—2SP	1		0	1
1ST—8DATA—1PA—2SP	1	0: Even parity	1	0
1ST—7DATA—1PA—2SP	1	1: Odd parity	1	1

Notes 1: Stop bit length selection bit
2: Parity selection bit
3: Parity enable bit
4: Character length selection bit

Table 1.14.4 Function of UART Transfer Data Bits

Name	Function
Start Bit (ST)	Indicates the start of data transmission. The LOW signal of one-bit wide is added to the head of the transmit data.
Data Bits (DATA)	The transmit data written into UART transmit buffer register. Data '0' represents LOW, and '1' the HIGH signal.
Parity Bit (PA)	Added to the end of the data bits to enhance the reliability of communications. The number of '1' in transmit/receive data including parity bit keeps even or odd according to the setting value of parity selection bit.
Stop Bit(s) (SP)	Added to the end of the data bits (or after the parity bit if parity is valid) and indicates the transmission is completed. The HIGH signal of one or two bits wide is output as stop bit.

(1) Block Diagram of UART

Figure 1.14.16 shows the block diagram of UART.

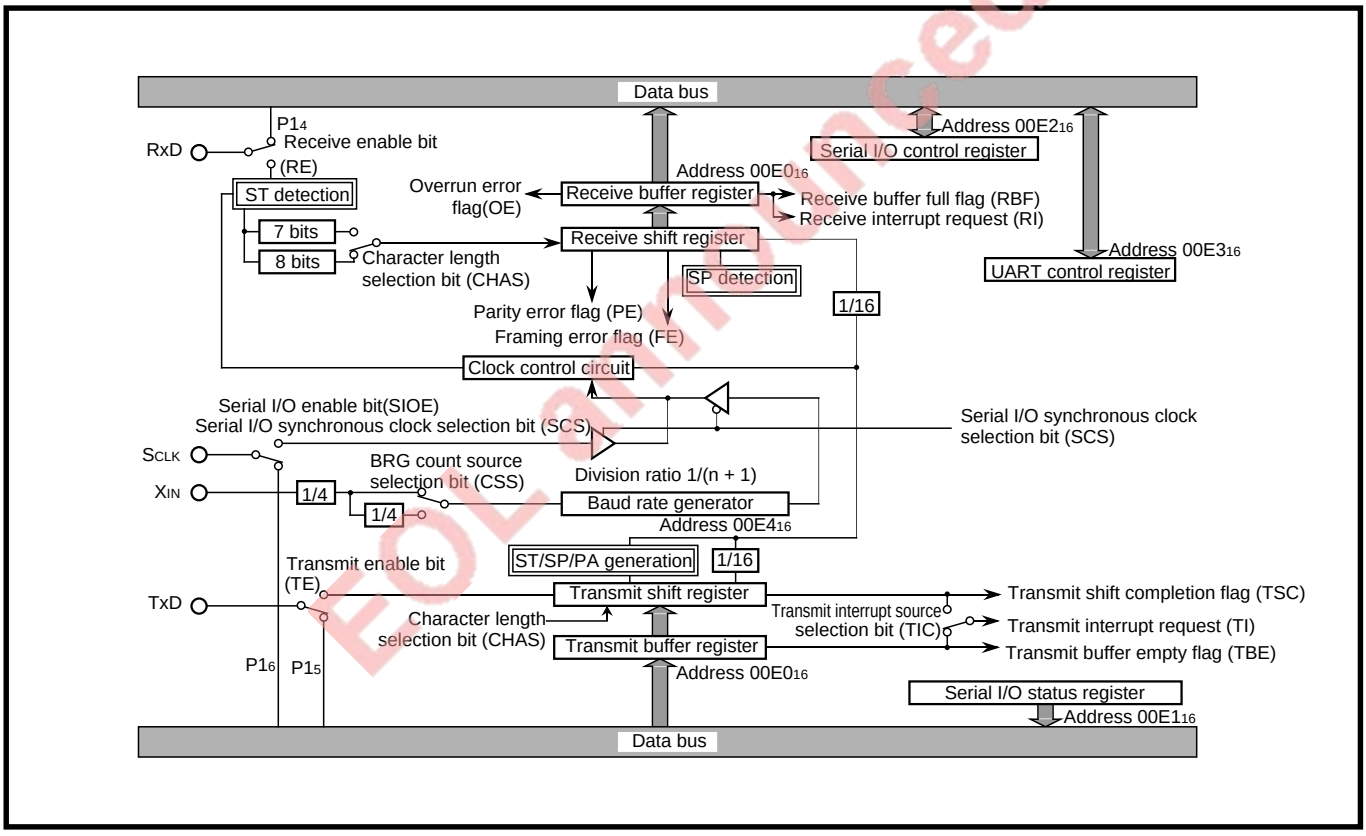


Figure 1.14.16 Block Diagram of UART

(2) Operations of UART Transmission

Transmit Operation

① When transmit data is written to the transmit buffer register, the transmit buffer empty flag of the serial I/O status register is cleared to '0'.

② The transmit data written to the transmit buffer register is transferred to the transmit shift register. When the data transfer to the transmit shift register is completed, the transmit buffer empty flag goes to '1' (**Note 1**).

③ Synchronized with a falling edge of the synchronous clock, the start bit (the LOW level) is output from the TxD pin.

④ Synchronized with the next falling edge of the synchronous clock, the least significant bit (LSB) of the transmit data transferred to the transmit shift register is output from the TxD pin. At this time, the contents of the transmit shift register are shifted to the low-order direction by one bit, and the serial I/O transmit shift completion flag is cleared to '0'.

⑤ By repeating the shift operation of 'Transmit Operation ④' 'n' times ('n': the number of bits set by the character length selection bit of the UART control register), the transmit data is output from the TxD pin by the bit from the LSB.

⑥ After the transmit data is output, the parity bit, and then the stop bit (the HIGH level), are output from the TxD pin synchronized with falling edges of the synchronous clock. The parity bit and the stop bit are generated and output automatically, according to the setting of the parity enable bit, the parity selection bit, and the stop bit length selection bit of the UART control register.
When the last stop bit of the transfer format is output, the transmit shift completion flag is set to '1' at the next rising edge of the synchronous clock (**Note 2**).

Notes 1: When the transmit buffer empty flag is '1', the next transmit data can be written to the transmit buffer register.

2: The supply of the synchronous clock pulse to the transmit shift register stops automatically upon transmit completion when the BRG output/16 is selected as the synchronous clock. However, when the next transmit data is written to the transmit buffer register during the '0' state of the transmit shift completion flag, the supply of the synchronous clock pulse continues, and data is successively transmitted.

Serial I/O Transmit Interrupt

In the following cases, the serial I/O transmit interrupt request bit of interrupt request register 1 is set to '1'; then the interrupt request is generated:

- When the transmit interrupt source selection bit is '0', and the data written to the transmit buffer register is transferred to the transmit shift register ('Transmit Operation ②').
- When the transmit interrupt source selection bit is '1', and the shift operation of the transmit shift register is completed ('Transmit Operation ⑥').

Figure 1.14.17 shows the transmit operation of UART, and Figure 1.14.18 shows a transmit timing example in UART.

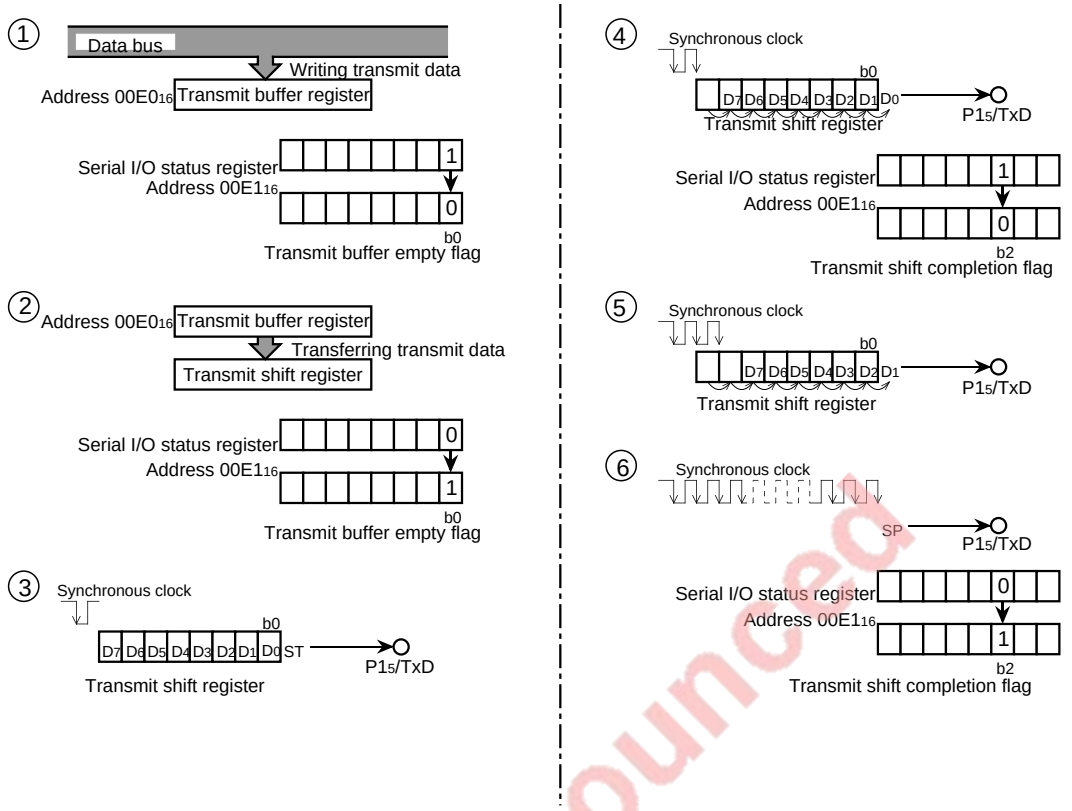
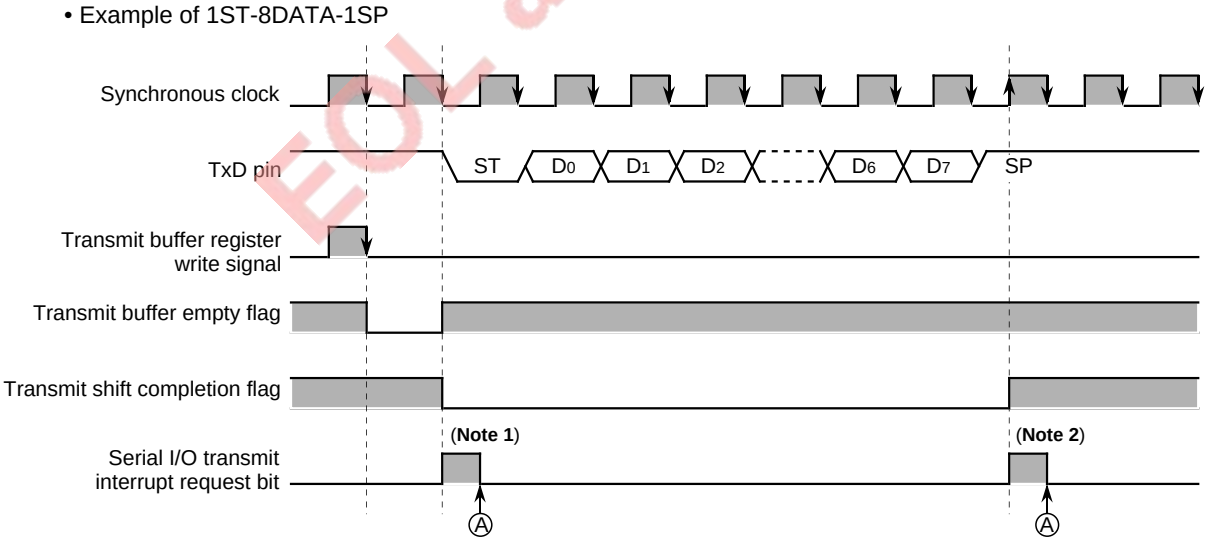


Figure 1.14.17 Transmit Operation of UART



Ⓐ: Clearing by writing '0' to the serial I/O transmit interrupt request bit or accepting a serial I/O transmit interrupt request.

Notes 1: When the transmit interrupt source selection bit is '0' (transmit buffer is emptied).
2: When the transmit interrupt source selection bit is '1' (transmit shift operation is completed).

Figure 1.14.18 Transmit Timing example in UART

(3) Operations of UART Reception

Receive Operation

① When a falling edge of the RxD pin input is detected, this input level to the RxD pin is identified according to the subsequent rising edge of the synchronous clock as follows:
- As the start bit when the level is LOW.
- As noise when the level is HIGH. In this case, the CPU suspends the receive operation and enters the waiting state for the next start bit.

② Synchronized with the rising edge of the synchronous clock, transmitted data is received on the RxD pin by the bit and stored in the most significant bit (MSB) of the receive shift register. Every time a data bit is received, the contents of the receive shift register are shifted by one bit to the low-order direction.

③ The receive shift operation of 'Receive Operation ②' is performed 'n' times ('n': the number of bits set by the character length selection bit of the UART control register), and the received data is stored completely in the receive shift register (**Note 1**).

④ The received data stored completely in the receive shift register is transferred to the receive buffer register.

⑤ The parity bit and the stop bit are input to the RxD pin synchronized with rising edges of the synchronous clock. When the last stop bit (the HIGH level) is input to the RxD pin, the receive buffer full flag of the serial I/O status register is set to '1' at the subsequent falling edge of the synchronous clock (**Note 2**).
At this time, error flags are checked.

Notes 1: When the character length selection bit is '1' (7 bits wide), the MSB of the receive buffer register becomes '0'.

2: If the next data is stored completely in the receive shift register before the data transferred from the receive shift register to the receive buffer register is read through (the receive buffer full flag is '1'), the overrun error is generated. At this time, the overrun error flag and the summing error flag of the serial I/O status register is set to '1'. Refer to **(5) Notes on Usage of UART**.
When the receive buffer register is read, the receive buffer full flag is cleared to '0'.

Serial I/O Receive Interrupt

When the receive buffer full flag goes to '1' ('Receive Operation ⑤'), the serial I/O receive interrupt request bit of interrupt request register 1 is set to '1'; then the interrupt request is generated.

Figure 1.14.19 shows the receive operation of UART and Figure 1.14.20 shows a receive timing example in UART.

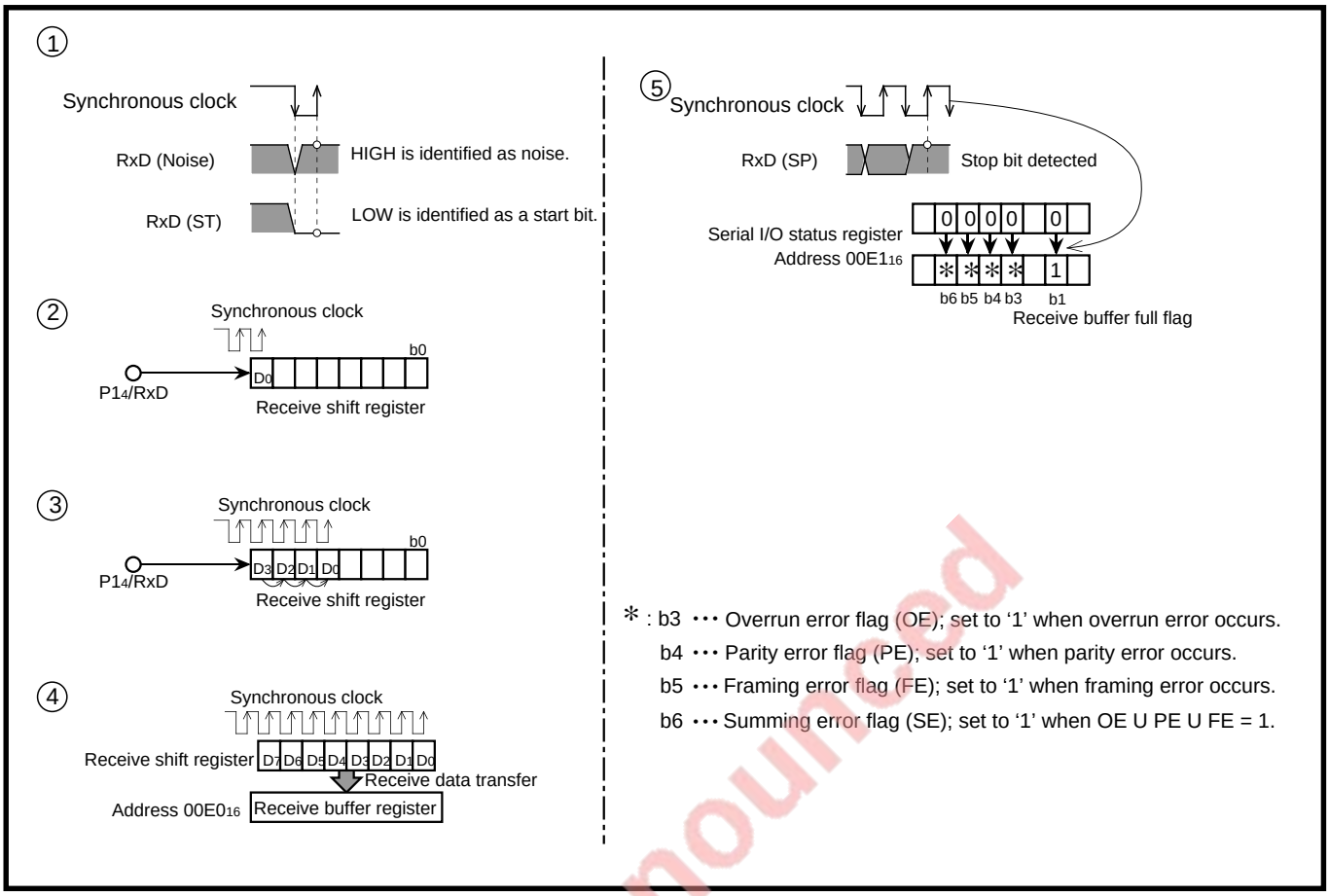


Figure 1.14.19 Receive Operation of UART

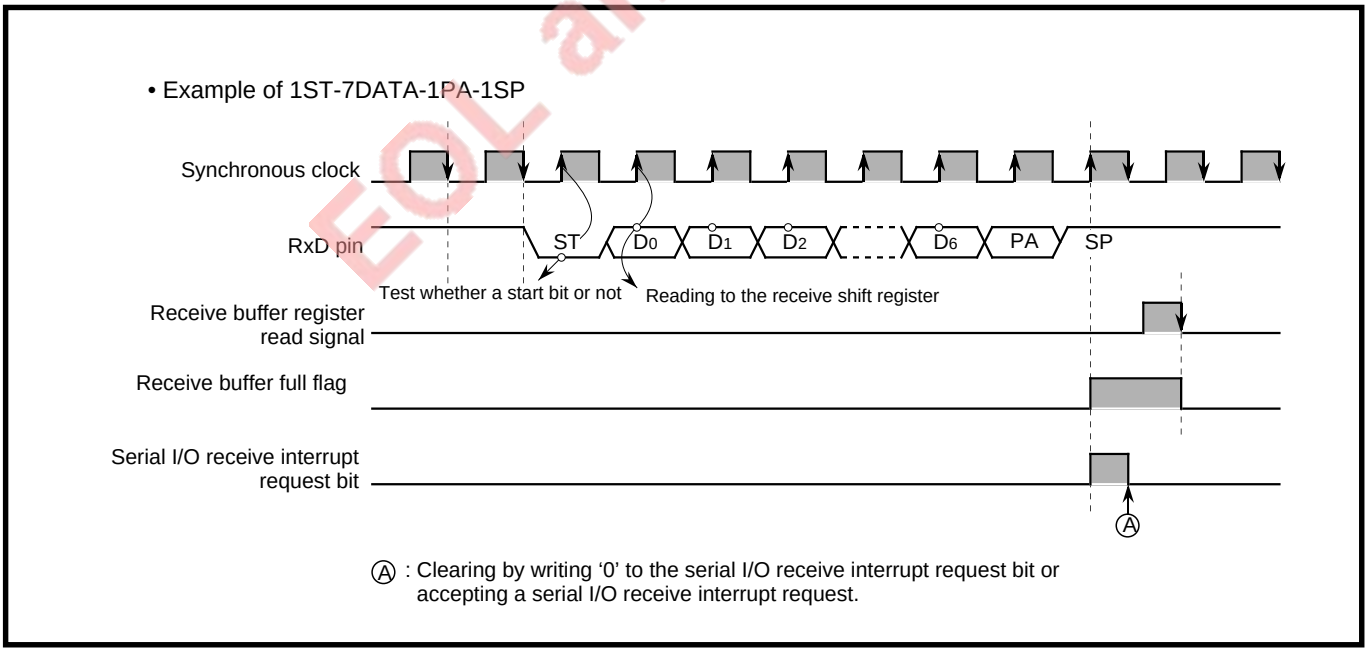
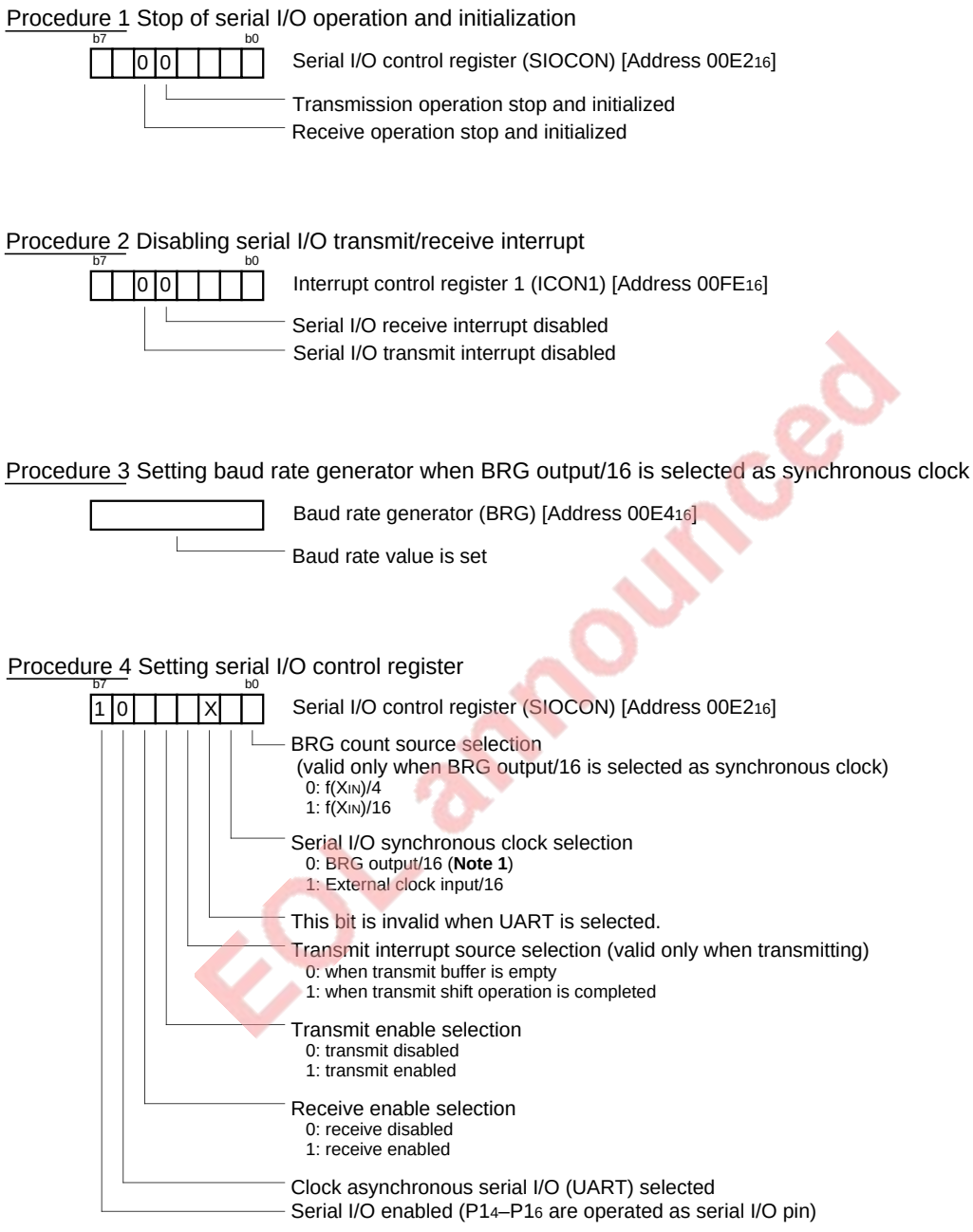


Figure 1.14.20 Receive Timing Example in UART

(4) Setting of UART

Figures 1.14.21 and 1.14.22 show the setting of UART.



Notes 1: When BRG output/16 is selected as the synchronous clock, P16/SCLK pin can be used as port pin P16 because the synchronous clock is not output externally from SCLK pin.

Figure 1.14.21 Setting of UART (1)

Procedure 5 Setting UART control register

UART control register (UARTCON) [Address $00E3_{16}$]

Character length selection
0: 8 bits
1: 7 bits

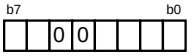
Parity enable selection
0: Parity disabled
1: Parity enabled

Parity selection (valid only when parity enabled)
0: Even parity
1: Odd parity

Stop bit length selection
0: 1 stop bit
1: 2 stop bits

Procedure 6 Setting interrupt when serial I/O transmit/receive interrupt is used (**Note 2**)

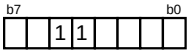
1. '0' is set to serial I/O transmit/receive interrupt request bit



Interrupt request register 1 (IREQ1) [Address $00FC_{16}$]

Serial I/O receive interrupt request bit (when receiving)
Serial I/O transmit interrupt request bit (when transmitting)

2. Serial I/O transmit/receive interrupt is enabled

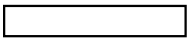


Interrupt control register 1 (ICON1) [Address $00FE_{16}$]

Serial I/O receive interrupt enabled (when receiving)
Serial I/O transmit interrupt enabled (when transmitting)

Notes 2: Refer to **Using Serial I/O Transmit Interrupt and Serial I/O Receive Interrupt** in **(5) Notes on Usage of UART.**

Procedure 7 Start of data transmit when transmitting



Transmit buffer register (TB) [Address $00E0_{16}$]

Writing transmit data

Figure 1.14.22 Setting of UART (2)

(5) Notes on Usage of UART

Pay attention to the following notes when UART is selected.

Selecting BRG output/16 as Synchronous Clock

Since the SCLK pin is not used to output the synchronous clock to the external, the P16/SCLK pin can be used as normal port pin P16.

Selecting External Clock/16 Input as Synchronous Clock

Keep the HIGH- and the LOW- width (T_{WH} and T_{WL}) of the pulses used as the external clock source T_{WH}, T_{WL} [s] ≥ (2/f(X_{IN}) [Hz]). For example, use a frequency of 2 MHz or less (50% duty cycle) as the external clock source when f(X_{IN}) = 8 MHz.

Handling Recovering from Errors Generated

■ Handling when parity error or framing error is generated

When the parity error or the framing error occurs, the flag corresponding to each error and the summing error flag of the serial I/O status register are set to '1'. To clear these flags to '0', perform either of the following operations.

- Clear the receive enable bit of the serial I/O control register to '0'.
- Write dummy data to the serial I/O status register.

■ Handling when overrun error is generated

If the next data is stored completely in the receive shift register before the data transferred from the receive shift register to the receive buffer register is read through, the overrun error is generated. At this time, the overrun error flag and the summing error flag of the serial I/O status register are set to '1'. The contents of the receive shift register are not transferred to the receive buffer register, so that the contents of the receive buffer register remain unaffected. As a result, if the contents of the receive buffer register are read, the data of the receive shift register is not transferred to the receive buffer register and becomes invalid.

When the overrun error occurs, clear the overrun error flag to '0' by any of the following operations and perform receive preparation again.

- Clear the serial I/O enable bit of the serial I/O control register to '0'. (In this case, only the overrun error flag returns to '0'.)
- Clear the receive enable bit of the serial I/O control register to '0'.
- Write dummy data into the serial I/O status register.

Referring to Transmit Shift Completion Flag

The transmit shift completion flag changes from '1' to '0' with a delay of 0.5 to 1.5 clocks of the synchronous clock. Therefore, pay attention to this delay when data transmission is controlled, by referring to the transmit shift completion flag after the transmit data is written to the transmit buffer register.

Stopping Transmission/Reception of UART

■ In order to stop the transmit operation of UART, clear the transmit enable bit of the serial I/O control register to '0'. As a result, the following stop and initialization of transmit operation are performed:

- To stop and initialize the clock supplied to the transmit shift register
- To clear the transmit shift register.
- To clear the transmit buffer empty flag and transmit shift completion flag

■ In order to stop the receive operation of UART, clear the receive enable bit or the serial I/O enable bit of the serial I/O control register to '0'. As a result, the following stop and initialization of the receive operation are performed:

- To stop and initialize the clock supplied to the receive shift register
- To clear the receive shift register
- To clear every error flag
- To clear the receive buffer full flag

Re-setting Serial I/O Control Register

Re-set the serial I/O control register according to the following sequence to stop and initialize transmit and receive operations:

① Clear both of the transmit and receive enable bits of the serial I/O control register to '0'.
② Set bits 0 to 3 and 6 of the serial I/O control register.
③ Set both the transmit and receive enable bits to '1'.
(Procedures ② and ③ can be performed simultaneously with the **LDM** instruction.)

Using Serial I/O Transmit Interrupt and Serial I/O Receive Interrupt

■ Set the associated registers in the following sequence to use serial I/O transmit interrupt.

① Clear the serial I/O transmit interrupt enable bit of interrupt control register 1 to '0'.
② Set the serial I/O control register.
③ Execute one or more instructions such as **NOP**.
④ Clear the serial I/O transmit interrupt request bit of interrupt request register 1 to '0'.
⑤ Set the serial I/O transmit interrupt enable bit of interrupt control register 1 to '1'.

REASONS 1: If normal port pins are switched to serial I/O pins with the serial I/O control register, the serial I/O transmit interrupt request bit may become '1'.

2: If the transmit enable bit of the serial I/O control register is set to '1', the transmit buffer empty flag and the transmit shift completion flag are '1'. As a result, the serial I/O transmit interrupt request bit becomes '1' regardless of the state of the transmit interrupt source selection bit of the serial I/O control register, and the interrupt request is generated.

■ Set the associated registers in the following sequence to use serial I/O receive interrupt.

① Clear the serial I/O receive interrupt enable bit of interrupt control register 1 to '0'.
② Set the serial I/O control register.
③ Execute one or more instructions, such as **NOP**.
④ Clear the serial I/O receive interrupt request bit of interrupt request register 1 to '0'.
⑤ Set the serial I/O receive interrupt enable bit of interrupt control register 1 to '1'.

REASON: If normal port pins are switched to serial I/O pins with the serial I/O control register, the serial I/O receive interrupt request bit may become '1'.

1.14 Serial I/O

1.14.4 Bus Arbitration

In the serial I/O communications of the contention bus system shown in Figure 1.14.23, transmit data may not correctly be sent on the transmission line because of bus collision.

In the 7480 Group and 7481 Group, if the comparison of the level of serial I/O transmit pin TxD with that of serial I/O receive pin RxD results in a mismatch, the bus arbitration interrupt request is generated. This indicates that the bus collision occurred.

When the bus collision detection enable bit of the bus collision detection control register is set to '1', bus collision detection can be performed. In addition, bus collision detection is valid when any of the following conditions is selected:

Serial I/O mode

- Clock synchronous serial I/O
- Clock asynchronous serial I/O (UART)

Synchronous clock

- BRG output divided
- External clock (or external clock/16)

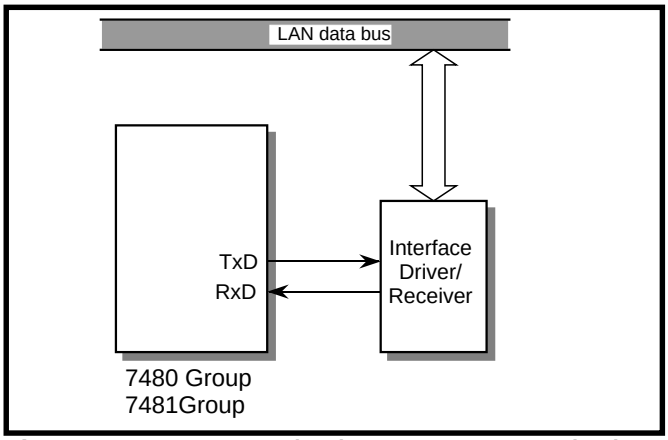


Figure 1.14.23 Contention bus system communications

(1) Block Diagram

Figure 1.14.24 shows the block diagram of the bus arbitration interrupt.

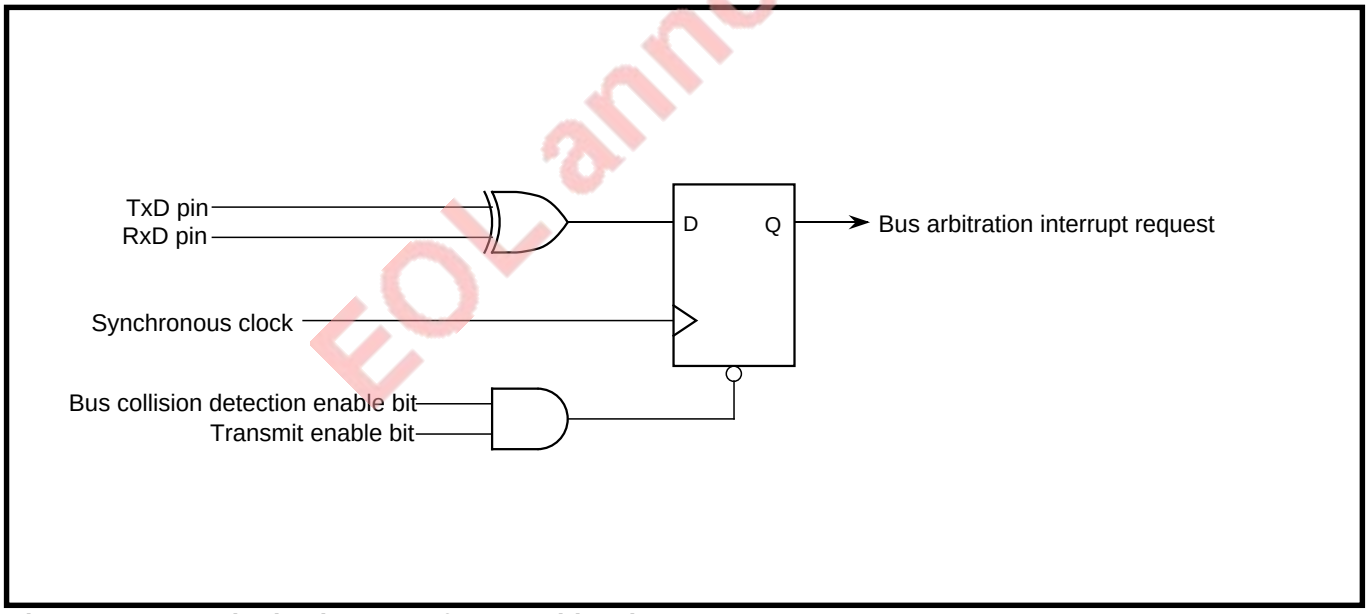


Figure 1.14.24 Block Diagram of Bus Arbitration Interrupt

(2) Operations of Bus Arbitration

Operations of bus arbitration in the serial I/O communications are described below.

Bus Collision Detection

The level of serial I/O transmit pin TxD is compared with that of serial I/O receive pin RxD, synchronized with rising edges of the synchronous clock which is used in serial I/O communications.

- The level of 8-bit transmit data is referred for comparison in clock synchronous serial I/O.
- The levels of all transmitted bits, from the start to the stop bits, are referred for comparison in UART.

Bus Arbitration Interrupt

When a mismatch results from the comparison of the level of the TxD pin with that of the RxD pin in bus collision detection, the bus arbitration interrupt request bit of interrupt request register 1 is set to '1'; then the interrupt request is generated.

Figure 1.14.25 shows a timing of bus collision detection.

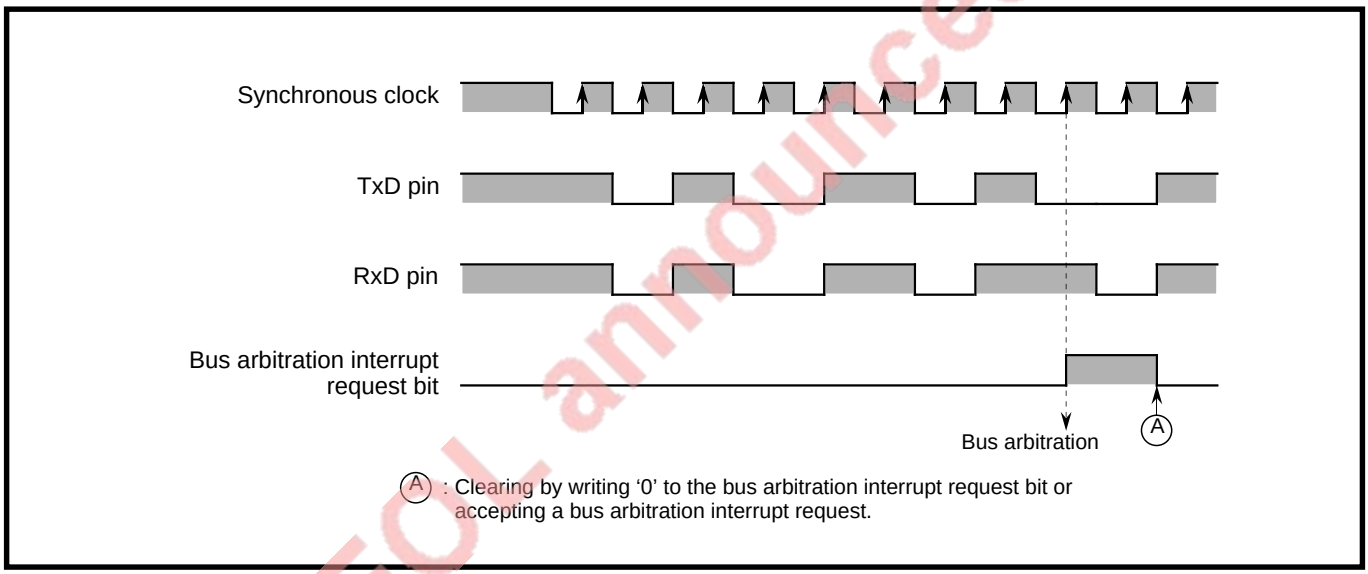


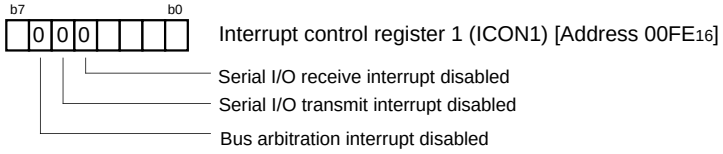
Figure 1.14.25 Timing of Bus Collision Detection

1.14 Serial I/O

(3) Setting of Bus Arbitration Interrupt

Figure 1.14.26 shows the setting of bus arbitration interrupt.

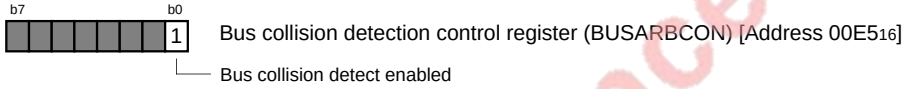
Procedure 1 Disabling of the acceptance of the interrupts associated serial I/O



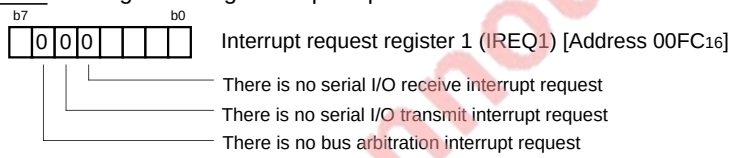
Procedure 2 Setting serial I/O full-duplex communication

1. Set Baud Rate Generator when the divided BRG output/4 or 16 is selected as the synchronous clock.
2. Set Serial I/O Control Register (in full-duplex communication).
3. Set UART Control Register (in UART).

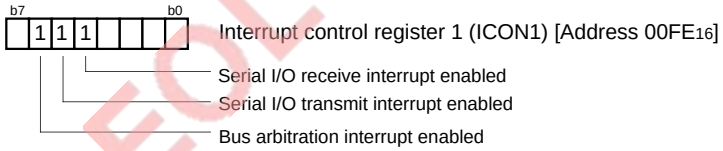
Procedure 3 Setting bus collision detection control register



Procedure 4 Setting the using interrupt request bit to '0'



Procedure 5 Enabling the acceptance of the using interrupts



Procedure 6 Writing transmit data into transmit buffer register

Figure 1.14.26 Setting of Bus Arbitration Interrupt

1.15 A-D Converter

The 7480 Group and 7481 Group have a built-in A-D converter with:

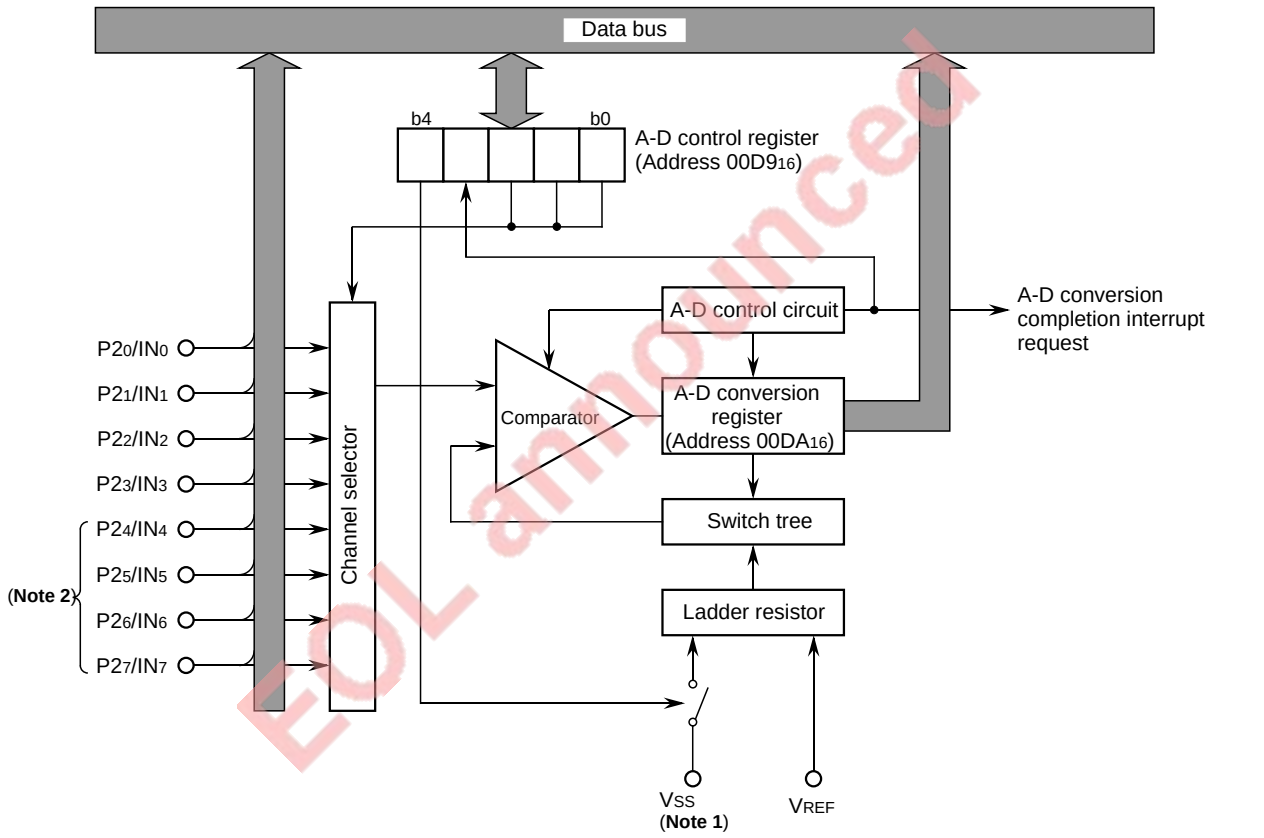
- analog input pins ..... 8 channels (alternative functions of port P2) (**Note**), and
- conversion system ..... 8-bit successive comparison.

When the A-D converter is not used, power dissipation can be reduced by clearing the V_{REF} connection selection bit of the A-D control register to '0' and switching off V_{REF}.

Note: In the 7480 Group, 4-channel analog input pins are implemented.

1.15.1 Block Diagram of A-D Converter

Figure 1.15.1 shows the block diagram of the A-D converter.



Notes 1: AV_{SS} for the 44P6N package of the 7481 Group.
2: Port pins $P2_4/IN_4$–$P2_7/IN_7$ are not implemented in the 7480 Group.

Figure 1.15.1 Block Diagram of A-D Converter

1.15.2 Registers Associated with A-D Converter

Figure 1.15.2 shows the memory map of the registers associated with the A-D converter.

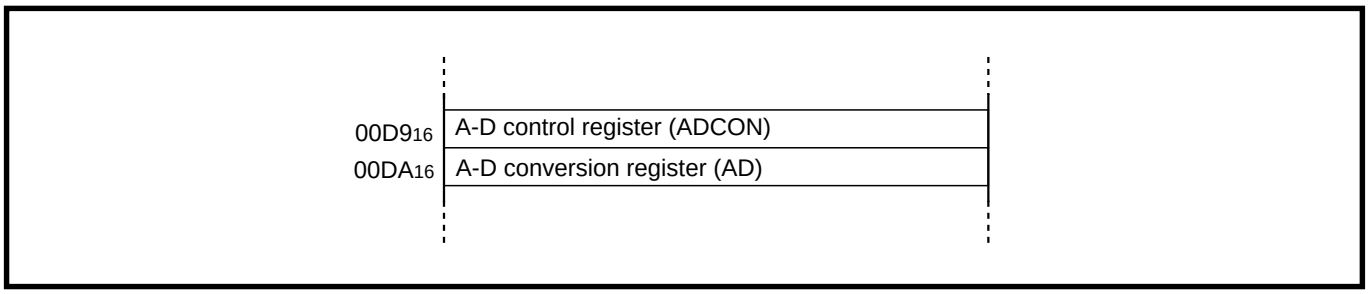


Figure 1.15.2 Memory Map of Registers Associated with A-D Converter

(1) A-D Control Register

The A-D control register consists of the bits controlling the A-D converter. Figure 1.15.3 shows the A-D control register.

A-D control register

b7 b6 b5 b4 b3 b2 b1 b0

A-D control register (ADCON) [Address 00D9$_{16}$]

b	Name	Function	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0 : P2$_0$/IN$_0$ 0 0 1 : P2$_1$/IN$_1$ 0 1 0 : P2$_2$/IN$_2$ 0 1 1 : P2$_3$/IN$_3$ 1 0 0 : P2$_4$/IN$_4$ 1 0 1 : P2$_5$/IN$_5$ 1 1 0 : P2$_6$/IN$_6$ 1 1 1 : P2$_7$/IN$_7$ (**Note**)	0	O	O
1			0	O	O
2			0	O	O
3	A-D conversion completion bit	0 : Conversion in progress 1 : Conversion completed	1	O	*
4	V_{REF} connection selection bit	0 : Disconnect between V_{REF} pin and ladder resistor 1 : Connect between V_{REF} pin and ladder resistor	0	O	O
5	Not implemented. Writing to these bits is disabled. These bits are undefined at reading.		Undefined	Undefined	×
6			Undefined	Undefined	×
7			Undefined	Undefined	×

*: The bit can be set to '0' by software, but cannot be set to '1'.

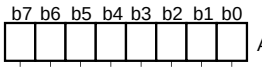
Note: Do not perform setting in the 7480 Group.

Figure 1.15.3 A-D Control Register

(2) A-D Conversion Register

This is a read-only register in which an A-D conversion result is stored. Figure 1.15.4 shows the A-D conversion register.

A-D conversion register



A-D conversion register (AD) [Address $00DA_{16}$]

b	Function	At reset	R	W
0	Read-only register to store the A-D conversion result.	Undefined	O	×
1		Undefined	O	×
2		Undefined	O	×
3		Undefined	O	×
4		Undefined	O	×
5		Undefined	O	×
6		Undefined	O	×
7		Undefined	O	×

Figure 1.15.4 A-D Conversion Register

1.15.3 Operations of A-D Converter

The A-D conversion system of the 7480 Group and 7481 Group is successive comparison conversion. The comparison result of internally generated comparison voltage V_{ref} with input voltage V_{IN} from an analog input pin is stored in the A-D conversion register.
The operations of the A-D converter are described below.

Start of A-D Conversion

When the A-D conversion completion bit of the A-D control register is cleared to '0', A-D conversion is started.

A-D Conversion

① The A-D conversion register goes to '00_{16}'.

② Analog input voltage V_{IN} is compared with comparison voltage V_{ref} 8 times. The contents of the A-D conversion register are determined by the bit from the MSB, each time a comparison is performed. Comparison voltage V_{ref} is determined by the following formula depending on the contents of the A-D conversion register and reference voltage V_{REF}, which is input from the V_{REF} pin.

Expression of comparison voltage V_{ref}

$$V_{ref} = \begin{cases} 0 & \text{when } n = 0 \\ \dfrac{V_{REF}}{256} \times (n - 0.5) & \text{when } n = 1 \text{ to } 255 \end{cases}$$

V_{REF}: Reference voltage input from the V_{REF} pin
n : The contents of the A-D conversion register

■ The first comparison (determination of bit 7 of A-D conversion register)
Bit 7 of the A-D conversion register is set to '1', and comparison voltage V_{ref} obtained by the above formula is input to the comparator. V_{ref} is compared with V_{IN}, and bit 7 of the A-D conversion register is determined, depending on the result of the comparison as follows:
- Bit 7 remains '1' (retention) if $V_{ref} < V_{IN}$.
- Bit 7 is converted to '0' if $V_{ref} > V_{IN}$.

■ Comparison from the second time (determination of bits 6 to 0 of A-D conversion register)
Every bit of bits 6 to 0 of the A-D conversion register is successively determined as bit 7 is done in the first time. (The next bit to be determined is set to '1', and the value of the bit is determined by the comparison result of V_{IN} with V_{ref}.)

Figure 1.15.5 shows the change of the A-D conversion register and the comparison voltage during A-D conversion.

③ A-D conversion is completed when comparison voltage V_{ref} is compared with analog input voltage V_{IN} 8 times and all bits of the A-D conversion register are determined. At this time, the A-D conversion completion bit of the A-D control register becomes '1'.

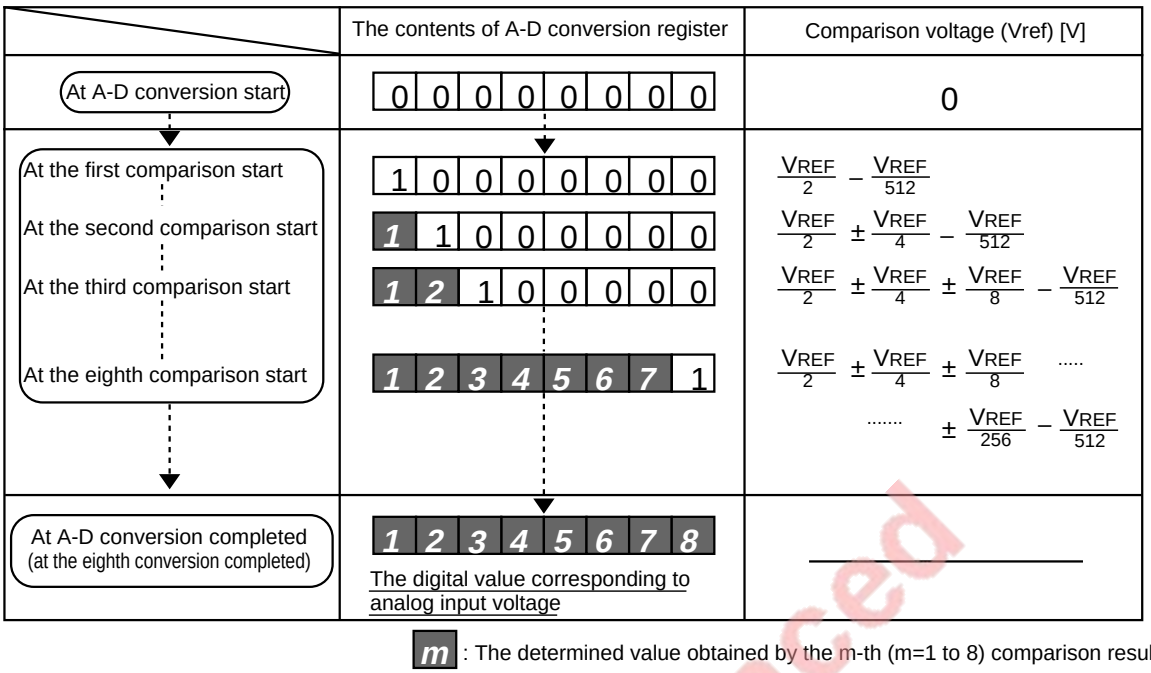


Figure 1.15.5 Change of A-D Conversion Register and Comparison Voltage during A-D Conversion

A-D conversion interrupt

When A-D conversion is completed, the A-D conversion completion interrupt request bit of interrupt request register 1 is set to '1'; then an A-D conversion completion interrupt request is generated.

Reads from A-D conversion register

When A-D conversion is completed, the A-D conversion register is read to obtain the A-D conversion result. The completion of A-D conversion can be acknowledged by any of the following conditions:

- The A-D conversion completion bit is '1'.
- The A-D conversion completion interrupt request bit is '1'.
- The branch to A-D conversion completion interrupt service routine occurs (when A-D conversion completion interrupt enabled).

Note: Do not read from the A-D conversion register during A-D conversion operation.

A-D conversion time

A-D conversion ends in 50 cycles after its start. Because the A-D converter uses the clock input divided by 2, f(XIN)/2, as the operating clock, A-D conversion time is fundamentally obtained by the following formula:

$$\text{A-D conversion time} = \frac{2}{f(X_{IN})} \times \text{conversion cycles (50 cycles)}$$

(12.5 μs at f(XIN) = 8 MHz)

1.15.4 Setting of A-D Conversion

Figure 1.15.6 shows the setting of A-D conversion.

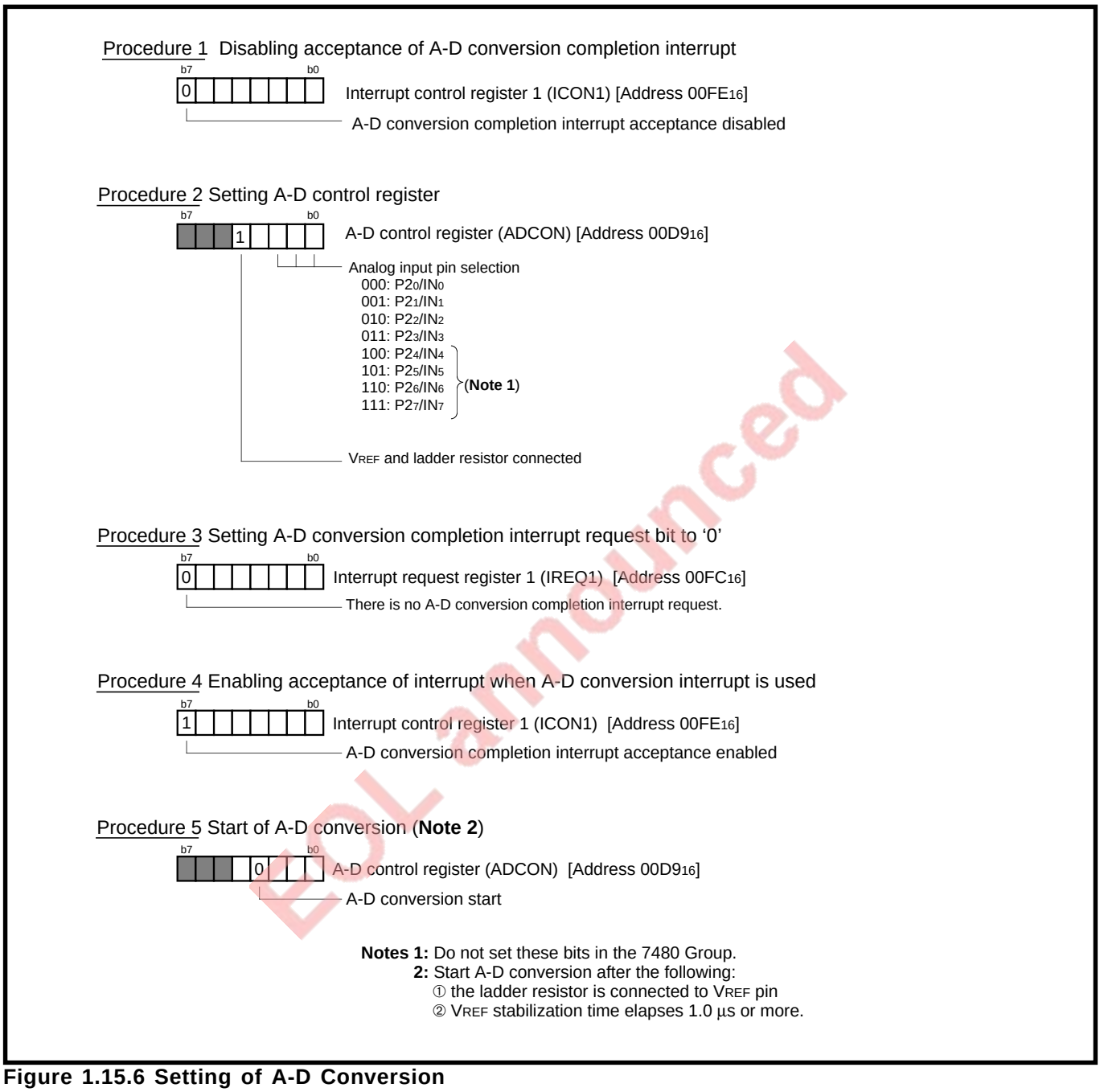


Figure 1.15.6 Setting of A-D Conversion

1.15.5 Notes on Usage

Pay attention to the following notes when the A-D converter is used.

■ The comparator consists of a capacitive coupling circuit, so that low clock input frequencies cause electric charge to be lost.
- Use 1 MHz or more of f(XIN) during A-D conversion is performed.
- Do not execute the **STP** instruction during A-D conversion.

■ Voltages to be applied to the reference voltage input pin are as follows:
- VREF = 2 to VCC [V] when VCC = 2.7 V to 4.0 V
- VREF = 0.5 VCC to VCC [V] when VCC = 4.0 V to 5.5 V

When the A-D converter is not used, connect VREF pin to the VCC pin.

■ Apply the same voltage as to the VSS pin to analog power source voltage input pin AVSS. (The AVSS pin is dedicated to the 44P6N-A package in the 7481 Group.)

■ Even when A-D conversion is started, the A-D conversion completion interrupt request bit is not automatically cleared to '0'.
Clear this bit to '0' before A-D conversion starts.

■ A-D conversion resumes when '0' is written into the A-D conversion completion bit of the A-D control register during A-D conversion.

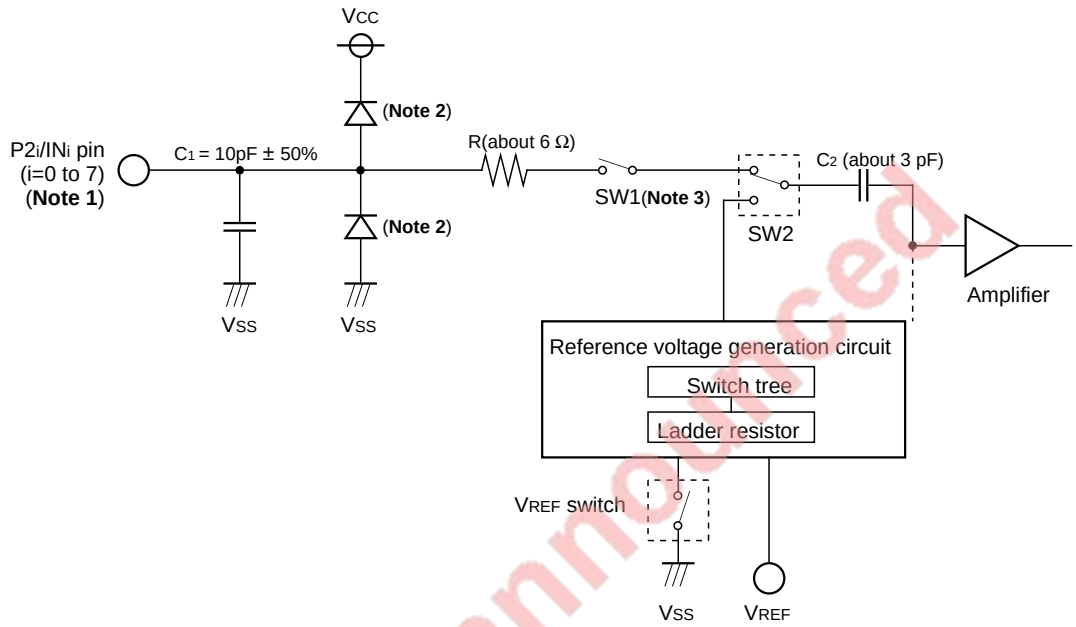
■ To start A-D conversion, set the VREF connection selection bit of the A-D control register to '1' to connect ladder resistor and the VREF. A-D conversion can then be started, after the VREF stabilization time elapses 1.0 μs or more.

1.15 A-D Converter

■ Figure 1.15.7 shows the internal equivalent circuit of analog input circuit. In order to perform the A-D conversion correctly, complete the charge to the internal capacitor by the specified time. The maximum output impedance of analog input source to complete the charge to the internal capacitor by this specified time is shown below.

About 10 kΩ (at f(XIN) = 8 MHz)

When the maximum value of output impedance is over the above value, take countermeasures, for example, connect a capacitor (0.1 μF to 1 μF) between analog input pins and VSS.



Notes 1: In the 7480 Group, i = 0 to 3.
2: This is a parasitic diode of output transistor.
3: SW1 is turned on only when analog input pin is selected.

Figure 1.15.7 Internal equivalent circuit of analog input circuit

1.16 Watchdog Timer

If a program cannot run a normal loop by a runaway, etc., the watchdog timer provides the means of returning the CPU to the reset state.

In the 7480 Group and 7481 Group, invalidating the **STP** and **WIT** instructions causes a runaway to be detected more effectively. For the selection of the valid/invalid of the **STP** and **WIT** instructions, refer to **Section 1.19 Power Saving Function**.

The watchdog timer is comprised of 7-bit watchdog timer L and 8-bit watchdog timer H (address 00FE16).

1.16.1 Block Diagram of Watchdog Timer

Figure 1.16.1 shows the block diagram of the watchdog timer.

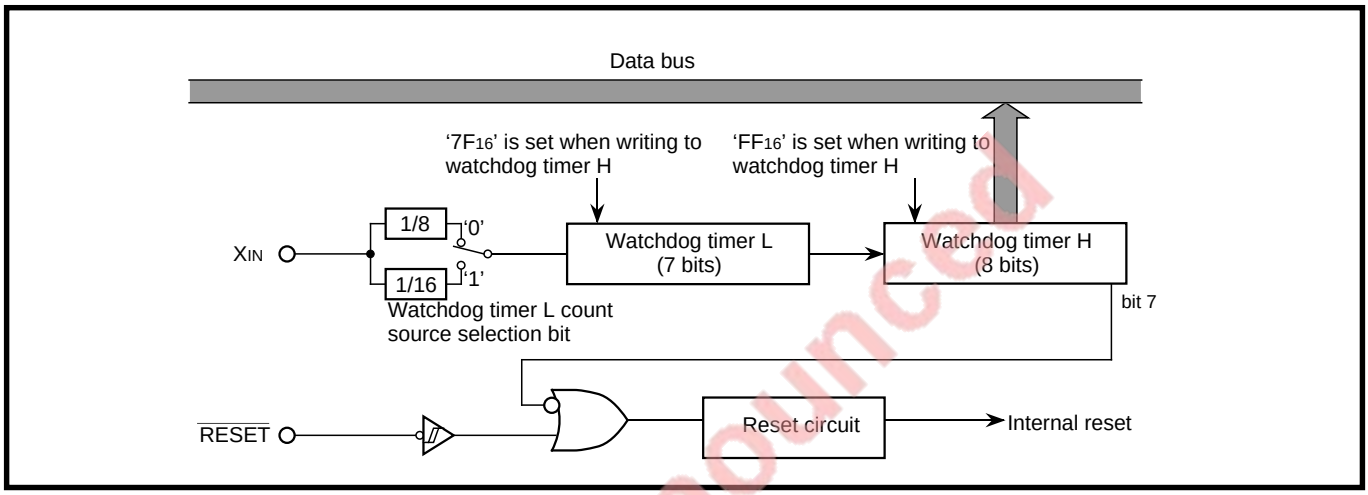


Figure 1.16.1 Block Diagram of Watchdog Timer

1.16.2 Registers Associated with Watchdog Timer

Figure 1.16.2 shows the memory map of the registers associated with the watchdog timer.

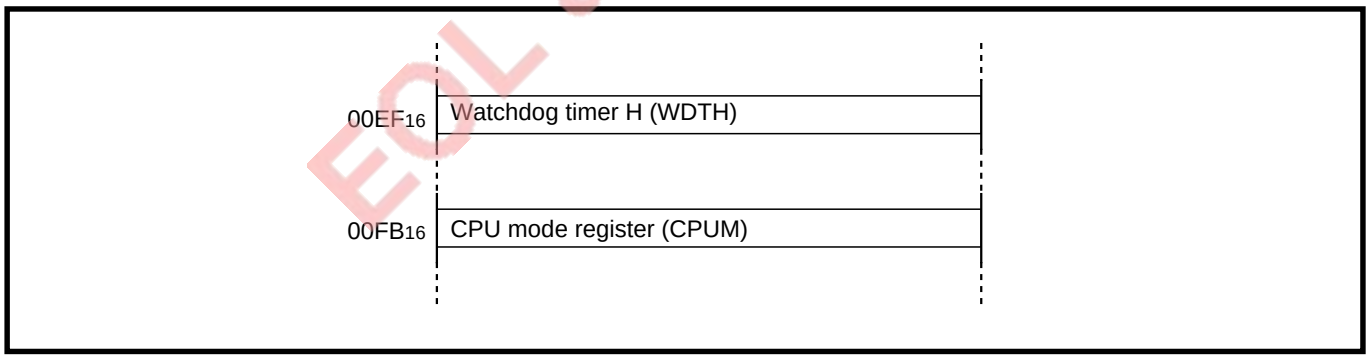


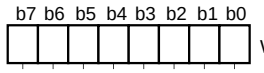
Figure 1.16.2 Memory Map of Registers Associated with Watchdog Timer

1.16 Watchdog Timer

(1) Watchdog Timer H

Watchdog timer H indicates the high-order 8 bits of the count value of the watchdog timer. Figure 1.16.3 shows the watchdog timer H.

Watchdog timer H



Watchdog timer H (WDTH) [Address 00EF16]

b	Function	At reset	R	W
0	The high-order count value of watchdog timer is indicated.	1	O	Note
1		1	O	
2		1	O	
3		1	O	
4		1	O	
5		1	O	
6		1	O	
7		1	O	

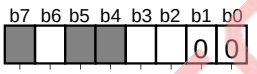
Note: The following value is set by writing arbitrary data.
- watchdog timer L ← '7F16'
- watchdog timer H ← 'FF16'

Figure 1.16.3 Watchdog Timer H

(2) CPU Mode Register

This register consists of the bits that select a stack page and an internal clock, as well as the bit that selects a count source of the watchdog timer.
Figure 1.16.4 shows the CPU mode register.

CPU mode register



CPU mode register (CPUM) [Address 00FB16]

b	Name	Function	At reset	R	W
0	Fix these bits to '0'		0	O	0
1			0	O	0
2	Stack page selection bit (**Note**)	0 : Zero page 1 : 1 page	0	O	O
3	Watchdog timer L count source selection bit	0 : f(XIN)/8 1 : f(XIN)/16	0	O	O
4	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
5	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
6	Clock division ratio selection bit	0 : f(XIN)/2 (high-speed mode) 1 : f(XIN)/8 (medium-speed mode)	0	O	O
7	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×

Note: In the products whose RAM size is 192 bytes or less, set this bit to '0'.

Figure 1.16.4 CPU Mode Register

1.16.3 Operations of Watchdog Timer

The operations of the watchdog timer are described below.

Count Source

The watchdog timer can select the following count sources using watchdog timer L count source selection bit of the CPU mode register:

- f(XIN)/8 when the bit is '0'
- f(XIN)/16 when the bit is '1'

Internal Operation

① When the write instructions (**Note 1**) are executed to watchdog timer H, the following values are placed in watchdog timers H and L, regardless of the written value:
- 'FF16' into watchdog timer H
- '7F16' into watchdog timer L

The watchdog timer starts counting by writing to watchdog timer H, and every time the count source is input, the watchdog timer is decremented by 1.

② When bit 7 of watchdog timer H becomes '0' by a down count (**Note 2**), the internal reset signal changes from HIGH to LOW and the CPU enters the reset state. As a result, the internal state of the microcomputer is set as shown in **Figure 1.17.2 Internal State at Reset** in **Section 1.17 Reset**. Timer 1 goes to 'FF16' to generate the wait time for the system releasing from reset, and then the watchdog timer starts counting, using f(XIN)/8 as the count source.

③ When an underflow occurs in timer 1, the internal reset signal is raised to the HIGH state and the system is released from reset. The program is executed at the address stored in the reset vector area.

Notes 1: Write instructions which generate write signals, such as **STA**, **LDM**, and **CLB**.

2: The time from writing data into watchdog timer H to placing '0' in bit 7 is 16384 (400016) cycles of the count source.

Examples

At f(XIN) = 8 MHz :
- 16.384 ms; when the frequency of the count source is f(XIN)/8
- 32.768 ms; when the frequency of the count source is f(XIN)/16

Figure 1.16.5 shows the internal processing sequence during reset by the watchdog timer.

1.16 Watchdog Timer

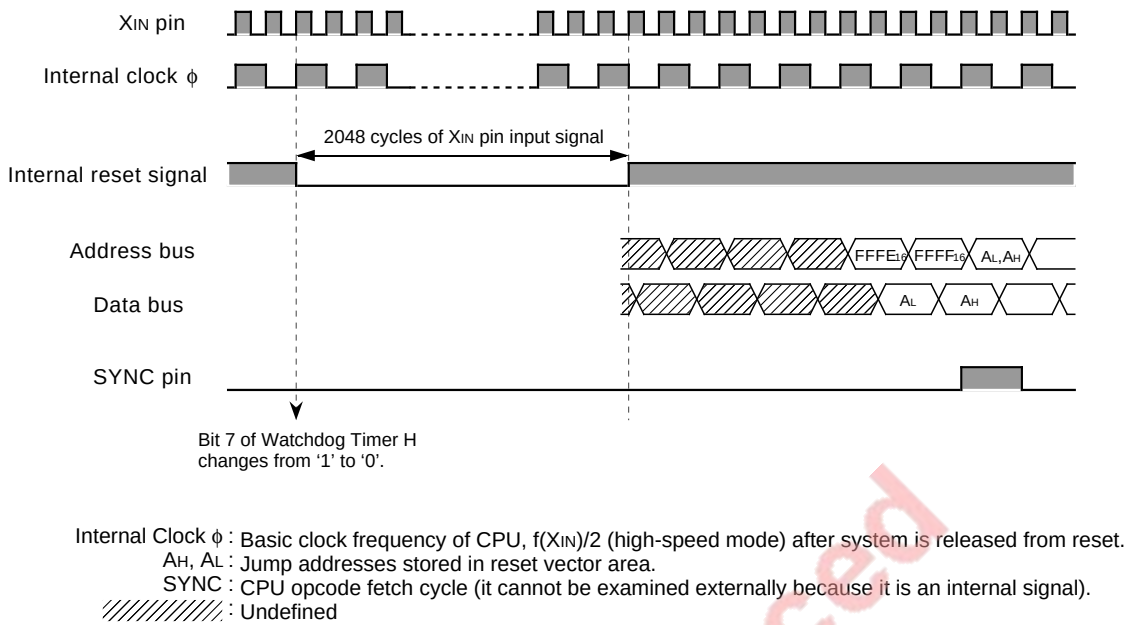


Figure 1.16.5 Internal Processing Sequence during Reset by Watchdog Timer

Countermeasures with Watchdog Timer against Runaway

Watchdog timer H is written to with the main routine, etc., to keep bit 7 from going to '0.' (keep it at '1'). In case of a program runaway by noise, watchdog timer H is not written to, so that bit 7 becomes '0' and the CPU returns to the reset state.

Operations in Stop and Wait Modes

- When the **STP** instruction is executed to enter the stop mode, the f(XIN) stops, causing the watchdog timer to stop counting. When the stop mode is terminated, the watchdog timer starts counting in response to the restarting of f(XIN) oscillation.
- When the **WIT** instruction is executed to enter the wait mode, CPU stops operating, whereas the watchdog timer continues counting because the oscillation of f(XIN) does not stop.

Note: The watchdog timer continues counting even during the oscillator start-up stabilization time (2048 cycles of the XIN pin input signal) after the stop mode is terminated, and the wait mode. Write to watchdog timer in order to prevent bit 7 of watchdog timer H from going to '0'.

In the 7480 Group and 7481 Group, the valid/invalid of the **STP** and **WIT** instructions can be selected. Invalidating these instructions causes a runaway to be detected more effectively when the watchdog timer is used.

For details on the setting of the valid/invalid of the stop and wait modes and the **STP** and **WIT** instructions, refer to **Section 1.19 Power Saving Function**.

1.16.4 Setting of Watchdog Timer

Figure 1.16.6 shows the setting of the watchdog timer.

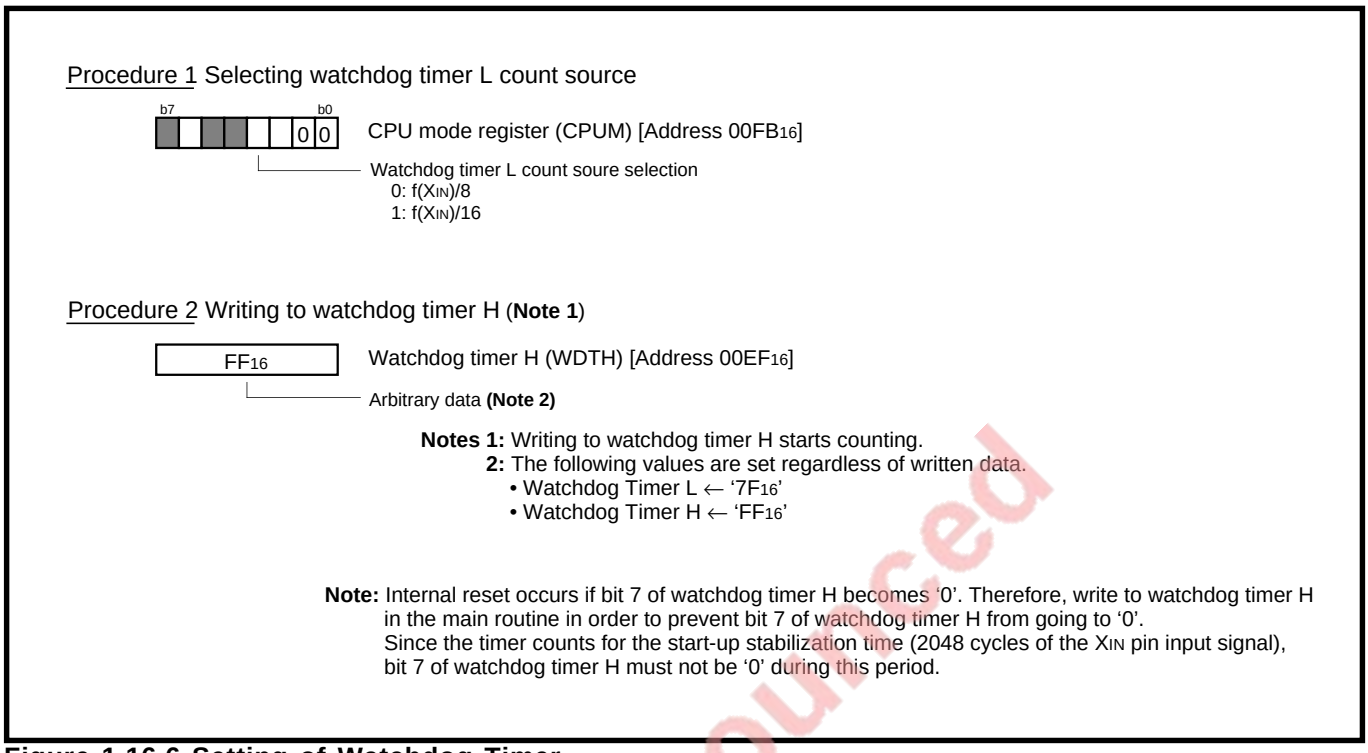


Figure 1.16.6 Setting of Watchdog Timer

1.16.5 Notes on Usage

Pay attention to the following notes when the watchdog timer is used.

- Write to watchdog timer in the main routine in order to prevent bit 7 of watchdog timer H from going to '0'.
- The watchdog timer continues counting even during the oscillator start-up stabilization time (2048 cycles of the XIN pin input signal) after the stop mode is terminated, and the wait mode. Write to watchdog timer in order to prevent bit 7 of watchdog timer H from going to '0'.
- Do not operate the watchdog timer during system evaluation.

1.17 Reset

When the LOW level is applied to the $\overline{\text{RESET}}$ pin for 2 μs or more, the internal reset signal becomes LOW, and the CPU enters the reset state. Subsequently, when the HIGH level is applied to the $\overline{\text{RESET}}$ pin, the internal reset signal becomes HIGH, and the system is released from reset after the oscillator start-up stabilization time (**Note**) elapses. The program is resumed at the jump address stored in the reset vector area after the system is released from reset.
In the 7480 Group and 7481 Group, even when bit 7 of watchdog timer H changes from '1' to '0', the internal reset signal changes from HIGH to LOW, causing the CPU to enter the reset state. For details of the watchdog timer, refer to **Section 1.16 Watchdog Timer**.

Note: 2048 cycles of the XIN pin input signal (counted by timer 1).

For an example of reset circuits, refer to **Section 2.5 Reset**.

1.17.1 Reset Operations

The reset operations are described below.

① When the power source voltage is within specifications (**Note**) and clock input oscillation frequency f(XIN) is stabled, the internal reset signal changes from HIGH to LOW by applying the LOW level to the $\overline{\text{RESET}}$ pin for 2 μs or more, causing the CPU to enter the reset state.
② Then the HIGH level is applied to the $\overline{\text{RESET}}$ pin, so that the internal state of the microcomputer is set, as shown in **Figure 1.17.2 Internal State at Reset**. Timer 1 goes to 'FF16' to generate the f(XIN) oscillator start-up stabilization time and then starts counting, using f(XIN)/8 as the count source.
③ When an underflow occurs in timer 1, the internal reset signal becomes HIGH and the system is released from reset. The program is resumed at the address stored in the reset vector area.

Note: • 2.7 V to 4.5 V at f(XIN) = (2.2 VCC–2) MHz
• 4.5 V to 5.5 V at f(XIN) = 8 MHz

Figure 1.17.1 shows the internal processing sequence after reset release.

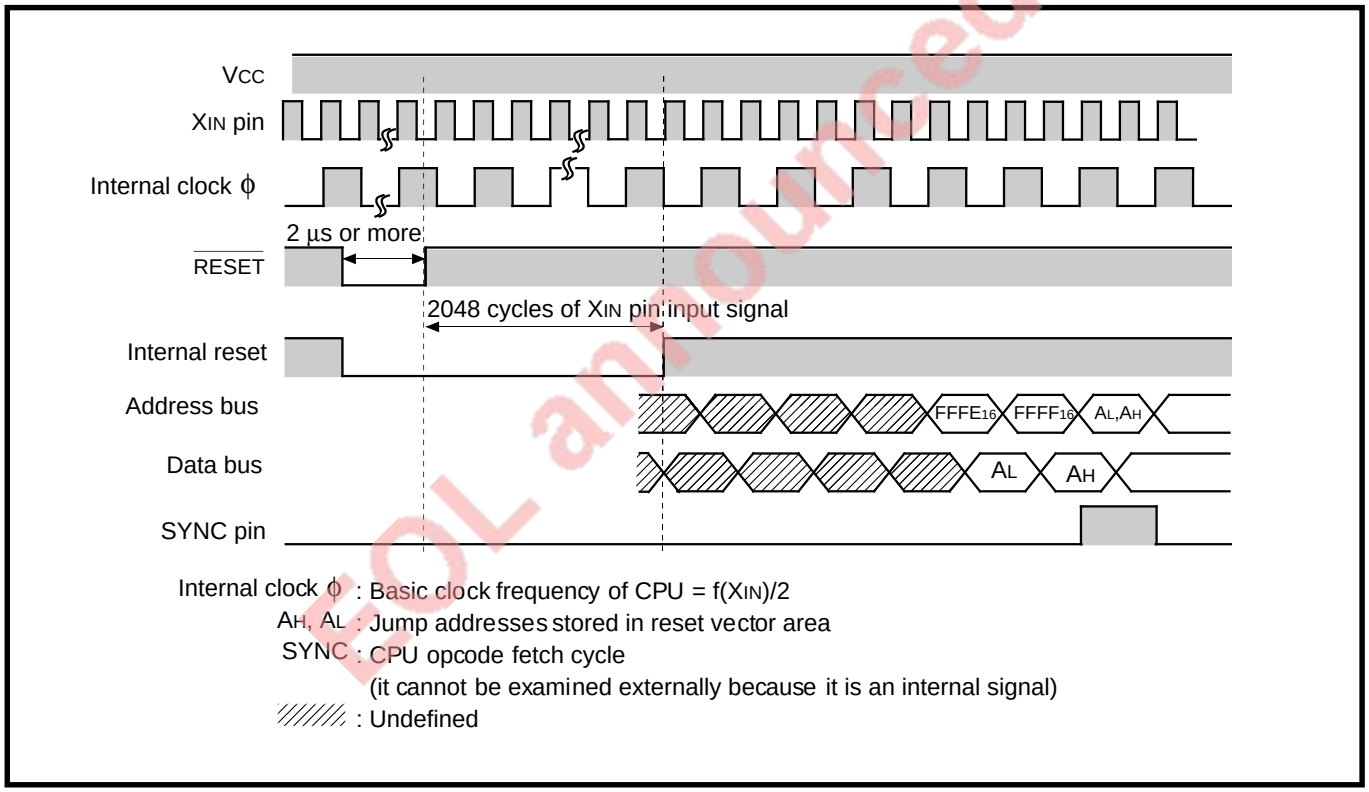


Figure 1.17.1 Internal Processing Sequence after Reset Release

1.17.2 Internal State at Reset

Figure 1.17.2 shows the internal state at reset.

		Address	Contents of register (b7 … b0)
(1)	Port P0 direction register (P0D)	$00C1_{16}$	00_{16}
(2)	Port P1 direction register (P1D)	$00C3_{16}$	00_{16}
(3)	Port P4 direction register (P4D)	$00C9_{16}$	? ? ? ? 0 0 0 0
(4)	Port P5 direction register (P5D)	$00CB_{16}$	? ? ? ? 0 0 0 0
(5)	Port P0 pull-up control register (P0PCON)	$00D0_{16}$	00_{16}
(6)	Port P1 pull-up control register (P1PCON)	$00D1_{16}$	? ? ? ? ? ? 0 0
(7)	Port P4P5 input control register (P4P5CON)	$00D2_{16}$	00_{16}
(8)	Edge polarity selection register (EG)	$00D4_{16}$	? ? 0 ? 0 0 0 0
(9)	A-D control register (ADCON)	$00D9_{16}$	? ? ? 0 1 0 0 0
(10)	STP instruction operation control register (STPCON)	$00DE_{16}$	0 0 0 0 0 0 0 1
(11)	Serial I/O status register (SIOSTS)	$00E1_{16}$	1 0 0 0 0 0 0 0
(12)	Serial I/O control register (SIOCON)	$00E2_{16}$	00_{16}
(13)	UART control register (UARTCON)	$00E3_{16}$	1 1 1 1 0 0 0 0
(14)	Bus collision detection control register (BUSARBCON)	$00E5_{16}$	00_{16}
(15)	Watchdog timer H (WDTH)	$00EF_{16}$	FF_{16}
(16)	Timer X low-order (TXL)	$00F0_{16}$	FF_{16}
(17)	Timer X high-order (TXH)	$00F1_{16}$	FF_{16}
(18)	Timer Y low-order (TYL)	$00F2_{16}$	FF_{16}
(19)	Timer Y high-order (TYH)	$00F3_{16}$	FF_{16}
(20)	Timer 1 (T1)	$00F4_{16}$	FF_{16}
(21)	Timer X mode register (TXM)	$00F6_{16}$	00_{16}
(22)	Timer Y mode register (TYM)	$00F7_{16}$	00_{16}
(23)	Timer XY control register (TXYCON)	$00F8_{16}$	0 0 0 0 0 0 1 1
(24)	Timer 1 mode register (T1M)	$00F9_{16}$	00_{16}
(25)	Timer 2 mode register (T2M)	$00FA_{16}$	00_{16}
(26)	CPU mode register (CPUM)	$00FB_{16}$	? 0 ? ? 0 0 0 0
(27)	Interrupt request register 1 (IREQ1)	$00FC_{16}$	00_{16}
(28)	Interrupt request register 2 (IREQ2)	$00FD_{16}$	? ? ? ? 0 0 0 0
(29)	Interrupt control register 1 (ICON1)	$00FE_{16}$	00_{16}
(30)	Interrupt control register 2 (ICON2)	$00FF_{16}$	? ? ? ? 0 0 0 0
(31)	Program counter (PC_H)		Contents of $FFFF_{16}$
	(PC_L)		Contents of $FFFE_{16}$
(32)	Processor status register (PS)		? ? ? ? ? 1 ? ?

? : Read back as undefined at reset.

Note: Since the contents of the registers and RAM not mentioned above are undefined at reset, initialize them by software.
There are bits not implemented for some products.
For these bits, refer to each register.

Figure 1.17.2 Internal State at Reset

1.17.3 Notes on Usage

Pay attention to the following notes when reset is used.

- Internal clock ϕ becomes f(XIN)/2 (high-speed mode) when the system is released from reset.
- Timer 1 and timer 2 are counting when the system is released from reset.
- Apply 0.32 V or less to the $\overline{\text{RESET}}$ pin at the time that power source voltage passes 2.7V, at power on.
- When the **STP** instruction is executed in normal operations, I/O port pins retain the states immediately before internal clock ϕ stops. If the CPU is then forced to the reset state from the stop mode, the I/O pins go to the input mode with the high-impedance state.

1.18 Oscillation Circuit

The 7480 Group and 7481 Group are equipped with a built-in clock generator providing the clock necessary for operation of the microcomputer. An oscillation circuit is constructed by connecting a ceramic resonator between the XIN and XOUT pins (**Note 1**). Also, an external clock can be supplied to the clock generator (**Note 2**).

The built-in feedback resistor connected between the XIN and XOUT pins allows the user to omit an external resistor.

Notes 1: For an example of an oscillation circuit using a ceramic resonator, refer to **Section 2.6 Oscillation Circuit**. Consult the manufacturer of the resonator for the oscillator start-up stabilization time.

2: Also, for an external clock circuit, refer to **Section 2.6 Oscillation Circuit**. Use a 50% duty cycle pulse signal as the external clock input to the XIN pin. At this time, leave the XOUT pin open.

1.18.1 Block Diagram of Clock Generator

The clock generator controls the oscillation circuit. The generated clock (internal clock ϕ) is supplied to the CPU and the peripherals.

Figure 1.18.1 shows the block diagram of the clock generator.

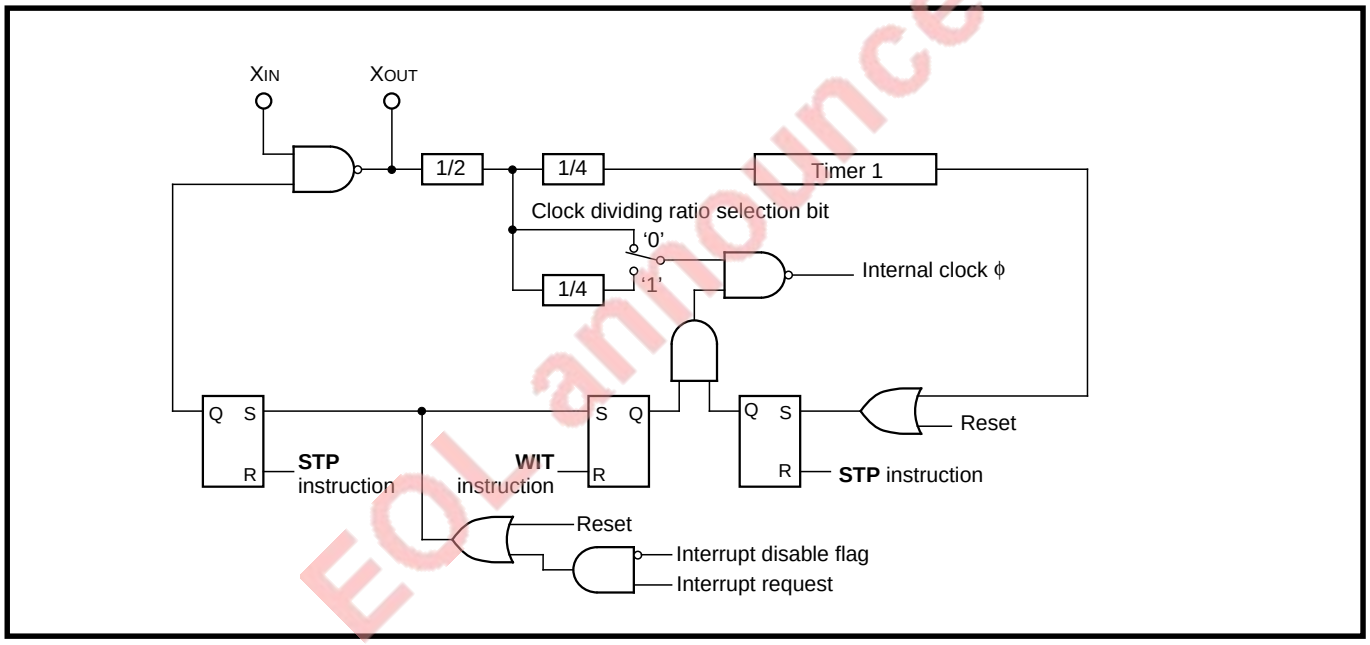


Figure 1.18.1 Block Diagram of Clock Generator

1.18.2 Register Associated with Oscillation Circuit

Figure 1.18.2 shows the memory map of the register associated with the oscillation circuit.

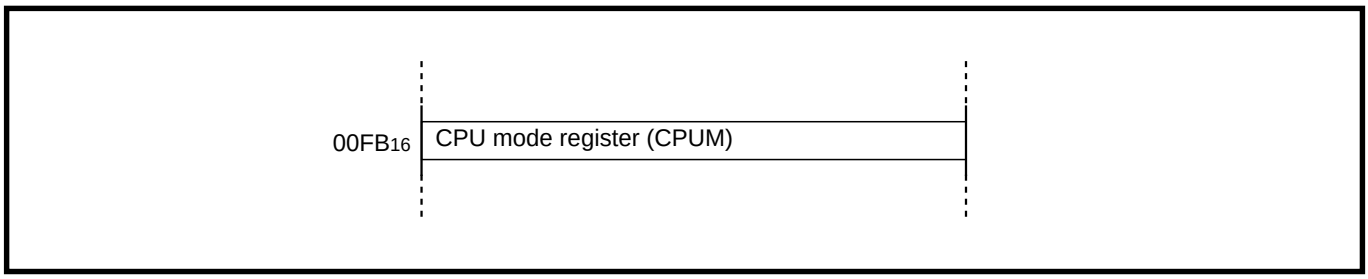


Figure 1.18.2 Memory Map of Register Associated with Oscillation Circuit

The CPU mode register consists of the bits that select a stack page and an internal clock, as well as the bit that selects a count source of the watchdog timer.
Figure 1.18.3 shows the CPU mode register.

CPU mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

CPU mode register (CPUM) [Address 00FB16]

b	Name	Function	At reset	R	W
0	Fix these bits to '0'.		0	O	0
1			0	O	0
2	Stack page selection bit (**Note**)	0 : Zero page 1 : 1 page	0	O	O
3	Watchdog timer L count source selection bit	0 : f(XIN)/8 1 : f(XIN)/16	0	O	O
4	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
5	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
6	Clock division ratio selection bit	0 : f(XIN)/2 (high-speed mode) 1 : f(XIN)/8 (medium-speed mode)	0	O	O
7	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×

Note: In the products whose RAM size is 192 bytes or less, set this bit to '0'.

Figure 1.18.3 CPU Mode Register

1.18 Oscillation Circuit

1.18.3 Oscillation Operations

(1) Oscillation Operations

The following clocks can be selected as internal clock ϕ by the clock division ratio selection bit of the CPU mode register.

- X_{IN} pin input divided by 2 (high-speed mode) when the bit is '0'.
- X_{IN} pin input divided by 8 (medium-speed mode) when the bit is '1'.

Note: The oscillation circuit is held in the high-speed mode after the system is released from reset.

(2) Oscillation in Stop Mode

When the **STP** instruction is executed to enter the stop mode, internal clock ϕ stops in the HIGH state, and the oscillation of $f(X_{IN})$ stops, as well. At this time, timer 1 goes to 'FF_{16}', and $f(X_{IN})/8$ is selected as the count source.

CPU returns from stop mode by reset or accepting an external interrupt request (**Note 1**). In this time, internal clock ϕ is not supplied to the CPU until an underflow occurs in timer 1, though the oscillation of $f(X_{IN})$ and internal clock ϕ are started. The reason is that oscillator start-up stabilization time is required when an external resonator is used.

Note: Activate timer 1 and disable the acceptance of a timer 1 interrupt request before the **STP** instruction is executed.

Notes 1: For interrupt sources that can be used to return from the stop mode, refer to **Table 1.11.2 Interrupt Sources Available for CPU's Return from Stop/Wait Mode**.

For details of the stop mode, refer to **Section 1.19.2 Stop Mode**.

(3) Oscillation in Wait Mode

When the **WIT** instruction is executed to enter the wait mode, only internal clock ϕ stops in the HIGH state.

When the CPU returns from the wait mode by reset or accepting an interrupt request (**Note 2**), the supply of internal clock ϕ to the CPU is resumed. Since $f(X_{IN})$ continues oscillation during the wait mode, instructions can be executed immediately after the CPU returns from the wait mode.

Notes 2: For interrupt sources that can be used to return from the wait mode, refer to **Table 1.11.2 Interrupt Sources Available for CPU's Return from Stop/Wait Mode**.

For details of the wait mode, refer to **Section 1.19.3 Wait Mode**.

1.18.4 Oscillator Start-Up Stabilization Time

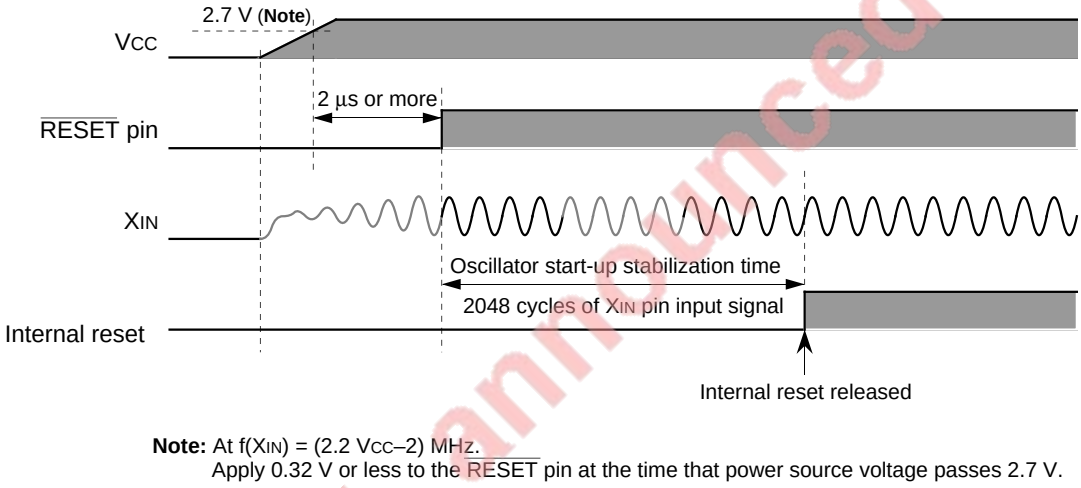
Oscillation is unstable immediately after oscillation is started in the oscillation circuit which uses a ceramic resonator. Necessary time for stabilizing oscillation is called an oscillator start-up stabilization time.
The oscillator start-up stabilization time necessitated varies with the structure of the oscillation circuit used. Consult the manufacturer of the resonator for the oscillator start-up stabilization time.

(1) Oscillator Start-Up Stabilization Time at Power On

The oscillator start-up stabilizing time of 2048 cycles of the XIN pin input signal is automatically generated after the system is released from reset by timer 1 in the 7480 Group and 7481 Group (**Note**).

Note: Timer 1 goes to 'FF16' to select f(XIN)/8 as the count source.

Figure 1.18.4 shows an oscillator start-up stabilization time at power on.



Note: At f(XIN) = (2.2 VCC–2) MHz.
Apply 0.32 V or less to the RESET pin at the time that power source voltage passes 2.7 V.

Figure 1.18.4 Oscillator Start-Up Stabilization Time at Power On

(2) Oscillator Start-Up Stabilization Time after Stop Mode

Oscillation stops in the stop mode. When the CPU returns from the stop mode by reset or accepting an interrupt request, the oscillator start-up stabilization time of 2048 cycles of the input signal to the XIN pin is automatically generated by timer 1, as occurs at power on.

1.18.5 Notes on Usage

Pay attention to the following notes when an oscillation circuit is used.

- The oscillation circuit is held in the high-speed mode after the system is released from reset.
- When a ceramic resonator is connected between the XIN and XOUT pins, consult the manufacturer of the resonator for the oscillator start-up stabilization time.
- When an external clock is input to the XIN pin, use a 50% duty cycle pulse signal as the external clock input to the XIN pin. At this time, leave the XOUT pin open.
- Activate timer 1 and disable the acceptance of a timer 1 interrupt request before the **STP** instruction is executed.

1.19 Power Saving Function

The 7480 Group and 7481 Group are provided with the function to halt the CPU operation and make it stand by in the following two power saving modes by software:

- Stop mode with the **STP** instruction
- Wait mode with the **WIT** instruction

Also, the valid/invalid of the **STP** and **WIT** instructions can be selected with the **STP** instruction operation control register.

Table 1.19.1 lists the states of the microcomputer in the power saving modes.

Figure 1.19.1 shows the transitions from the power saving modes.

Table 1.19.1 States of Microcomputer in Power Saving Modes

		Stop Mode	Wait Mode
Clock f(XIN)		Stopped	Operating
Internal Clock ϕ		Suspended at the HIGH level	Suspended at the HIGH level
CPU		Stopped	Stopped
I/O Ports		Retains the state at **STP** instruction execution	Retains the state at **WIT** instruction execution
Timers	Event Count Mode (external clock as count source)	Operating	Operating
	Other Modes (divided main clock as count source)	Stopped	
Serial I/O	Divided BRG output as synchronous clock	Stopped	Operating
	External clock or its 1/16 as synchronous clock	Operating	
RAM		Retains the state at **STP** instruction execution	Retains the state at **WIT** instruction execution
SFR	Registers associated with Timer 1	Used to generate oscillator start-up stabilization time	Retains the state at **WIT** instruction execution
	Other registers	Retains the state at **STP** instruction execution	
CPU Internal Registers (**Note**)		Retains the state at **STP** instruction execution	Retains the state at **WIT** instruction execution

Note: The CPU internal registers are composed of the following six registers:

- Accumulator
- Index register X
- Index register Y
- Stack pointer
- Program counter
- Processor status register

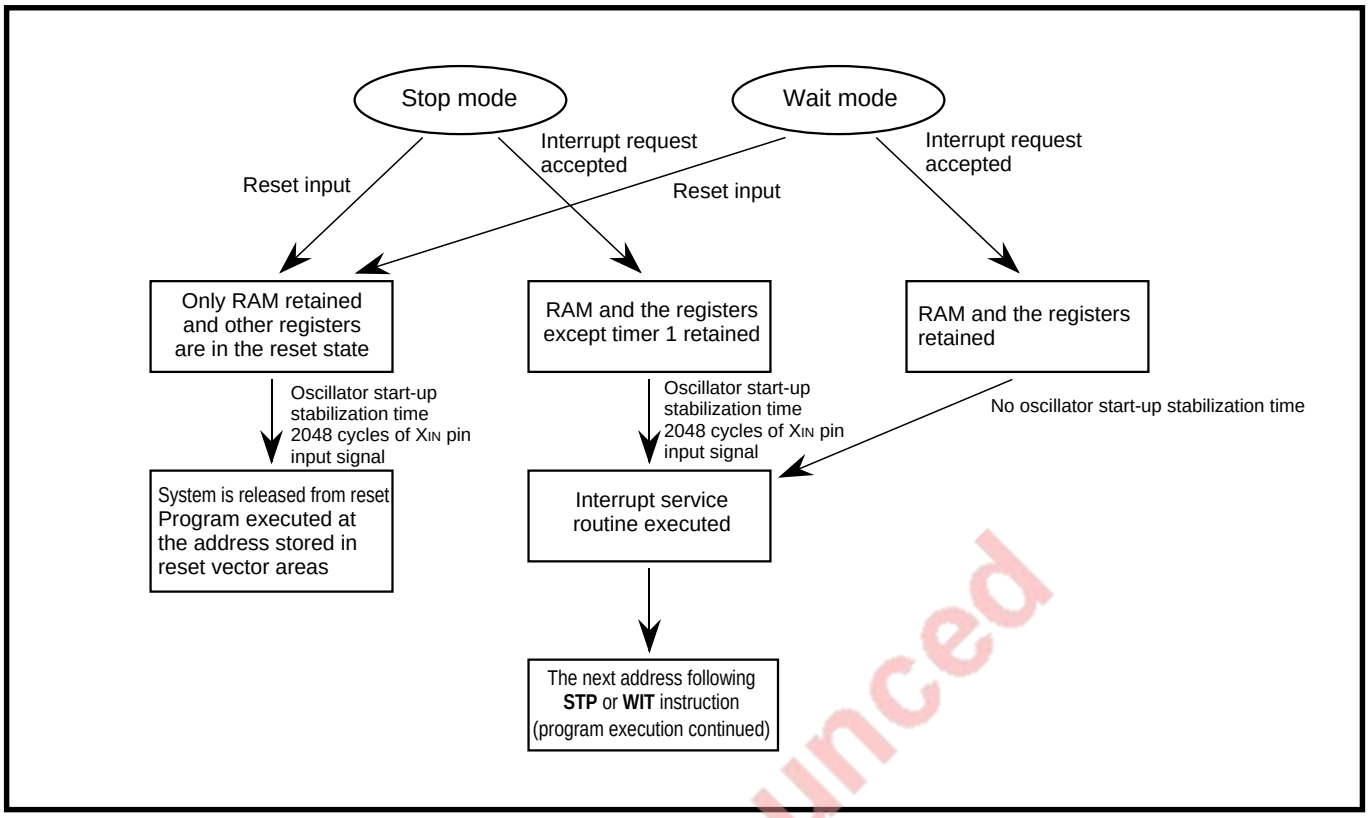


Figure 1.19.1 Transitions from Power Saving Modes

1.19 Power Saving Function

1.19.1 Registers Associated with Power Saving

Figure 1.19.2 shows the memory map of the registers associated with power saving.

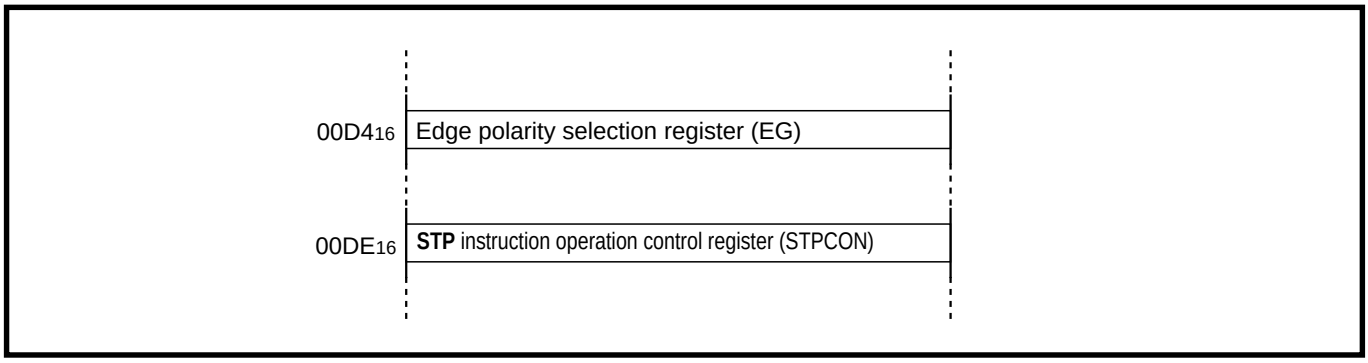


Figure 1.19.2 Memory Map of Registers Associated with Power Saving

(1) STP Instruction Operation Control Register

The **STP** instruction operation control register has only one bit that selects the valid/invalid of the **STP** and the **WIT** instruction.

Figure 1.19.3 shows the **STP** instruction operation control register.

STP instruction operation control register

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

STP instruction operation control register (STPCON) [Address 00DE₁₆]

b	Name	Function	At reset	R	W
0	**STP** and **WIT** valid/invalid selection bit (**Note**)	0 : **STP/WIT** instruction valid 1 : **STP/WIT** instruction invalid	1	O	O
1	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
2			0	0	×
3			0	0	×
4			0	0	×
5			0	0	×
6			0	0	×
7			0	0	×

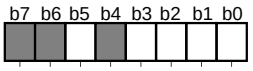
Note: The **STP** and **WIT** instructions are invalid after the system is released from reset. When using these instructions, set **STP** and **WIT** valid/invalid selection bit of the **STP** instruction operation control register to '1', then set this bit to '0'. (Writing twice successively)
When not using the **STP** and **WIT** instructions, set this bit to '1' either once or twice.

Figure 1.19.3 STP Instruction Operation Control Register

(2) Edge Polarity Selection Register

The edge polarity selection register consists of the bits that select the polarity of the valid edge of INT and CNTR pins, as well as the bit that selects the valid/invalid of key-on wakeup.
Figure 1.19.4 shows the edge polarity selection register.

Edge polarity selection register



Edge polarity selection register (EG) [Address $00D4_{16}$]

b	Name	Function	At reset	R	W
0	INT_0 edge selection bit	0 : Falling edge 1 : Rising edge	0	O	O
1	INT_1 edge selection bit	0 : Falling edge 1 : Rising edge	0	O	O
2	$CNTR_0$ edge selection bit	0: In event count mode, rising edge counted. : In pulse output mode, operation started at HIGH level output. : In pulse period measurement mode, a period from falling edge until falling edge measured. : In pulse width measurement mode, HIGH-level period measured. : In programmable one-shot output mode, one-shot HIGH pulse generated after operation started at LOW level output. : Interrupt request is generated by detecting falling edge. 1: In event count mode, falling edge counted : In pulse output mode, operation started at LOW level output. : In pulse period measurement mode, a period from rising edge until rising edge measured. : In pulse width measurement mode, LOW-level period measured. : In programmable one-shot output mode, one-shot LOW pulse generated after operation started at HIGH level output. : Interrupt request is generated by detecting rising edge.	0	O	O
3	$CNTR_1$ edge selection bit		0	O	O
4	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
5	INT_1 source selection bit at STP or WIT	0 : $P3_1/INT_1$ 1 : $P0_0$–$P0_7$ LOW level (for key-on wake-up)	0	O	O
6	Not implemented. Writing to these bits are disabled. These bits are undefined at reading.		Undefined	Undefined	×
7			Undefined	Undefined	×

Note: When setting bits 0 to 3, the interrupt request bit may be set to '1'.
After setting the following, enable the interrupt.
① Disable interrupts
② Set the edge polarity selection register
③ Set the interrupt request bit to '0'

Figure 1.19.4 Edge Polarity Selection Register

1.19.2 Stop Mode

(1) Operations in Stop Mode

State in Stop Mode

When the **STP** instruction is valid, its execution causes the CPU to enter the stop mode. In this mode, the CPU operation is halted because internal clock ϕ stops in the HIGH state. In addition, the operation of the peripherals stops as well, because the oscillation of f(XIN) stops. As a result, power dissipation can be reduced.

Timer 1 goes to 'FF16' to generate the oscillator start-up stabilization time necessary for terminating the stop mode, and a frequency of f(XIN)/8 is selected as the count source.

Note: Timers continue counting in the event count mode, as done the serial I/O does when the external clock (or its 1/16) is selected as the synchronous clock.

For the operations in the stop mode, refer to **Table 1.19.1 States of Microcomputer in Power Saving Modes.**

The stop mode is terminated by reset or accepting an interrupt request, and the CPU returns to the normal mode.

The operation at recovery from the stop mode by reset or accepting an interrupt request is described below.

Recovery from Stop Mode by Reset Input

① By applying the LOW level to the $\overline{\text{RESET}}$ pin for 2 μs or more in the stop mode, the CPU enters the reset state and is brought out of the stop mode, causing the XIN oscillation to resume.
② When the $\overline{\text{RESET}}$ pin is restored to the HIGH level, the oscillator start-up stabilization time is generated by timer 1.
③ After the oscillator start-up stabilization time elapses, internal clock ϕ is supplied to the CPU.
④ The program is executed at the address stored in the reset vector area.

Figure 1.19.5 shows the operation at recovery from the stop mode by reset input.

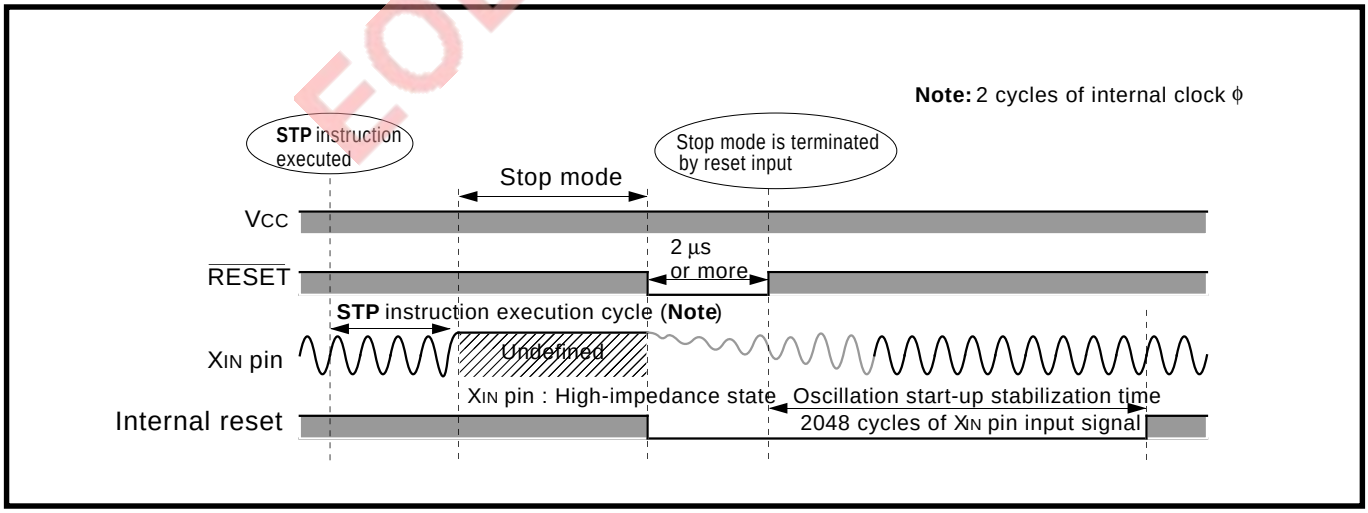


Figure 1.19.5 Operation at Recovery from Stop Mode by Reset Input

For details of reset, refer to **Section 1.17 Reset.**

Recovery from Stop Mode by Interrupt

① The stop mode is terminated and the X_{IN} oscillation is resumed when an interrupt request is generated and its interrupt is acceptable in the stop mode.
② Next, the oscillator start-up stabilization time is generated by timer 1. After the oscillator start-up stabilization time elapses, internal clock ϕ is resumed and supplied to the CPU.
③ The interrupt request used to terminate the stop mode is accepted and the interrupt service routine is executed.
④ After the interrupt service routine is completed, the program is executed at the instruction following the **STP** instruction.

Note: The state of timer 1 is affected by recovering from the stop mode.

The interrupt sources used for recovery from the stop mode are as follows:
- INT_0, INT_1
- $CNTR_0$, $CNTR_1$
- Serial I/O (only when external clock (or its 1/16) is selected as the synchronous clock)
- Timer X and timer Y (only in event count mode)
- Key inputs (in key-on wakeup)

Figure 1.19.6 shows an operation example at recovery from the stop mode by the INT_0 interrupt.

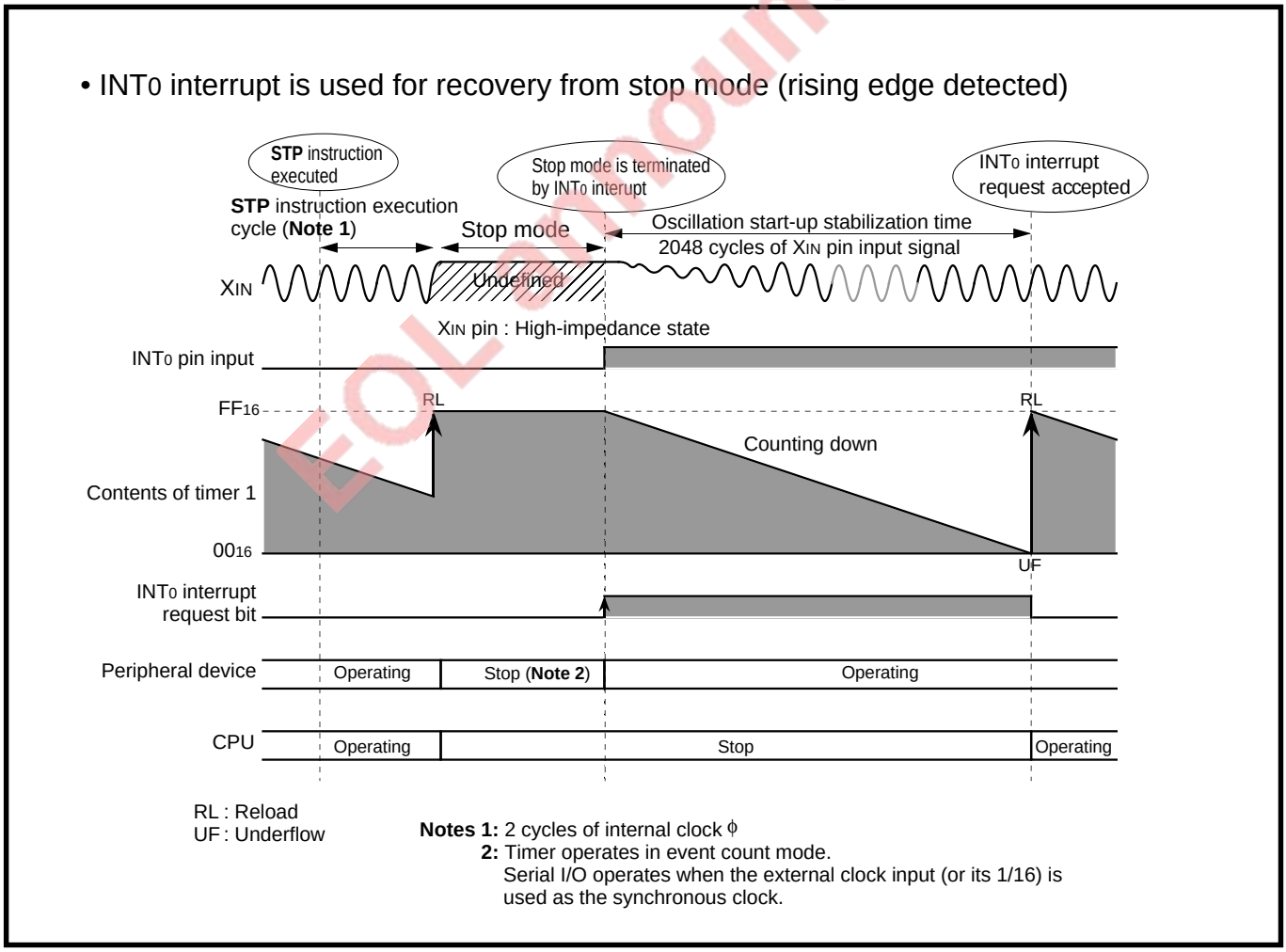


Figure 1.19.6 Operation Example at Recovery from Stop Mode by INT_0 Interrupt

(2) Transition to Stop Mode

The transition from the normal mode to the stop mode is described below.

Recovery from Stop Mode by Reset Input

Execute the **STP** instruction while the **STP** instruction is valid.

Recovery from Stop Mode by Accepting Interrupt Request

Execute the **STP** instruction while the **STP** instruction is valid after the following sequence is completed:

- Set the interrupt that is used to terminate the stop mode.
- Clear the timer 1 interrupt enable bit to '0' (disabled).
- Clear the timer 1 stop control bit to '0' (count operation).

For the setting of the valid/invalid of the **STP** instruction, refer to **Section 1.19.4 Setting of Valid/ Invalid of STP and WIT Instructions.**

1.19.3 Wait Mode

(1) Operations in Wait Mode

State in Wait Mode

When the **WIT** instruction is valid, its execution causes the CPU to enter the wait mode. In this mode, internal clock ϕ stops, though f(XIN) continues oscillation. As a result, the CPU is halted but the peripherals continue to operate.

For the operations in the wait mode, refer to **Table 1.19.1 States of Microcomputer at Power Saving Modes.**

The wait mode is terminated by reset or accepting an interrupt request, and the CPU returns to the normal mode.

The operation at recovery from the wait mode by reset or accepting an interrupt request is described below.

Recovery from Wait Mode by Reset Input

① By applying the LOW level to the $\overline{\text{RESET}}$ pin for 2 μs or more in the wait mode, the CPU enters the reset state and is brought out of the wait mode.
② When the $\overline{\text{RESET}}$ pin is restored to the HIGH level, the oscillator start-up stabilization time is generated by timer 1.
③ After the oscillator start-up stabilization time elapses, internal clock ϕ is supplied to the CPU.
④ The program is executed at the address stored in the reset vector area.

For details of reset, refer to **Section 1.17 Reset**.

Recovery from Wait Mode by Interrupt

① The wait mode is terminated, when an interrupt request is generated and its interrupt is acceptable in the wait mode.
② Next, internal clock ϕ is resumed and supplied to the CPU.
③ The interrupt request used to terminate the wait mode is accepted and the interrupt service routine is executed.
④ After the interrupt service routine is completed, the program is executed at the instruction following the **WIT** instruction.

All interrupt sources except the **BRK** instruction interrupt, are available for recovering from the wait mode.

(2) Transition to Wait Mode

The transition from the normal mode to the wait mode is described below.

Recovery from Wait Mode by Reset Input

Execute the **WIT** instruction while the **WIT** instruction is valid.

Recovery from Wait Mode by Accepting Interrupt Request

Execute the **WIT** instruction while the **WIT** instruction is valid after the interrupt for terminating the wait mode is set.

For the setting of the valid/invalid of the **WIT** instruction, refer to **Section 1.19.4 Setting of Valid/ Invalid of STP and WIT Instructions**.

1.19.4 Setting of Valid/Invalid of STP and WIT Instructions

In the 7480 Group and 7481 Group, the valid/invalid of the **STP** and **WIT** instructions can be selected with the **STP** instruction operation control register. The **STP** and the **WIT** instruction are invalid after the system is released from reset to prevent the program from a runaway.
Writing twice successively to the **STP** instruction operation control register makes the **STP** and the **WIT** instruction valid, while non-successive writing to the register (for example, a single write) makes these instructions invalid. As the **STP** and the **WIT** instruction remain invalid after the system is released from reset, successive writing is used to prevent the clock oscillation from stopping due to erroneous data written during a program runaway.

Figure 1.19.7 shows the setting of valid/invalid of the **STP** and **WIT** instructions.

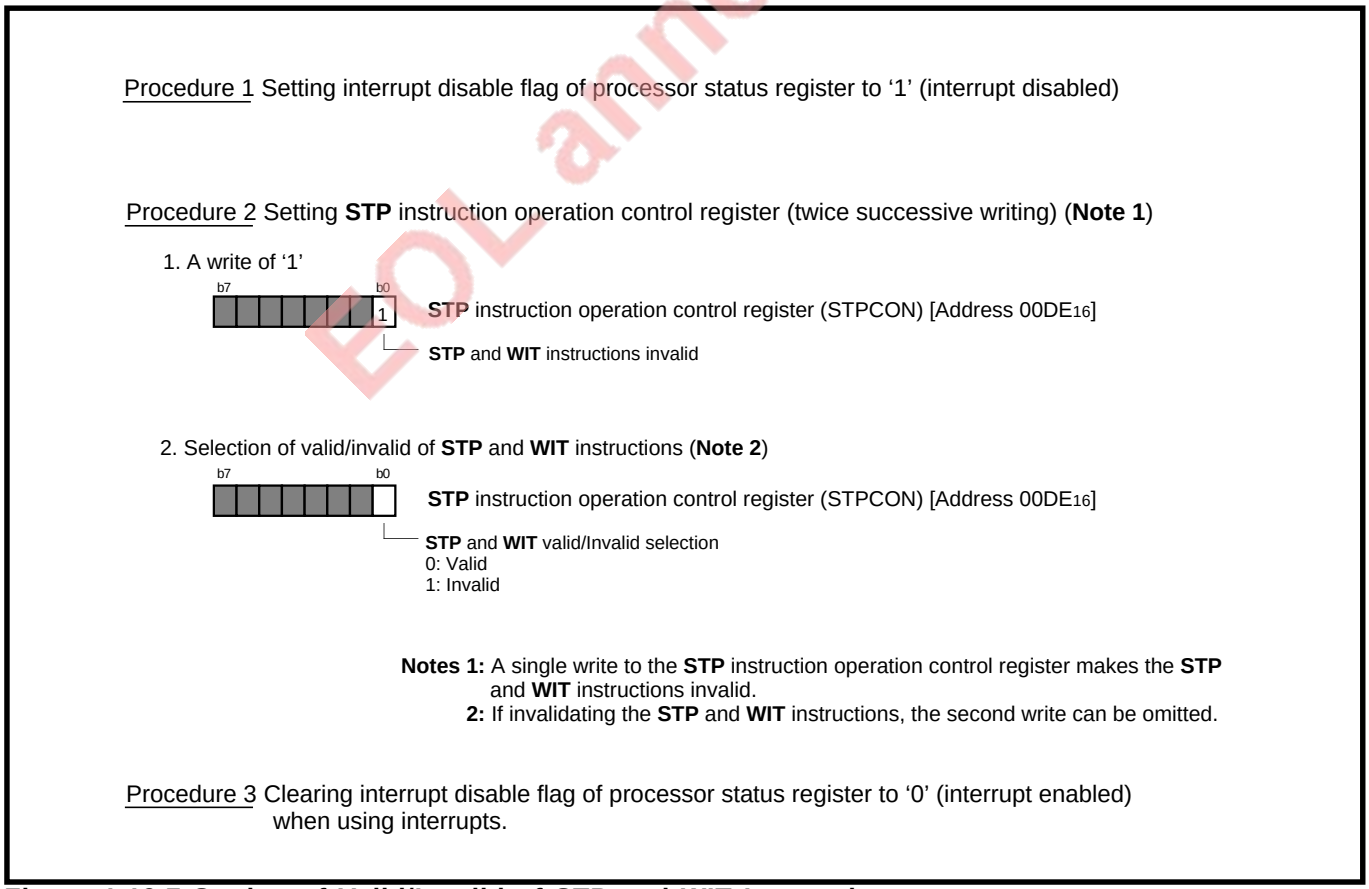


Figure 1.19.7 Setting of Valid/Invalid of STP and WIT Instructions

1.19 Power Saving Function

1.19.5 Notes on Usage

Pay attention to the following notes when the power saving function is used.

(1) Setting of Valid/Invalid of STP and WIT Instructions

To make the **STP** and the **WIT** instruction valid, write twice successively to the **STP** instruction operation control register while interrupts are disabled.

REASON: Execution of an interrupt service routine may cause this register not to be successively written.

(2) In Stop Mode

■ After the CPU is brought out of the stop mode, timer 1 operates in the following conditions. Re-set timer 1 if necessary.

- Contents of the timer 1 latch: 'FF_{16}'
- Count source: f(X_{IN})/8

■ Since the A-D converter stops in the stop mode, execute the **STP** instruction after A-D conversion is completed.

■ In the stop mode, timer X and timer Y continue counting only in the event count mode.

■ Serial I/O operates only when an external clock (or its 1/16) is selected as the synchronous clock in the stop mode.

1.20 Built-in PROM Version

A microcomputer with built-in PROM (PROM) is called a built-in programmable ROM version (built-in PROM version), in contrast to a mask ROM version.
The 7480 Group and 7481 Group offer the following two versions of this type.

- One Time PROM version
  The one time PROM are programmable only once. Erasing and reprogramming are not possible.

- Built-in EPROM version (with a transparent window)
  The microcomputer has a built-in erasable PROM (EPROM) with a transparent window in top of the package. The built-in EPROM are programmable, erasable and reprogrammable.

The built-in PROM version has the EPROM mode to program into the built-in PROM, in addition to the operation modes of the mask ROM version.
For details, refer to **Sections 1.3 Performance Overviews, 1.4 Pinouts,** and **1.6 Functional Block Diagrams.**
The 7480 Group and 7481 Group support the built-in PROM version products listed in Table 1.20.1.

Table 1.20.1 Supported Built-in PROM Version Products in 7480 Group and 7481 Group
(As of September 1997)

Product	PROM (bytes)	RAM (bytes)	I/O Port	Package	Remarks
M37480E8SP	16384	448	I/O ports: 18 Input ports: 8 (Including 4 analog input pins.)	32P4B	One Time PROM Version (shipped in blank)
M37480E8FP				32P2W-A	
M37480E8-XXXSP				32P4B	One Time PROM Version
M37480E8-XXXFP				32P2W-A	
M37480E8T-XXXSP				32P4B	One Time PROM Version (Extended operating temperature range version)
M37480E8T-XXXFP				32P2W-A	
M37481E8SP			I/O ports: 24 Input ports: 12 (Including 8 analog input pins.)	42P4B	One Time PROM Version (shipped in blank)
M37481E8FP				44P6N-A	
M37481E8-XXXSP				42P4B	One Time PROM Version
M37481E8-XXXFP				44P6N-A	
M37481E8T-XXXSP				42P4B	One Time PROM Version (Extended operating temperature range version)
M37481E8T-XXXFP				44P6N-A	
M37481E8SS				42S1B-A	Built-in EPROM Version

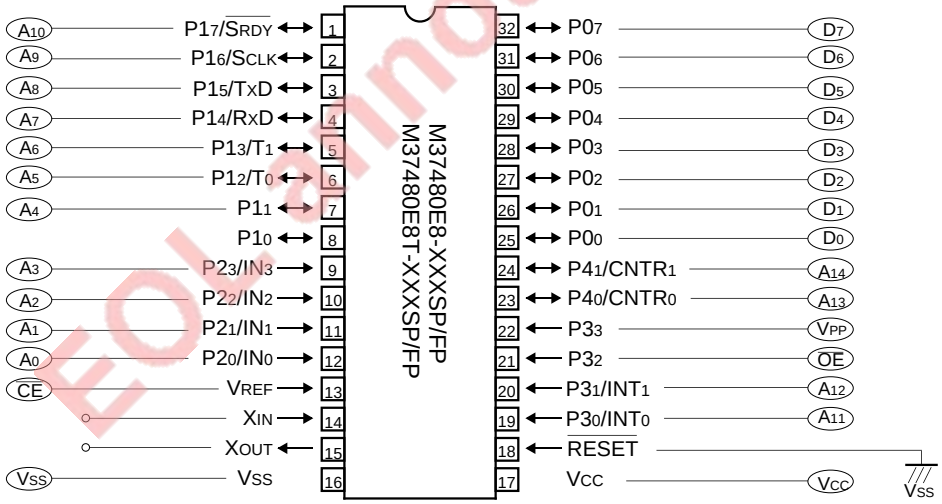
1.20 Built-in PROM Version

1.20.1 EPROM Mode

The built-in PROM version has the EPROM mode in addition to the operation modes of the mask ROM version. The EPROM mode is the mode used to program into and read from the built-in PROM. Programming, reading and erasing of the built-in PROM can be performed by the same operation as in the M5M27C256K. Table 1.20.2 lists the pin functions in the EPROM mode, and Figures 1.20.1 to 1.20.3 show the pinouts.

Table 1.20.2 Pin Functions in EPROM Mode

	Built-in PROM version	M5M27C256K
Pin name	V_{CC}	V_{CC}
	$P3_3$	V_{PP}
	V_{SS}	V_{SS}
	$P1_1$–$P1_7$, $P2_0$–$P2_3$, $P3_0$, $P3_1$, $P4_0$, $P4_1$	A_0–A_{14}
	$P0_0$–$P0_7$	D_0–D_7
	V_{REF}	$\overline{CE}$
	$P3_2$	$\overline{OE}$

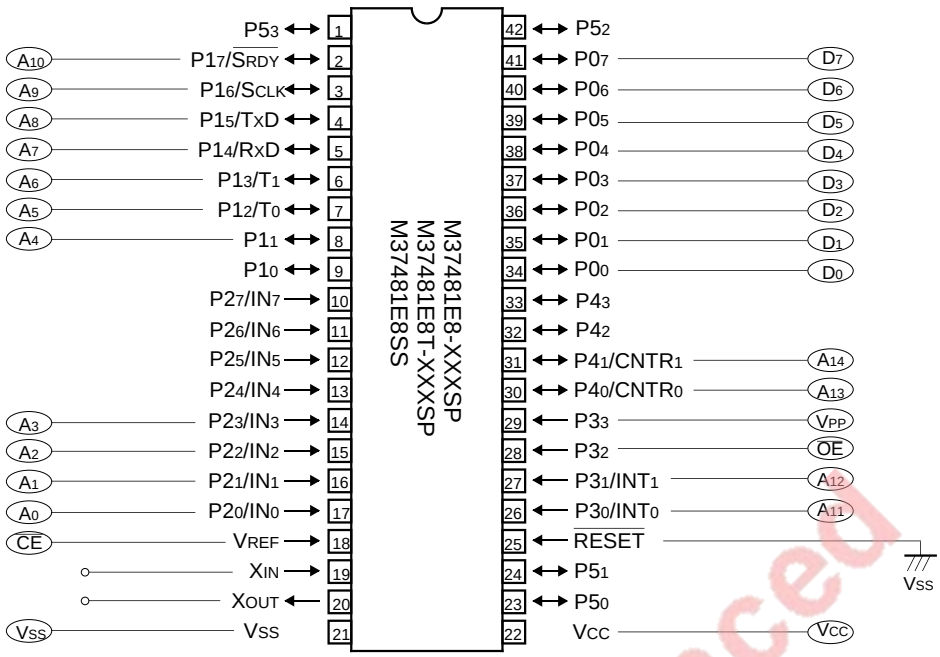


Outline 32P4B
32P2W-A (**Note**)

: PROM pin (Same functions as M5M27C256K)

Note: The only differences between the 32P4B package product and the 32P2W-A package product are package shape and the absolute maximum ratings.

Figure 1.20.1 Pinout in EPROM Mode of 7480 Group



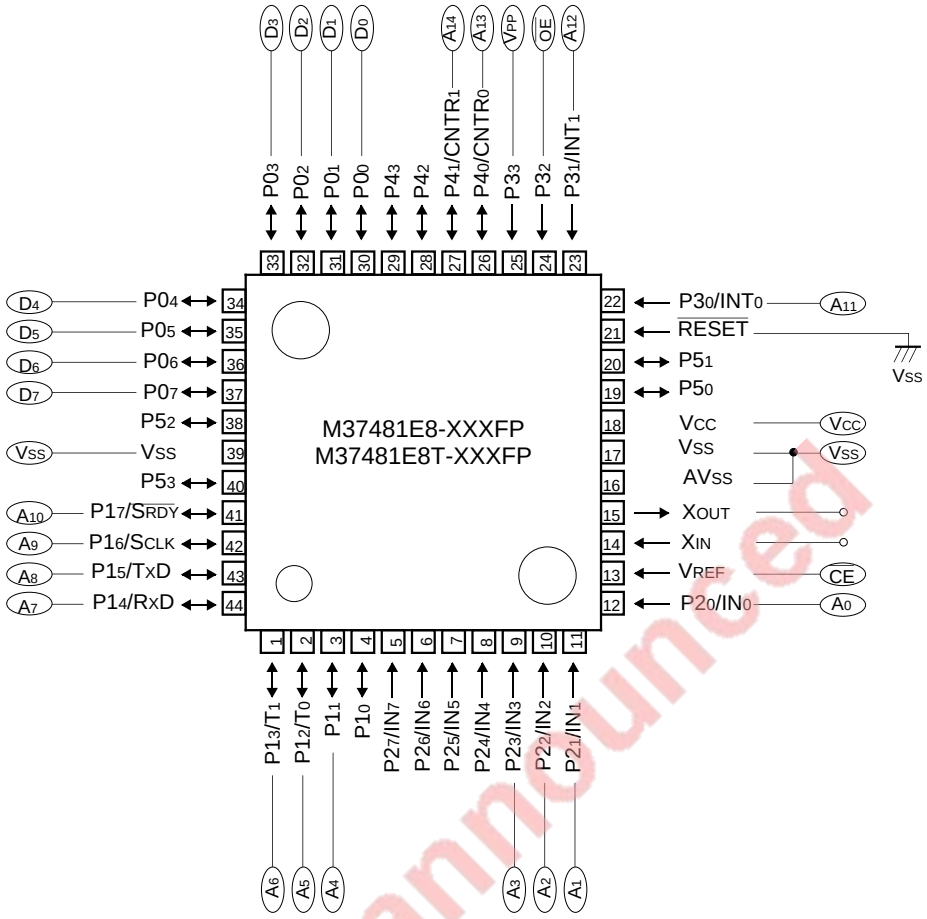
Outline 42P4B (**Note**)
42S1B-A (M37481E8SS)

: PROM pin (Same functions as M5M27C256K)

Note: The only differences between the 42P4B package product and the 44P6N-A package product are package shape, the absolute maximum ratings and the fact that the 44P6N-A package product has the AVSS pin.

Figure 1.20.2 Pinout in EPROM Mode of 7481 Group (1)

1.20 Built-in PROM Version



Outline 44P6N-A (**Note**)

: PROM pin (Same functions as M5M27C256K)

Note: The only differences between the 42P4B package product and the 44P6N-A package product are package shape, the absolute maximum ratings and the fact that the 44P6N-A package product has the AVSS pin.

Figure 1.20.3 Pinout in EPROM Mode of 7481 Group (2)

1.20.2 Pin Descriptions

Tables 1.20.3 and 1.20.4 list pin descriptions in the Ordinary and EPROM modes.

Table 1.20.3 Pin Descriptions (1)

Pin	Mode	Name	Input/ Output	Function
V_{CC}, V_{SS}	Ordinary/ EPROM	Power source		• Apply the following voltage to the V_{CC} pin: 2.7 V to 4.5 V (at f(X_{IN}) = (2.2 V_{CC}–2) MHz), or 4.5 V to 5.5 V (at f(X_{IN}) = 8 MHz). • Apply 0 V to the V_{SS} pin.
AV_{SS}	Ordinary/ EPROM	Analog power source		• Ground level input pin for the A-D converter • Apply the same voltage as for the V_{SS} pin to the AV_{SS} pin. **Note:** This pin is dedicated to the 44P6N-A package products in the 7481 Group.
V_{REF}	Ordinary	Reference voltage input	Input	• Reference voltage input pin for A-D converter • Apply the following voltage to the V_{REF} pin: 2 V to V_{CC} when V_{CC} = 2.7 V to 4.0 V, or 0.5 V_{CC} to V_{CC} when V_{CC} = 4.0 V to 5.5 V. **Note:** When not using A-D converter, connect V_{REF} pin to V_{CC}.
	EPROM	Mode input	Input	• $\overline{CE}$ input pin
$\overline{RESET}$	Ordinary	Reset input	Input	• Reset input pin • System Reset: Holding the LOW level for 2 μs or more forces CPU into reset state.
	EPROM	Reset input	Input	• Connect it to V_{SS} pin.
X_{IN}	Ordinary/ EPROM	Clock input	Input	• I/O pins for clock generator • A ceramic resonator is connected between pins X_{IN} and X_{OUT}. • When an external clock is used, it is input to X_{IN} pin, and leave X_{OUT} pin open. • A feedback resistor is built in between pins X_{IN} and X_{OUT}.
X_{OUT}	Ordinary/ EPROM	Clock output	Output	

1.20 Built-in PROM Version

Table 1.20.4 Pin Descriptions (2)

Pin	Mode	Name	Input/ Output	Function
$P0_0$–$P0_7$	Ordinary	I/O port P0	I/O	• 8-bit I/O port pins • The output structure is CMOS output. • When an input port is selected, a pull-up transistor can be connectable **by the bit**. • In input mode, a key-on wake up function is provided.
	EPROM	Data I/O D_0–D_7	I/O	• Data (D_0–D_7) I/O pins
$P1_0$–$P1_7$	Ordinary	I/O port P1	I/O	• 8-bit I/O port pins • The output structure is CMOS output. • When an input port is selected, a pull-up transistor can be connected **by the 4 bits**. • $P1_2$ and $P1_3$ serve the alternative functions of the timer output pins T_0 and T_1. • $P1_4$, $P1_5$, $P1_6$, and $P1_7$ serve the alternative functions of the serial I/O pins RxD, TxD, SCLK and $\overline{\text{SRDY}}$, respectively.
	EPROM	Address input A_4–A_{10}	Input	• $P1_1$–$P1_7$ are the address (A_4–A_{10}) input pins. • Leave $P1_0$ open.
$P2_0$–$P2_7$	Ordinary	Input port P2	Input	• 8-bit input port pins • $P2_0$–$P2_7$ serve the alternative functions of the analog input pins IN_0–IN_7. **Note:** The 7480 Group has only four pins of $P2_0$–$P2_3$ (IN_0–IN_3).
	EPROM	Address input A_0–A_3	Input	• $P2_0$–$P2_3$ are the address (A_0–A_3) input pins. • Leave $P2_4$–$P2_7$ open.
$P3_0$–$P3_3$	Ordinary	Input port P3	Input	• 4-bit input port pins • $P3_0$ and $P3_1$ serve the alternative functions of the external interrupt input pins INT_0 and INT_1.
	EPROM	Address input A_{11}, A_{12} Mode input V_{PP} input	Input	• $P3_0$, $P3_1$ are the address (A_{11}, A_{12}) input pins. • $P3_2$ pin is the $\overline{\text{OE}}$ input pin. • $P3_3$ pin is the V_{PP} input pin used to apply V_{PP} when programming and program verifying.
$P4_0$–$P4_3$	Ordinary	I/O port P4	I/O	• 4-bit I/O port pins • The output structure is N-channel open-drain outputs with built-in clamping diodes. • $P4_0$ and $P4_1$ serve the alternative functions of the timer I/O pins $CNTR_0$ and $CNTR_1$. **Note:** The 7480 Group has only two pins of $P4_0$ and $P4_1$.
	EPROM	Address input A_{13}, A_{14}	Input	• $P4_0$, $P4_1$ are the address (A_{13}, A_{14}) input pins. • Leave $P4_2$, $P4_3$ open.
$P5_0$–$P5_3$	Ordinary	I/O port P5	I/O	• 4-bit I/O port pins • The output structure is N-channel open-drain outputs with built-in clamping diodes. **Note:** The 7480 Group is not provided with port P5.
	EPROM	Input port P5	Input	• Leave port P5 open.

1.20.3 Reading, Programming and Erasing of Built-in PROM

The built-in PROM version can be used in the EPROM mode by setting the $\overline{\text{RESET}}$ pin to LOW. Reading, programming, and erasing of the built-in PROM in the EPROM mode are described below.
Also, Table 1.20.5 lists the I/O signals in the EPROM mode.

(1) Reading from Built-in PROM

- 0 V is applied to the $\overline{\text{RESET}}$ pin, and 5 V to the V_{CC} pin.
- Address signals (A_0–A_{14}) are input, and the $\overline{\text{OE}}$ and the $\overline{\text{CE}}$ pins are set to LOW. Then, the contents of PROM are placed on data I/O pins (D_0–D_7).
- The $\overline{\text{CE}}$ or the $\overline{\text{OE}}$ pins are set to HIGH. Then, data I/O pins (D_0–D_7) float.

(2) Programming into Built-in PROM

- 0 V is applied to the $\overline{\text{RESET}}$ pin, and 5 V to the V_{CC} pin.
- The $\overline{\text{OE}}$ pin is set to HIGH and V_{PP} is applied to the V_{PP} pin. Then, the CPU enters the program mode.
- Addresses are set to address input pins (A_0–A_{14}), and the 8-bit data to be programmed is placed in parallel, on data I/O pins (D_0–D_7).
- Setting the $\overline{\text{CE}}$ pin to LOW starts programming.

Specify addresses 4000_{16} through $7FFF_{16}$ when programming with the PROM programmer.
Also, set all addresses 0000_{16} through $3FFF_{16}$ to 'FF_{16}' when programming into addresses 0000_{16} through $7FFF_{16}$.

(3) Erasing

- Only the built-in EPROM version with a window (M37481E8SS) is erasable.
- The EPROM can be erased when exposed to ultraviolet light with a wavelength of 2537 Å.
- Integrated dose necessary for erasure is a minimum of 15 W•s/cm^2.

Table 1.20.5 I/O Signals in EPROM Mode

Mode \ Pin name	$\overline{\text{CE}}$	$\overline{\text{OE}}$	V_{PP}	V_{CC}	$\overline{\text{RESET}}$	D_0–D_7
Read	V_{IL}	V_{IL}	V_{CC}	V_{CC}	0 V	Output
Output disable	V_{IL}	V_{IH}	V_{CC}			Floating
Write	V_{IL}	V_{IH}	V_{PP}			Input
Write-verify	V_{IH}	V_{IL}	V_{PP}			Output
Write disable	V_{IH}	V_{IH}	V_{PP}			Floating

Note: V_{IL} represents the LOW input voltage, and V_{IH}, the HIGH input voltage.

1.20.4 Notes on Usage

Pay attention to the following notes when the built-in PROM version is used.

(1) All Products of Built-in PROM Version Products

■ **Programming into Built-in PROM**

- A high voltage is used to program into the PROM. Be careful not to apply an overvoltage to pins, especially when power is turned on.

- The use of a dedicated programming adapter (**Note**) is recommended when the PROM programming is performed, so that general-purpose PROM programmers are available for programming.

■ **Reading from Built-in PROM**

The use of a dedicated programming adapter (**Note**) is recommended when the PROM contents are read, so that general-purpose PROM programmers are available for reading.

Note: Refer to Data Book **DEVELOPMENT SUPPORT TOOLS FOR MICROCOMPUTERS** for the dedicated programming adapter.

(2) One Time PROM Version

The one time PROM version (a blank product) is neither tested nor screened since Mitsubishi's assembly process. To improve reliability after programming, it is suggested that these products are used only after programming and verification, according to the procedure shown in Figure 1.20.4, is completed.

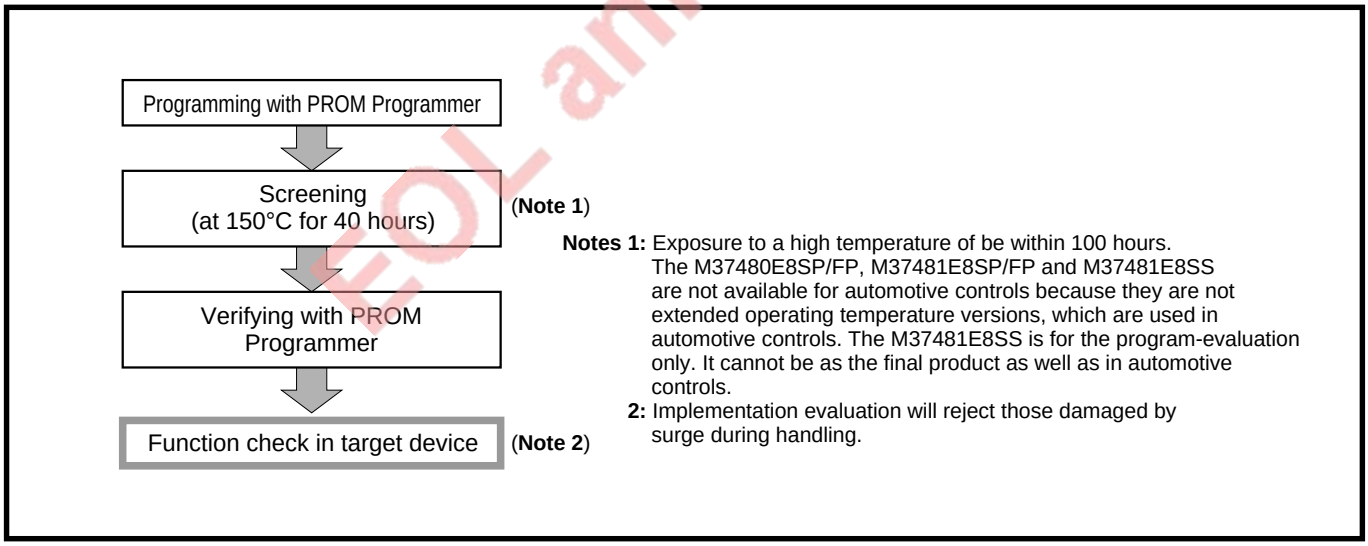


Figure 1.20.4 Programming and Verification of One Time PROM Version

(3) Built-in EPROM Version

- The built-in EPROM version can be used for program development only. Use them only for program development and implementation evaluation.

- Sunlight and fluorescent light include light that may erase the information programmed in the built-in PROM. When using the EPROM version in the read mode, be sure to cover the transparent glass portion with a seal.

- This seal to cover the transparent glass portion is prepared by Mitsubishi. Be careful not to bring the seal into contact with the microcomputer lead wires when covering the portion with the seal because this seal is made of metal (aluminum).

- Before erasing data, clean the transparent glass. If any finger stain or seal adhesive is stuck to the transparent glass, this prevents ultraviolet rays from passing, thereby affecting the erase characteristic adversely.

1.21 Electrical Characteristics

1.21.1 Electrical Characteristics

(1) Electrical Characteristics of 7480 Group

For the 7480 Group, Table 1.21.1 lists the absolute maximum ratings, and Tables 1.21.2 and 1.21.3 list the recommended operating conditions. Also, Tables 1.21.4 and 1.21.5 list the electrical characteristics, and Table 1.21.6 lists the A-D conversion characteristics.

Table 1.21.1 Absolute Maximum Ratings of 7480 Group

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltage	All voltages are measured based on V_{SS}. Output transistors are in the cut-off state.	–0.3 to 7	V
V_I	Input voltage		–0.3 to V_{CC}+0.3	V
V_O	Output voltage		–0.3 to V_{CC}+0.3	V
Pd	Power dissipation	Ta = 25°C	1000 (**Note 1**)	mW
Topr	Operating temperature range		–20 to 85 (**Note 2**)	°C
Tstg	Storage temperature range		–40 to 150 (**Note 3**)	°C

Notes 1: 500 mW for 32P2W-A package.
2: –40 °C to 85 °C for extended operating temperature range version.
3: –65 °C to 150 °C for extended operating temperature range version.

Table 1.21.2 Recommended Operating Conditions of 7480 Group (1) (Note 1)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage	f(X_{IN}) = 8 MHZ	4.5	5	5.5	V
		f(X_{IN}) = (2.2V_{CC}–2.0) MHz	2.7	3	4.5	V
V_{SS}	Power source voltage			0		V
V_{IH}	HIGH input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$		0.8V_{CC}		V_{CC}	V
V_{IH}	HIGH input voltage $P2_0$–$P2_3$		0.7V_{CC}		V_{CC}	V
V_{IH}	HIGH input voltage $P3_0$–$P3_3$, $P4_0$, $P4_1$ (**Note 2**)	V_{CC} = 4.5 V to 5.5V	0.8V_{CC}		V_{CC}	V
		V_{CC} = 2.7 V to 4.5V	0.9V_{CC}		V_{CC}	V
V_{IH}	HIGH input voltage X_{IN}, $\overline{RESET}$		0.8V_{CC}		V_{CC}	V
V_{IL}	LOW input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$		0		0.2V_{CC}	V
V_{IL}	LOW input voltage $P2_0$–$P2_3$		0		0.25V_{CC}	V
V_{IL}	LOW input voltage $P3_0$–$P3_3$, $P4_0$, $P4_1$	V_{CC} = 4.5 V to 5.5V	0		0.4V_{CC}	V
		V_{CC} = 2.7 V to 4.5V	0		0.3V_{CC}	V
V_{IL}	LOW input voltage X_{IN}		0		0.16V_{CC}	V
V_{IL}	LOW input voltage $\overline{RESET}$		0		0.12V_{CC}	V
I_I	Input current $P4_0$, $P4_1$ (**Note 2**) V_I > V_{CC}				1	mA

Notes 1: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85 °C (Ta = –40 ° C to 85 °C for extended operating temperature range version), unless otherwise noted.
For the clamping diodes of port P4, refer to **(4) Level Shift Ports** in **Section 1.10.3 I/O Ports**.
2: When voltage is applied through a resistor, current I of 1 mA or less maintains V_I > V_{CC}.
For this circuit, refer to **Figure 1.10.11 Port P4 and P5 Circuit**.

Table 1.21.3 Recommended Operating Conditions of 7480 Group (2) (Note 1)

Symbol	Parameter			Limits			Unit
				Min.	Typ.	Max.	
I_{OH}(sum)	HIGH output sum current $P0_0$–$P0_7$					–30	mA
I_{OH}(sum)	HIGH output sum current $P1_0$–$P1_7$					–30	mA
I_{OL}(sum)	LOW output sum current $P0_0$–$P0_7$, $P4_0$, $P4_1$					60	mA
I_{OL}(sum)	LOW output sum current $P1_0$–$P1_7$					60	mA
I_{OH}(peak)	HIGH output peak current $P0_0$–$P0_7$, $P1_0$–$P1_7$					–10	mA
I_{OL}(peak)	LOW output peak current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P4_0$, $P4_1$					20	mA
I_{OH}(avg)	HIGH output average current $P0_0$–$P0_7$, $P1_0$–$P1_7$ (**Note 2**)					–5	mA
I_{OL}(avg)	LOW output average current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P4_0$, $P4_1$ (**Note 2**)					10	mA
f(CNTR)	Timer input frequency $CNTR_0$ ($P4_0$)		f(X_{IN}) = 8 MHz			2	MHz
	$CNTR_1$ ($P4_1$) (**Note 3**)		f(X_{IN}) = 4 MHz			1	MHz
f(S_{CLK})	Serial I/O clock input frequency S_{CLK} ($P1_6$) (**Note 3**)	when selecting clock synchronous serial I/O	f(X_{IN}) = 8 MHz			500	kHz
			f(X_{IN}) = 4 MHz			250	kHz
		when selecting UART	f(X_{IN}) = 8 MHz			2	MHz
			f(X_{IN}) = 4 MHz			1	MHz
f(X_{IN})	Clock input oscillation frequency (**Note 3**)	V_{CC} = 4.5V to 5.5V				8	MHz
		V_{CC} = 2.7V to 4.5V				2.2V_{CC}–2.0	MHz

Notes 1: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85 °C (Ta = –40 ° C to 85 °C for extended operating temperature range version), unless otherwise noted.
For the clamping diodes of port P4, refer to **(4) Level Shift Ports** in **Section 1.10.3 I/O Ports**.

2: Output average currents I_{OH}(avg) and I_{OL}(avg) are average values for a period of 100 ms.

3: The frequency is the value at a 50% duty cycle.

4: Connect a bypass capacitor of capacity 0.1 μF between V_{CC} and V_{SS}, and one of capacity 0.01μF between V_{REF} and V_{SS}.

1.21 Electrical Characteristics

Table 1.21.4 Electrical Characteristics of 7480 Group (1) (Note 1)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	V_{CC} = 5 V, I_{OH} = –5 mA		3			V
	$P0_0$–$P0_7$, $P1_0$–$P1_7$	V_{CC} = 3 V, I_{OH} = –1.5 mA		2			V
V_{OL}	LOW output voltage	V_{CC} = 5 V, I_{OL} = 10 mA				2	V
	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P4_0$, $P4_1$	V_{CC} = 3 V, I_{OL} = 3 mA				1	V
V_{T+}–V_{T-}	Hysteresis $P0_0$–$P0_7$ (**Note 2**),	V_{CC} = 5 V			0.5		V
	$P3_0$–$P3_3$, $P4_0$, $P4_1$	V_{CC} = 3 V			0.3		V
V_{T+}–V_{T-}	Hysteresis $\overline{\text{RESET}}$	V_{CC} = 5 V			0.5		V
		V_{CC} = 3 V			0.3		V
V_{T+}–V_{T-}	Hysteresis	when used as	V_{CC} = 5 V		0.5		V
	$P1_4$/RxD, $P1_6$/S_{CLK}	RxD, S_{CLK}	V_{CC} = 3 V		0.3		V
I_{IH}	HIGH input current	V_I = V_{CC},	V_{CC} = 5 V			5	μA
	$P0_0$–$P0_7$, $P1_0$–$P1_7$	No pull-up transistor	V_{CC} = 3 V			3	μA
I_{IH}	HIGH input current	V_I = V_{CC} = 5 V				5	μA
	$P3_0$–$P3_3$, $P4_0$, $P4_1$	V_I = V_{CC} = 3 V				3	μA
I_{IH}	HIGH input current	V_I = V_{CC}, when not	V_{CC} = 5 V			5	μA
	$P2_0$–$P2_3$	selecting analog input	V_{CC} = 3 V			3	μA
I_{IH}	HIGH input current	V_I = V_{CC}, when X_{IN}	V_{CC} = 5 V			5	μA
	X_{IN}, $\overline{\text{RESET}}$	is stopped	V_{CC} = 3 V			3	μA
I_{IL}	LOW input current	V_I = 0 V,	V_{CC} = 5 V			–5	μA
	$P0_0$–$P0_7$, $P1_0$–$P1_7$	No pull-up transistor	V_{CC} = 3 V			–3	μA
		V_I = 0 V (**Note 3**),	V_{CC} = 5 V	–0.25	–0.5	–1.0	mA
		Pull-up transistor used	V_{CC} = 3 V	–0.08	–0.18	–0.35	mA
I_{IL}	LOW input current	V_I = 0 V	V_{CC} = 5 V			–5	μA
	$P3_0$–$P3_3$, $P4_0$, $P4_1$		V_{CC} = 3 V			–3	μA
I_{IL}	LOW input current	V_I = 0 V, when not	V_{CC} = 5 V			–5	μA
	$P2_0$–$P2_3$	selecting analog input	V_{CC} = 3 V			–3	μA
I_{IL}	LOW input current	V_I = 0 V, when X_{IN}	V_{CC} = 5 V			–5	μA
	X_{IN}, $\overline{\text{RESET}}$	is stopped	V_{CC} = 3 V			–3	μA

Notes 1: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85°C (Ta = –40 °C to 85°C for extended operating temperature range version), unless otherwise noted.

2: The limits when the key-on wakeup function of port P0 is used

3: When represented with electric resistance, the corresponding values are as follows:

- V_{CC} = 5 V: 5 kΩ (Min.), 10 kΩ (Typ.), and 20 kΩ (Max.)
- V_{CC} = 3 V: 8.6 kΩ (Min.), 16.7 kΩ (Typ.), and 37.5 kΩ (Max.)

Table 1.21.5 Electrical Characteristics of 7480 Group (2) (Note)

Symbol	Parameter	Test conditions			Limits			Unit
					Min.	Typ.	Max.	
I_{CC}	Power source current	At system operating	High-speed mode, $f(X_{IN})$ = 4 MHz, V_{CC} = 5 V	No A-D conversion		3.5	7	mA
				During A-D conversion		4	8	mA
			High-speed mode, $f(X_{IN})$ = 4 MHz, V_{CC} = 3 V	No A-D conversion		1.8	3.6	mA
				During A-D conversion		2	4	mA
			High-speed mode, $f(X_{IN})$ = 8 MHz, V_{CC} = 5 V	No A-D conversion		7	14	mA
				During A-D conversion		7.5	15	mA
			Medium-speed mode, $f(X_{IN})$ = 4 MHz, V_{CC} = 5 V	No A-D conversion		1.75	3.5	mA
				During A-D conversion		2	4	mA
			Medium-speed mode, $f(X_{IN})$ = 4 MHz, V_{CC} = 3 V	No A-D conversion		0.9	1.8	mA
				During A-D conversion		1	2	mA
			Medium-speed mode, $f(X_{IN})$ = 8 MHz, V_{CC} = 5 V	No A-D conversion		3.5	7	mA
				During A-D conversion		3.75	7.5	mA
		At wait	High-speed mode, $f(X_{IN})$ = 4 MHz	V_{CC} = 5 V		1	2	mA
				V_{CC} = 3 V		0.5	1	mA
			High-speed mode, $f(X_{IN})$ = 8 MHz	V_{CC} = 5 V		2	4	mA
			Medium-speed mode, $f(X_{IN})$ = 4 MHz	V_{CC} = 5 V		0.9	1.8	mA
				V_{CC} = 3 V		0.45	0.9	mA
			Medium-speed mode, $f(X_{IN})$ = 8 MHz	V_{CC} = 5 V		1.8	3.6	mA
		At stop	$f(X_{IN})$ = 0 MHz, V_{CC} = 5 V	Ta = 25 °C		0.1	1	μA
				Ta = 85 °C		1	10	μA
V_{RAM}	RAM back-up voltage	At clock stop			2.0			V

Note: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85°C (Ta = –40 °C to 85°C for extended operating temperature range version), unless otherwise noted.

1.21 Electrical Characteristics

Table 1.21.6 A-D Conversion Characteristics of 7480 Group (Note)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
–	Resolution				8	bits
–	Absolute accuracy (except quantification error)	V_{CC} = V_{REF} = 5.0 V			±2	LSB
T_{CONV}	Conversion time	V_{CC} = 2.7V to 4.5V, f(X_{IN}) = 4MHz			25	μs
		V_{CC} = 4.5V to 5.5V, f(X_{IN}) = 8MHz			12.5	μs
V_{VREF}	Reference voltage	V_{CC} = 2.7 V to 4.0V	2		V_{CC}	V
		V_{CC} = 4.0 V to 5.5 V	0.5 V_{CC}		V_{CC}	V
R_{LADDER}	Ladder resistor		12	35	100	kΩ
V_{IA}	Analog input voltage		0		V_{REF}	V
I_{VREF}	Reference power input current	V_{REF} = 5.0 V	50	143	416	μA

Note: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85°C (Ta = –40 °C to 85°C for extended operating temperature range version), unless otherwise noted.

(2) Electrical Characteristics of 7481 Group

For the 7481 Group, Table 1.21.7 lists the absolute maximum ratings, and Tables 1.21.8 and 1.21.9 list the recommended operating conditions. Also, Tables 1.21.10 and 1.21.11 list the electrical characteristics, and Table 1.21.12 lists the A-D conversion characteristics.

Table 1.21.7 Absolute Maximum Ratings of 7481 Group

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltage	All voltages are measured based on	–0.3 to 7	V
V_I	Input voltage	V_{SS}.	–0.3 to V_{CC}+0.3	V
V_O	Output voltage	Output transistors are in the cut-off state.	–0.3 to V_{CC}+0.3	V
Pd	Power dissipation	Ta = 25°C	1000 (**Note 1**)	mW
Topr	Operating temperature range		–20 to 85 (**Note 2**)	°C
Tstg	Storage temperature range		–40 to 150 (**Note 3**)	°C

Notes 1: 500 mW for 44P6N-A package.
2: –40 °C to 85 °C for extended operating temperature range version.
3: –65 °C to 150 °C for extended operating temperature range version.

Table 1.21.8 Recommended Operating Conditions of 7481 Group (1) (Note 1)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage	f(X_{IN}) = 8 MHZ	4.5	5	5.5	V
		f(X_{IN}) = (2.2V_{CC}–2.0) MHz	2.7	3	4.5	V
V_{SS}	Power source voltage			0		V
V_{IH}	HIGH input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$		0.8V_{CC}		V_{CC}	V
V_{IH}	HIGH input voltage $P2_0$–$P2_7$		0.7V_{CC}		V_{CC}	V
V_{IH}	HIGH input voltage	V_{CC} = 4.5 V to 5.5V	0.8V_{CC}		V_{CC}	V
	$P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_3$ (**Note 2**)	V_{CC} = 2.7 V to 4.5V	0.9V_{CC}		V_{CC}	V
V_{IH}	HIGH input voltage X_{IN}, $\overline{RESET}$		0.8V_{CC}		V_{CC}	V
V_{IL}	LOW input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$		0		0.2V_{CC}	V
V_{IL}	LOW input voltage $P2_0$–$P2_7$		0		0.25V_{CC}	V
V_{IL}	LOW input voltage	V_{CC} = 4.5 V to 5.5V	0		0.4V_{CC}	V
	$P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_3$ (**Note 2**)	V_{CC} = 2.7 V to 4.5V	0		0.3V_{CC}	V
V_{IL}	LOW input voltage X_{IN}		0		0.16V_{CC}	V
V_{IL}	LOW input voltage $\overline{RESET}$		0		0.12V_{CC}	V
I_I	Input current $P4_0$–$P4_3$, $P5_0$–$P5_3$ (**Note 2**) V_I > V_{CC}				1	mA

Notes 1: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85 °C (Ta = –40 ° C to 85 °C for extended operating temperature range version), unless otherwise noted.
For the clamping diodes of port P4, refer to **(4) Level Shift Ports** in **Section 1.10.3 I/O Ports**.
2: When voltage is applied through a resistor, current I of 1 mA or less maintains V_I > V_{CC}.
For this circuit, refer to **Figure 1.10.11 Port P4/P5 Circuitry**.

1.21 Electrical Characteristics

Table 1.21.9 Recommended Operating Conditions of 7481 Group (2) (Note 1)

Symbol	Parameter			Limits			Unit
				Min.	Typ.	Max.	
I_{OH}(sum)	HIGH output sum current $P0_0$–$P0_7$					–30	mA
I_{OH}(sum)	HIGH output sum current $P1_0$–$P1_7$					–30	mA
I_{OL}(sum)	LOW output sum current $P0_0$–$P0_7$, $P4_0$–$P4_3$, $P5_0$–$P5_2$					60	mA
I_{OL}(sum)	LOW output sum current $P1_0$–$P1_7$, $P5_3$					60	mA
I_{OH}(peak)	HIGH output peak current $P0_0$–$P0_7$, $P1_0$–$P1_7$					–10	mA
I_{OL}(peak)	LOW output peak current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P4_0$–$P4_3$, $P5_0$–$P5_3$					20	mA
I_{OH}(avg)	HIGH output average current $P0_0$–$P0_7$, $P1_0$–$P1_7$ (**Note 2**)					–5	mA
I_{OL}(avg)	LOW output average current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P4_0$–$P4_3$, $P5_0$–$P5_3$ (**Note 2**)					10	mA
f(CNTR)	Timer input frequency $CNTR_0$ ($P4_0$)		f(X_{IN}) = 8 MHz			2	MHz
	$CNTR_1$ ($P4_1$) (**Note 3**)		f(X_{IN}) = 4 MHz			1	MHz
f(S_{CLK})	Serial I/O clock input frequency S_{CLK} ($P1_6$) (**Note 3**)	when selecting clock synchronous serial I/O	f(X_{IN}) = 8 MHz			500	kHz
			f(X_{IN}) = 4 MHz			250	kHz
		when selecting UART	f(X_{IN}) = 8 MHz			2	MHz
			f(X_{IN}) = 4 MHz			1	MHz
f(X_{IN})	Clock input oscillation frequency (**Note 3**)	V_{CC} = 4.5V to 5.5V				8	MHz
		V_{CC} = 2.7V to 4.5V				2.2V_{CC}–2.0	MHz

Notes 1: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85 °C (Ta = –40 ° C to 85 °C for extended operating temperature range version), unless otherwise noted.
For the clamping diodes of port P4, refer to **(4) Level Shift Ports** in **Section 1.10.3 I/O Ports**.

2: Output average currents I_{OH}(avg) and I_{OL}(avg) are average values for a period of 100 ms.

3: The frequency is the value at a 50% duty cycle.

4: Connect a bypass capacitor of capacity 0.1 μF between V_{CC} and V_{SS}, and one of capacity 0.01μF between V_{REF} and V_{SS}.

Table 1.21.10 Electrical Characteristics of 7481 Group (1) (Note 1)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	V_{CC} = 5 V, I_{OH} = –5 mA		3			V
	$P0_0$–$P0_7$, $P1_0$–$P1_7$	V_{CC} = 3 V, I_{OH} = –1.5 mA		2			V
V_{OL}	LOW output voltage $P0_0$–$P0_7$,	V_{CC} = 5 V, I_{OL} = 10 mA				2	V
	$P1_0$–$P1_7$, $P4_0$–$P4_3$, $P5_0$–$P5_3$	V_{CC} = 3 V, I_{OL} = 3 mA				1	V
V_{T+}–V_{T-}	Hysteresis $P0_0$–$P0_7$ (**Note 2**),	V_{CC} = 5 V			0.5		V
	$P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_3$	V_{CC} = 3 V			0.3		V
V_{T+}–V_{T-}	Hysteresis $\overline{\text{RESET}}$	V_{CC} = 5 V			0.5		V
		V_{CC} = 3 V			0.3		V
V_{T+}–V_{T-}	Hysteresis	when used as	V_{CC} = 5 V		0.5		V
	$P1_4$/RxD, $P1_6$/S_{CLK}	RxD, S_{CLK}	V_{CC} = 3 V		0.3		V
I_{IH}	HIGH input current	V_I = V_{CC},	V_{CC} = 5 V			5	μA
	$P0_0$–$P0_7$, $P1_0$–$P1_7$	No pull-up transistor	V_{CC} = 3 V			3	μA
I_{IH}	HIGH input current	V_I = V_{CC} = 5 V				5	μA
	$P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_3$	V_I = V_{CC} = 3 V				3	μA
I_{IH}	HIGH input current	V_I = V_{CC}, when not	V_{CC} = 5 V			5	μA
	$P2_0$–$P2_7$	selecting analog input	V_{CC} = 3 V			3	μA
I_{IH}	HIGH input current	V_I = V_{CC}, when X_{IN}	V_{CC} = 5 V			5	μA
	X_{IN}, $\overline{\text{RESET}}$	is stopped	V_{CC} = 3 V			3	μA
I_{IL}	LOW input current	V_I = 0 V,	V_{CC} = 5 V			–5	μA
	$P0_0$–$P0_7$, $P1_0$–$P1_7$	No pull-up transistor	V_{CC} = 3 V			–3	μA
		V_I = 0 V (**Note 3**),	V_{CC} = 5 V	–0.25	–0.5	–1.0	mA
		Pull-up transistor used	V_{CC} = 3 V	–0.08	–0.18	–0.35	mA
I_{IL}	LOW input current	V_I = 0 V	V_{CC} = 5 V			–5	μA
	$P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_3$		V_{CC} = 3 V			–3	μA
I_{IL}	LOW input current	V_I = 0 V, when not	V_{CC} = 5 V			–5	μA
	$P2_0$–$P2_7$	selecting analog input	V_{CC} = 3 V			–3	μA
I_{IL}	LOW input current	V_I = 0 V, when X_{IN}	V_{CC} = 5 V			–5	μA
	X_{IN}, $\overline{\text{RESET}}$	is stopped	V_{CC} = 3 V			–3	μA

Notes 1: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85°C (Ta = –40 °C to 85°C for extended operating temperature range version), unless otherwise noted.

2: The limits when the key-on wakeup function of port P0 is used

3: When represented with electric resistance, the corresponding values are as follows:

- V_{CC} = 5 V: 5 kΩ (Min.), 10 kΩ (Typ.), and 20 kΩ (Max.)
- V_{CC} = 3 V: 8.6 kΩ (Min.), 16.7 kΩ (Typ.), and 37.5 kΩ (Max.)

1.21 Electrical Characteristics

Table 1.21.11 Electrical Characteristics of 7481 Group (2) (Note)

Symbol	Parameter	Test conditions			Limits			Unit
					Min.	Typ.	Max.	
I_{CC}	Power source current	At system operating	High-speed mode $f(X_{IN})$ = 4 MHz V_{CC} = 5 V	No A-D conversion		3.5	7	mA
				During A-D conversion		4	8	mA
			High-speed mode $f(X_{IN})$ = 4 MHz V_{CC} = 3 V	No A-D conversion		1.8	3.6	mA
				During A-D conversion		2	4	mA
			High-speed mode $f(X_{IN})$ = 8 MHz V_{CC} = 5 V	No A-D conversion		7	14	mA
				During A-D conversion		7.5	15	mA
			Medium-speed mode $f(X_{IN})$ = 4 MHz V_{CC} = 5 V	No A-D conversion		1.75	3.5	mA
				During A-D conversion		2	4	mA
			Medium-speed mode $f(X_{IN})$ = 4 MHz V_{CC} = 3 V	No A-D conversion		0.9	1.8	mA
				During A-D conversion		1	2	mA
			Medium-speed mode $f(X_{IN})$ = 8 MHz V_{CC} = 5 V	No A-D conversion		3.5	7	mA
				During A-D conversion		3.75	7.5	mA
		At wait	High-speed mode $f(X_{IN})$ = 4 MHz	V_{CC} = 5 V		1	2	mA
				V_{CC} = 3 V		0.5	1	mA
			High-speed mode $f(X_{IN})$ = 8 MHz	V_{CC} = 5 V		2	4	mA
			Medium-speed mode $f(X_{IN})$ = 4 MHz	V_{CC} = 5 V		0.9	1.8	mA
				V_{CC} = 3 V		0.45	0.9	mA
			Medium-speed mode $f(X_{IN})$ = 8 MHz	V_{CC} = 5 V		1.8	3.6	mA
		At stop	$f(X_{IN})$ = 0 MHz V_{CC} = 5 V	Ta = 25 °C		0.1	1	μA
				Ta = 85 °C		1	10	μA
V_{RAM}	RAM back-up voltage	At clock stop			2.0			V

Note: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85°C (Ta = –40 °C to 85°C for extended operating temperature range version), unless otherwise noted.

Table 1.21.12 A-D Conversion Characteristics of 7481 Group (Note)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
–	Resolution				8	bits
–	Absolute accuracy (except quantification error)	V_{CC} = V_{REF} = 5.0 V			±2	LSB
T_{CONV}	Conversion time	V_{CC} = 2.7V to 4.5V, f(X_{IN}) = 4MHz			25	μs
		V_{CC} = 4.5V to 5.5V, f(X_{IN}) = 8MHz			12.5	μs
V_{VREF}	Reference voltage	V_{CC} = 2.7 V to 4.0V	2		V_{CC}	V
		V_{CC} = 4.0 V to 5.5 V	0.5 V_{CC}		V_{CC}	V
R_{LADDER}	Ladder resistor		12	35	100	kΩ
V_{IA}	Analog input voltage		0		V_{REF}	V
I_{VREF}	Reference power input current	V_{REF} = 5.0 V	50	143	416	μA

Note: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, and Ta = –20 °C to 85°C (Ta = –40 °C to 85°C for extended temperature range version), unless otherwise noted.

1.21 Electrical Characteristics

1.21.2 Necessary Conditions for Timing and Switching Characteristics

Table 1.21.13 lists the necessary conditions for timing and the switching characteristics of the 7480 Group and 7481 Group, and Figure 1.21.1 shows the timing diagram.

Table 1.21.13 Necessary Conditions for Timing and Switching Characteristics (Note)

	Symbol	Parameter	Limits			Unit
			Min.	Typ.	Max.	
Clock synchronous	$t_{C(SCLK)}$	Serial I/O clock input cycle time	2000			ns
	$t_{WH(SCLK)}$	Serial I/O clock input HIGH pulse width	880			ns
	$t_{WL(SCLK)}$	Serial I/O clock input LOW pulse width	880			ns
	$t_{SU(RXD-SCLK)}$	Serial I/O input set-up time	160			ns
	$t_{h(SCLK-RXD)}$	Serial I/O input hold time	80			ns
	$t_{d(SCLK-TXD)}$	Serial I/O output delay time			100	ns
UART	$t_{C(SCLK)}$	Serial I/O clock input cycle time	500			ns
	$t_{WH(SCLK)}$	Serial I/O clock input HIGH pulse width	220			ns
	$t_{WL(SCLK)}$	Serial I/O clock input LOW pulse width	220			ns

Note: Values at V_{CC} = 4.5 V to 5.5 V, V_{SS} = 0 V, Ta = –40 °C to 85°C, and f(X_{IN}) = 8 MHz

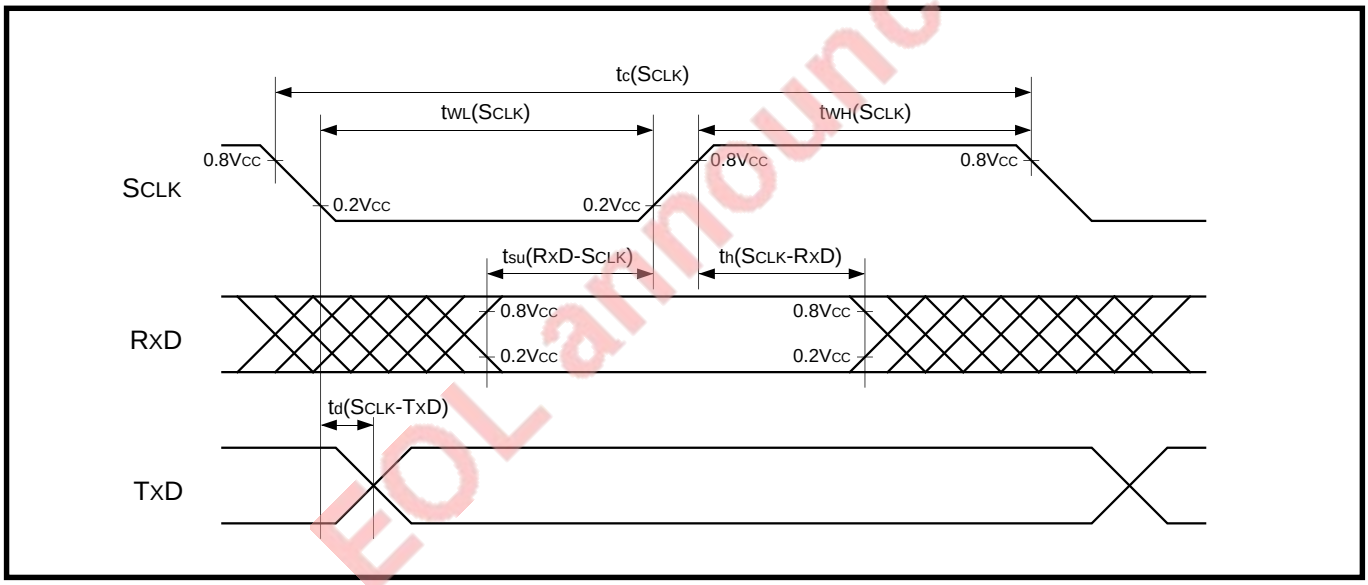


Figure 1.21.1 Timing Diagram

1.21.3 Typical Characteristics of Power Source Current

The typical characteristics of the power source current described in this section are based on a limited number of samples in the 7480 Group and 7481 Group. 'Typical values' are not guaranteed.
For the limits, refer to section **1.21.1 Electrical Characteristics**.

Figure 1.21.2 shows a measurement circuit of typical power source current characteristics.

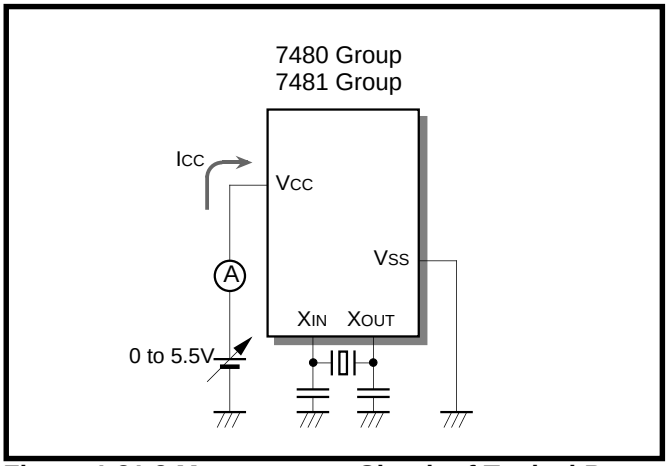


Figure 1.21.2 Measurement Circuit of Typical Power Source Current Characteristics

(1) Vcc–Icc Characteristics

Figures 1.21.3 to 1.21.6 show the Vcc–Icc characteristics of the 7480 Group and 7481 Group.

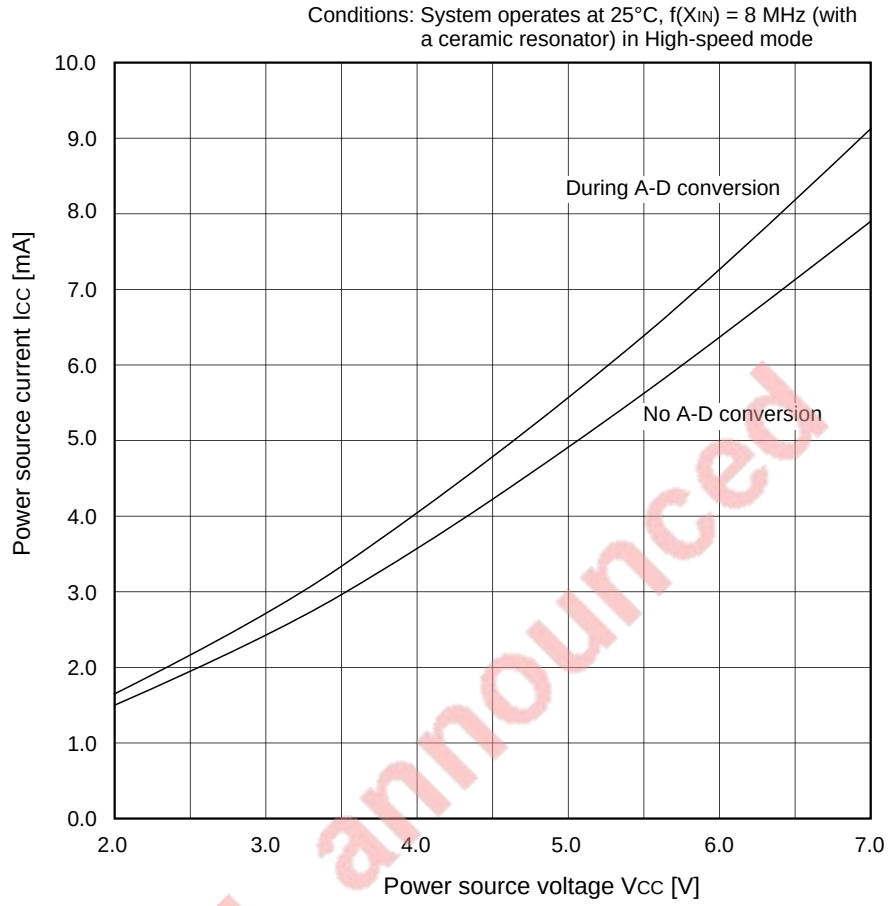


Figure 1.21.3 Vcc–Icc Characteristics (at System Operating in High-Speed Mode)

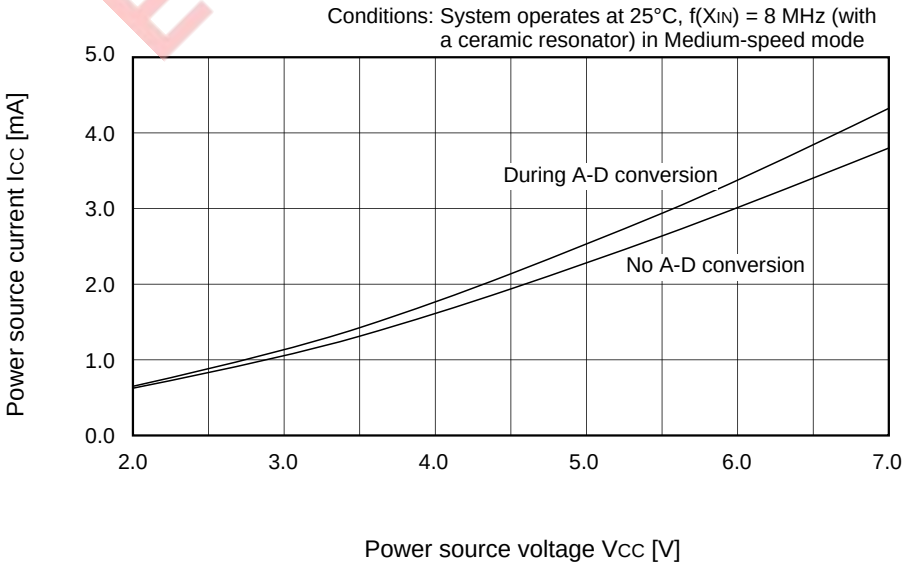


Figure 1.21.4 Vcc–Icc Characteristics (at System Operating in Medium-Speed Mode)

1.21 Electrical Characteristics

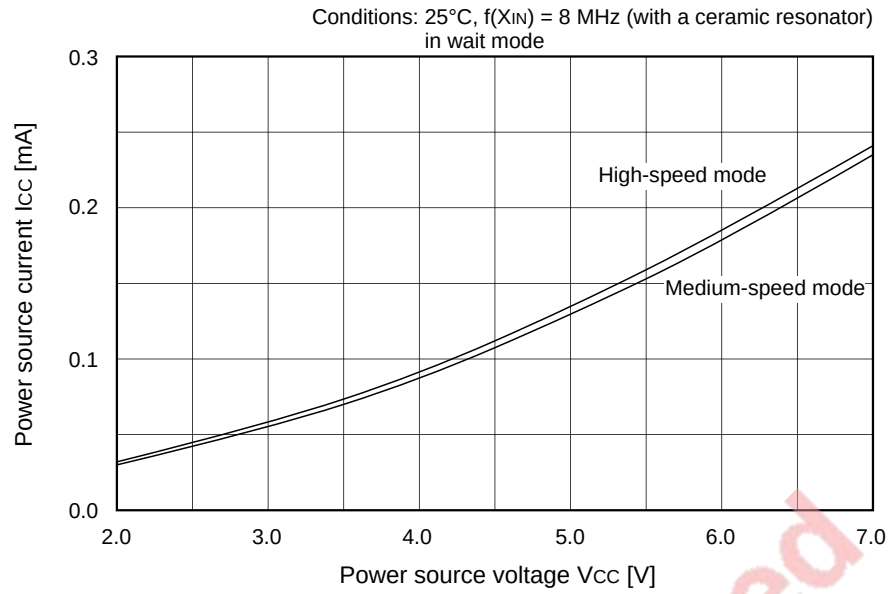


Figure 1.21.5 Vcc–Icc Characteristics (in Wait Mode)

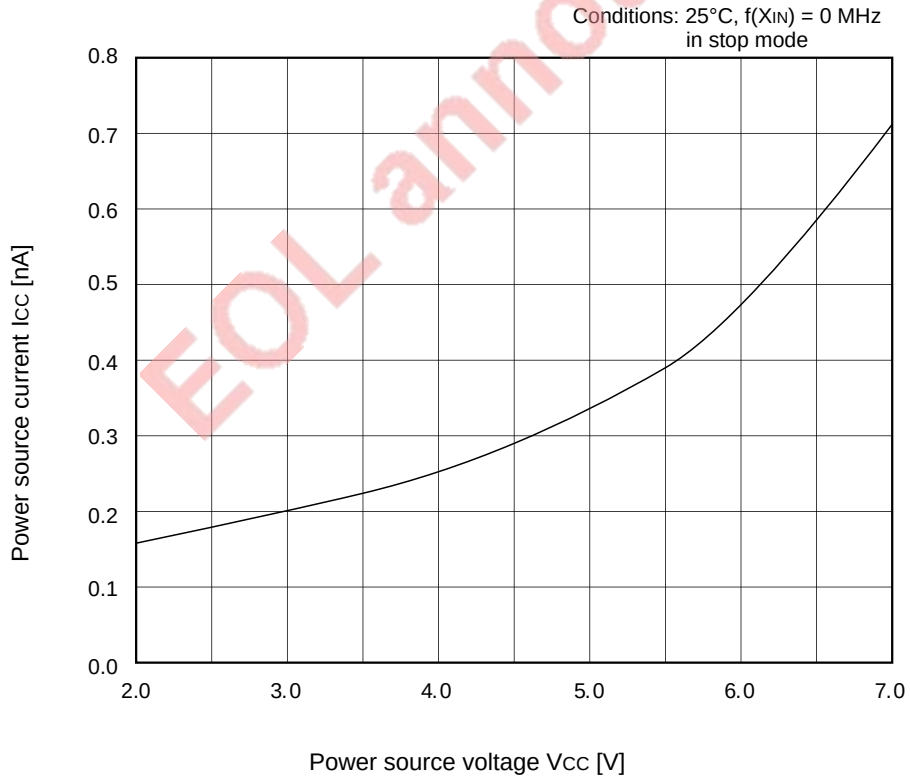


Figure 1.21.6 Vcc–Icc Characteristic (in Stop Mode)

(2) f(XIN)–ICC Characteristics

Figures 1.21.7 and 1.21.8 show the f(XIN)-ICC characteristics of the 7480 Group and 7481 Group.

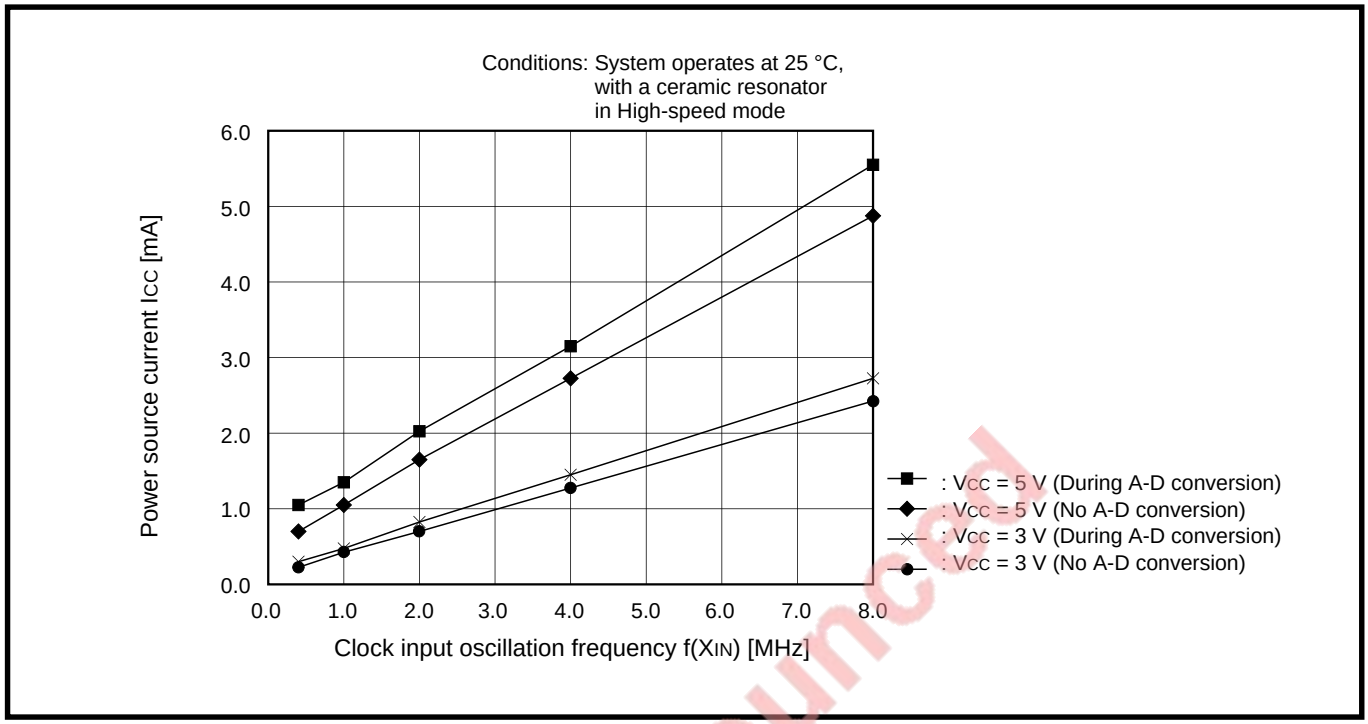


Figure 1.21.7 f(XIN)–ICC Characteristics (at System Operating in High-Speed Mode)

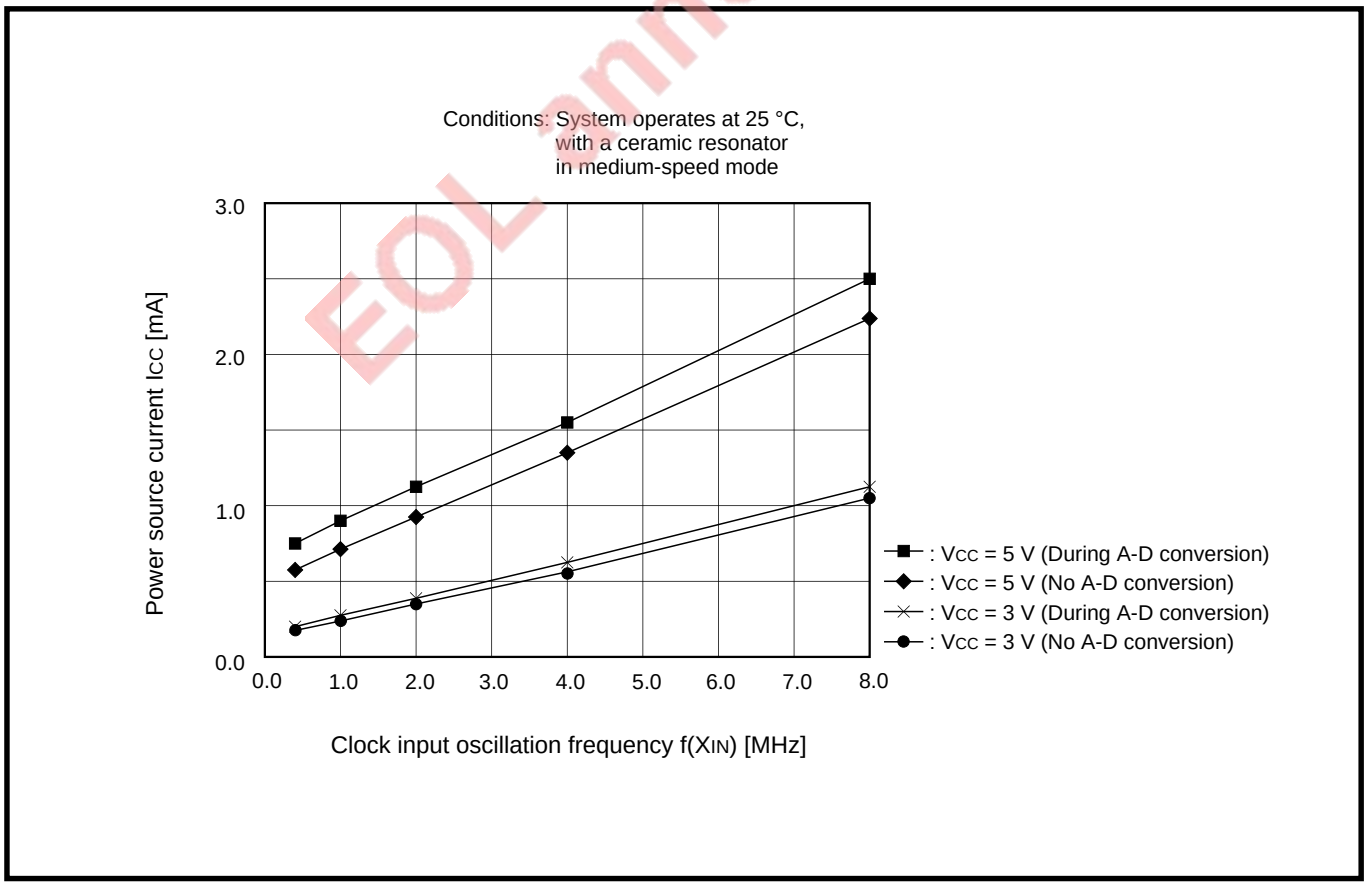


Figure 1.21.8 f(XIN)–ICC Characteristics (at System Operating in Medium-Speed Mode)

1.21 Electrical Characteristics

1.21.4 Typical Characteristics of Ports

The typical characteristics of the ports described in this section are based on limited numbers of samples in the 7480 Group and 7481 Group. 'Typical values' are not guaranteed. For the limits, refer to **Section 1.21.1 Electrical Characteristics**.

Figure 1.21.9 shows measurement circuits of typical port characteristics. Figures 1.21.10 through 1.21.12 show the typical characteristics of ports of the 7480 Group and 7481 Group.

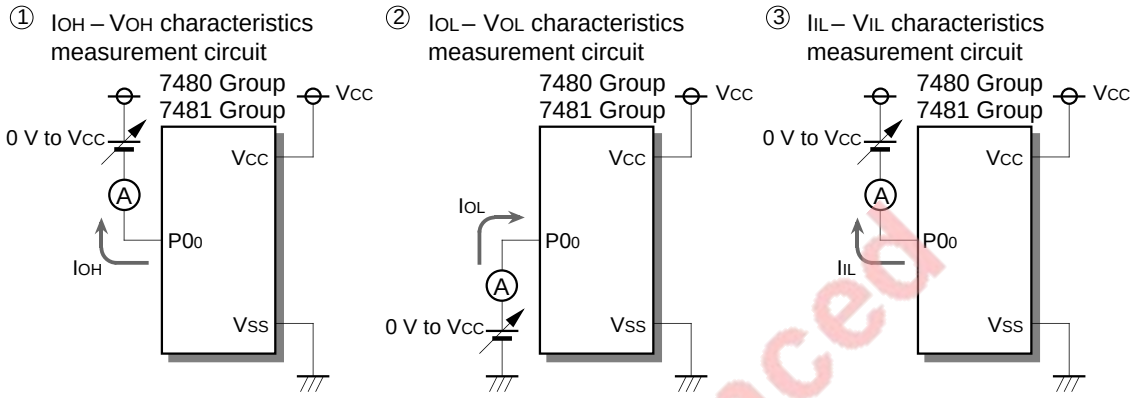


Figure 1.21.9 Measurement Circuits of Typical Port Characteristics

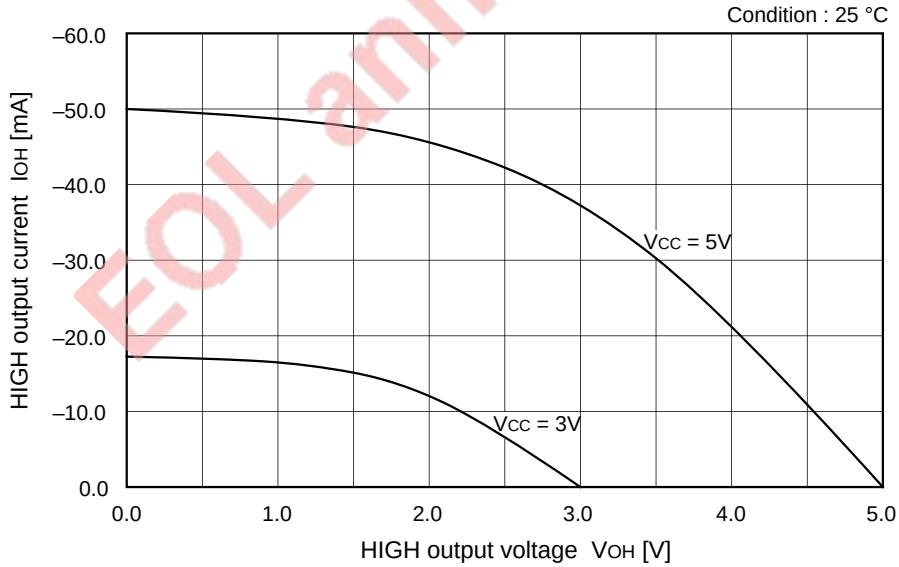


Figure 1.21.10 VOH–IOH Characteristics on P-Channel Side of Programmable I/O Port (CMOS Output)

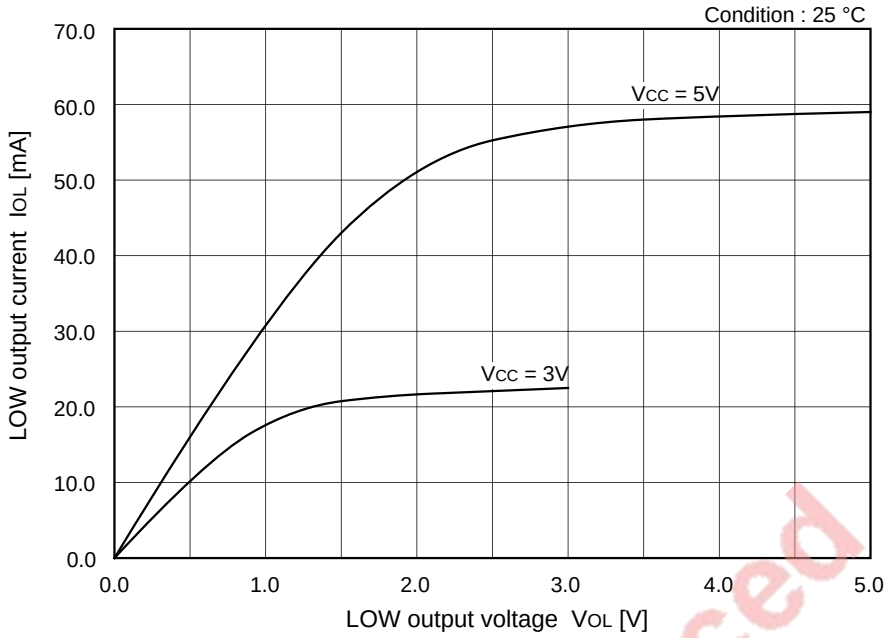


Figure 1.21.11 V_{OL}–I_{OL} Characteristics on N-Channel Side of Programmable I/O Port (CMOS Output)

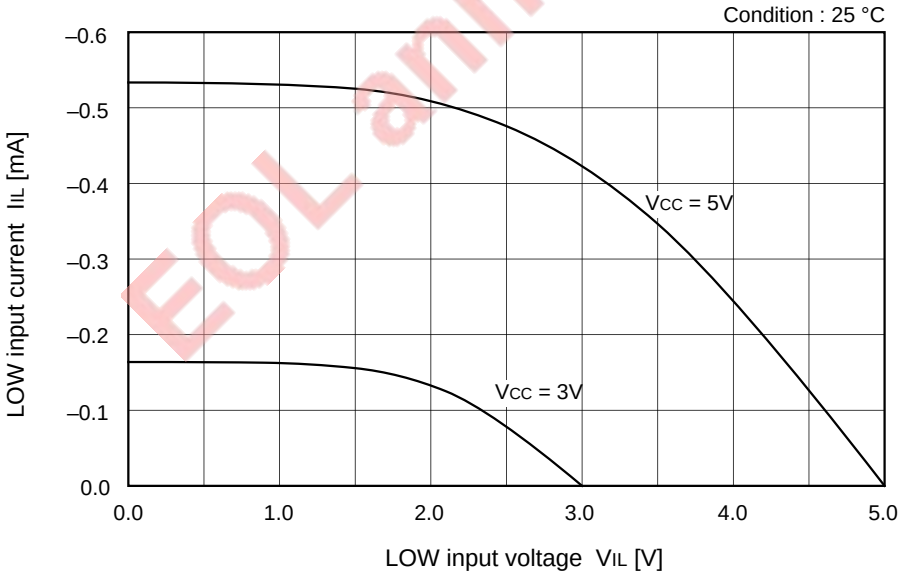


Figure 1.21.12 V_{IL}–I_{IL} Characteristics of Pull-up Transistor of Programmable I/O Port (CMOS Output)

1.21 Electrical Characteristics

1.21.5 Typical Characteristics of A-D Conversion

Figures 1.21.13 and 1.21.14 show typical characteristics of A-D conversion of the 7480 Group and 7481 Group in different measurement conditions.

The bottom line in each graph shows an absolute accuracy error (ERROR), indicating offset from the ideal value at the point where an output code changes.

For example, a '$3F_{16} \rightarrow 40_{16}$' change in an output code ideally takes place at the point where IN_0 = 1270mV, in Figure 1.21.13.

However, 1270–1 = 1269 mV is obtained as the measured changing point, because the absolute accuracy error is –1 mV.

The top line in each graph represents the width of input voltages that have the same output code in 1-LSB WIDTH.

For example, '21–20 = 1 mV (0.05 LSB)' is obtained as the differential non-linear error because the measurement value of the width of input voltages whose output codes are '$3F_{16}$', is 21 mV.

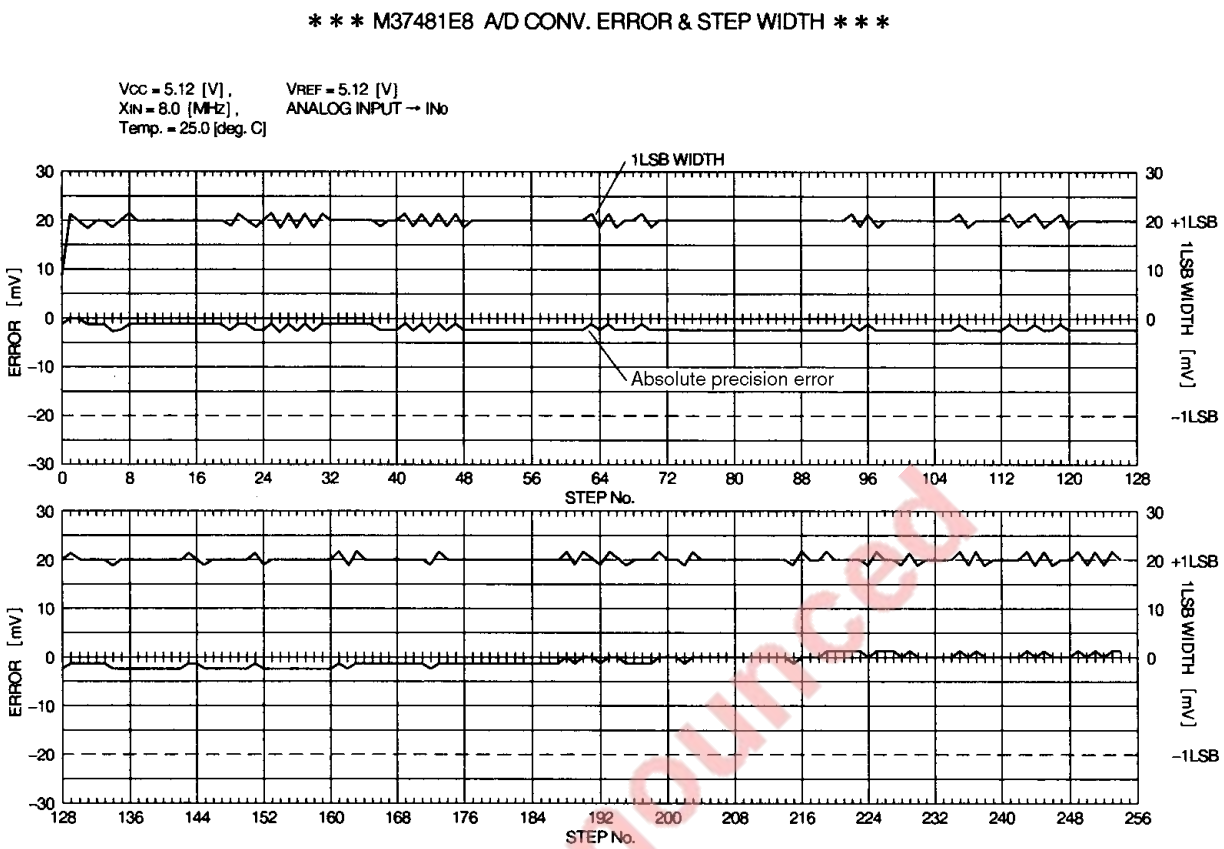


Figure 1.21.13 Typical Characteristics of A-D Conversion (1)

1.21 Electrical Characteristics

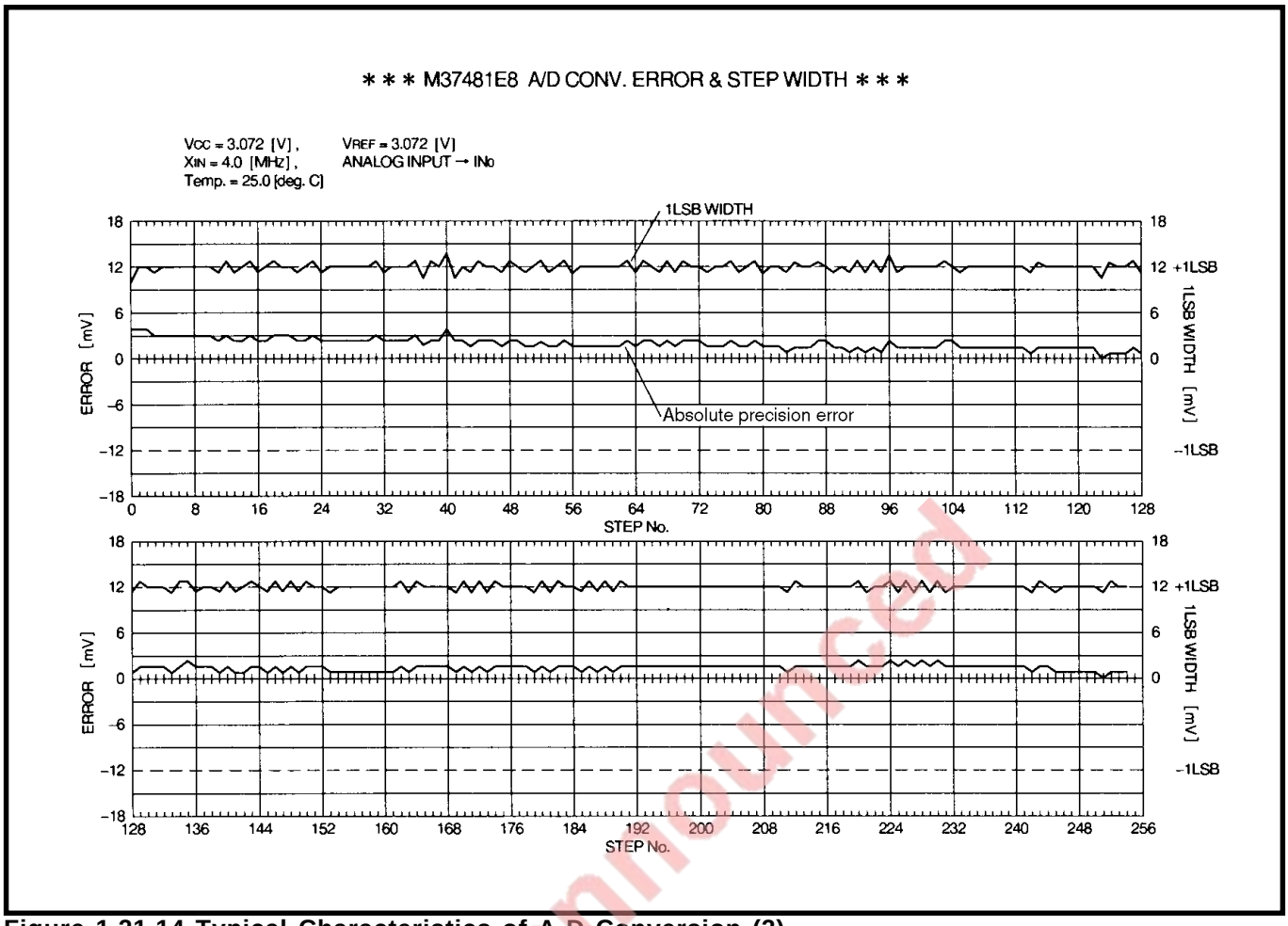


Figure 1.21.14 Typical Characteristics of A-D Conversion (2)

CHAPTER 2
APPLICATIONS

2.1 Input/Output Pins

(1) External Circuit Example for Output Ports

POINT: The following currents and voltages must be within specifications in the recommended operating conditions when external circuits for I/O ports are designed.

■For Input Ports
- ●Input voltage
- ●Input current

■For Output Ports
- ●Output sum currents
- ●Output peak current
- ●Output average current

For the recommended operating conditions, refer to **Section 1.21 Electrical Characteristics.**

Note: When a key matrix is used for multi-key inputs, take account of the total current which results from multiple inputs and is input to one port.

Figure 2.1.1 shows an external circuitry for output ports.

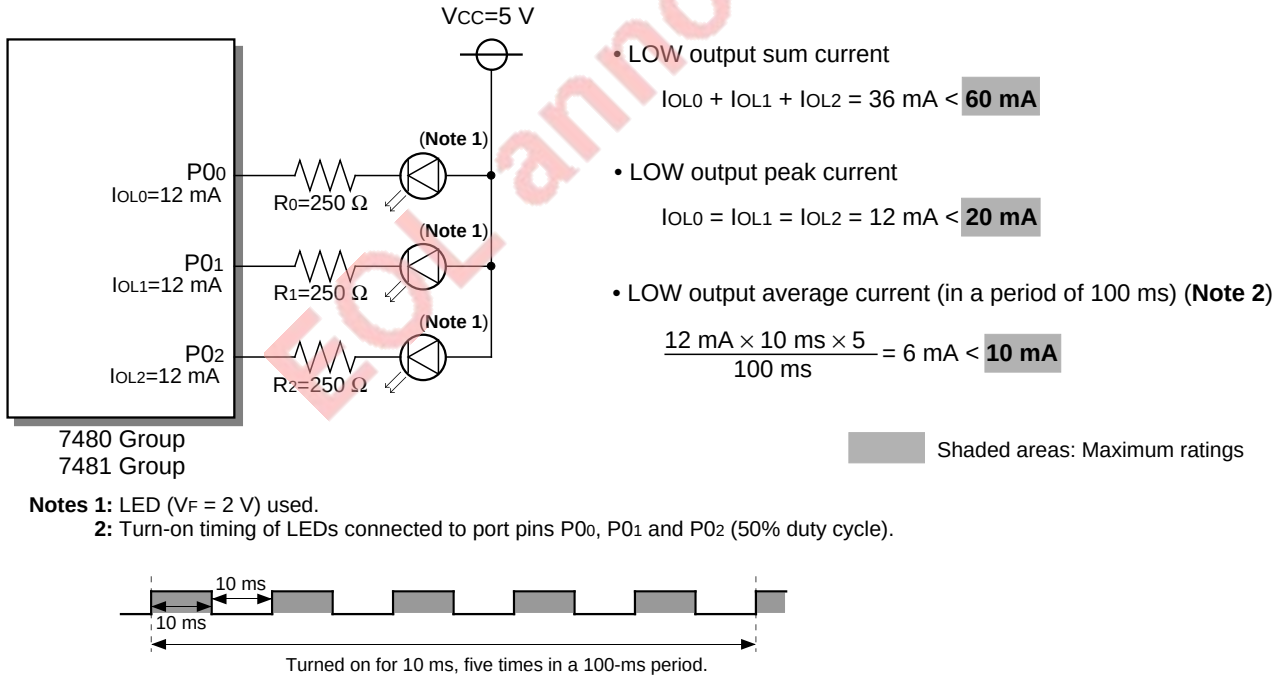


Figure 2.1.1 External Circuit for Output Ports

(2) Simplifying External Circuit Example by Using Level Shift Port and Noise Margin

POINT: ●Ports P4 and P5 have N-channel open-drain outputs. Built-in clamping diodes allow voltages V_{CC} or more to be applied to port pins when the current for a pin is 1 mA or less.
●Voltages V_{IL} = 0.4 V_{CC} and V_{IH} = 0.8 V_{CC} can be applied to ports P3, P4, and P5 (at V_{CC} = 4.5 V 5.5 V).

Figure 2.1.2 shows a simplified external circuit example by using a level shift port and noise margin.

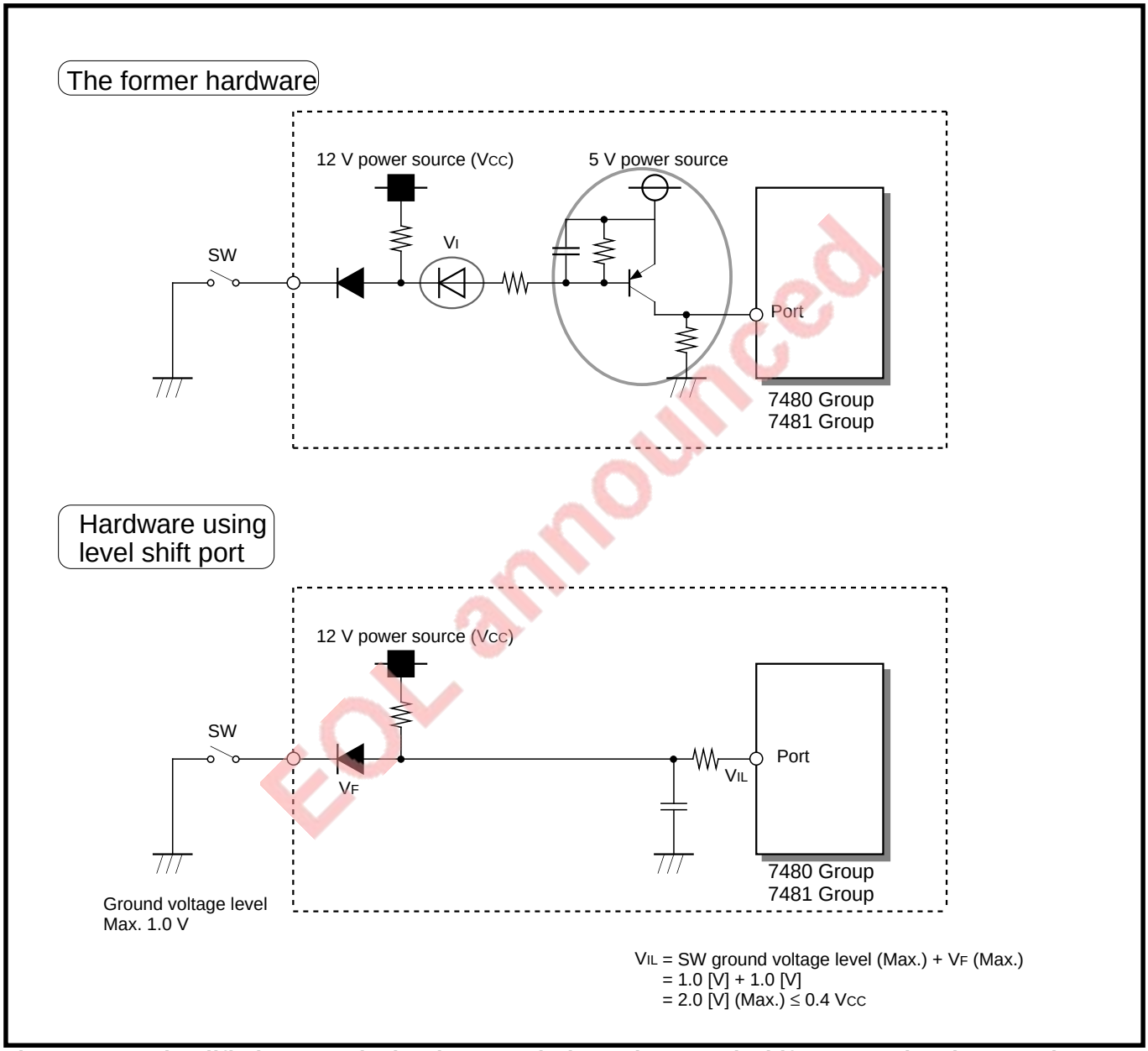


Figure 2.1.2 Simplified External Circuit Example by Using Level Shift Port and Noise Margin

2.2 Timer X and Timer Y

2.2.1 Application Example of Timer Mode

Generation of period of 100 ms (100-ms Periodic Processing)

POINT: The clock is divided by timer X, and the CPU performs periodic processing with a timer X interrupt service routine generated every 100 ms.

SPECIFICATIONS:
- Clock: f(XIN) = 8 MHz
- A timer X interrupt is generated every 100 ms using the timer mode of timer X.
- Periodic processing is performed every 100 ms with timer X interrupt service routine.

Figure 2.2.1 shows a setting example of the division ratio.
Figure 2.2.2 shows a control procedure example of 100-ms processing.

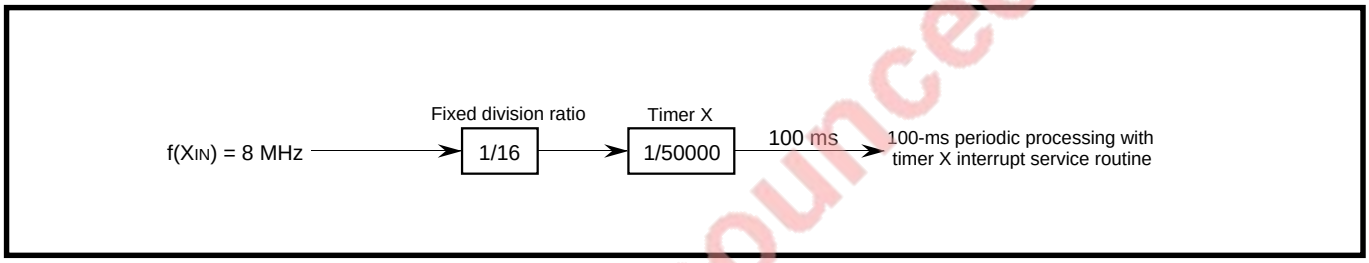
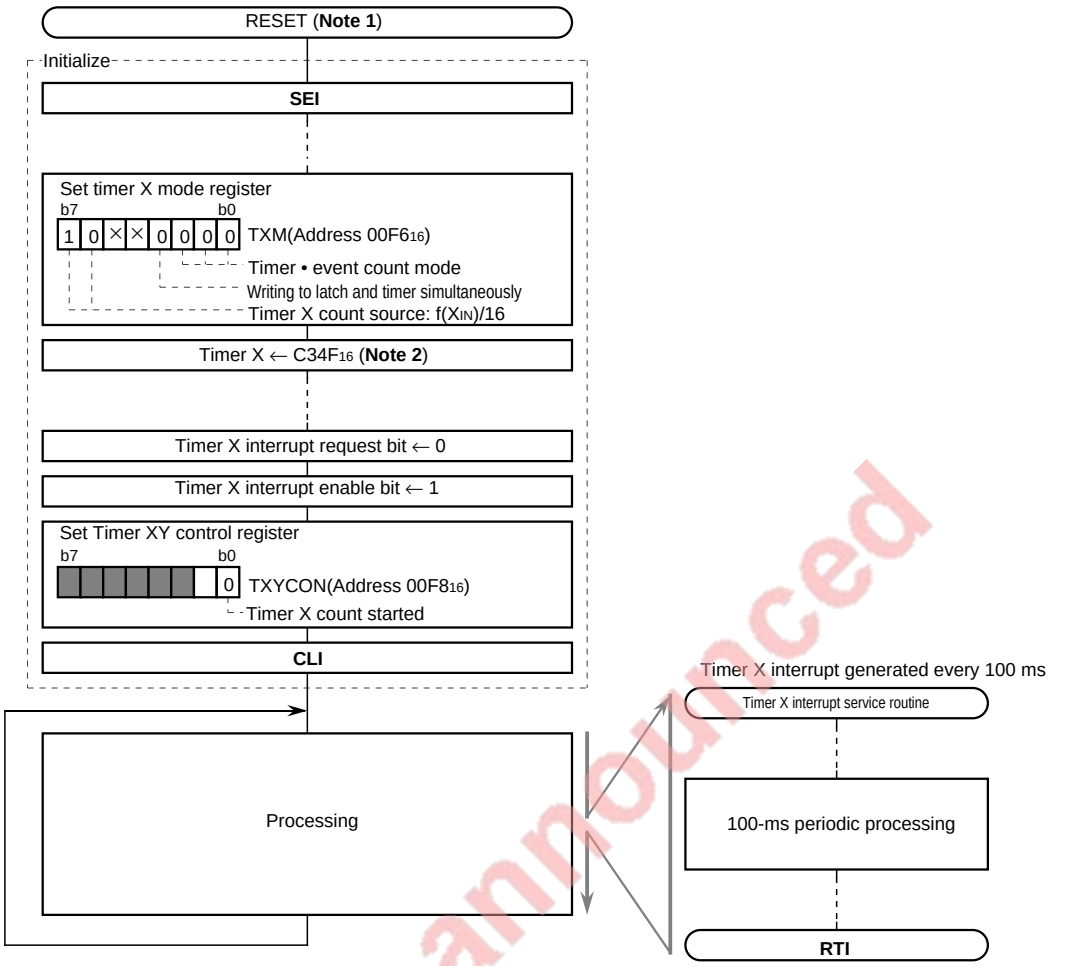


Figure 2.2.1 Setting Example of Division Ratio



Notes 1: State after system is released from reset
- Timer X stop control bit = 1 (count stopped)

2: • $C34F_{16} = 50000 - 1$
- Write in order of low-order to high-order byte.

Figure 2.2.2 Control Procedure Example of 100-ms Processing

2.2.2 Application Example of Event Count Mode

Measurement of Water Flow Rate

POINT: Pulses generated corresponding to the water flow rate are counted for a fixed period (100 ms), and the water flow rate during this period is calculated.

SPECIFICATIONS:
- Clock: f(XIN) = 8 MHz
- Pulses generated corresponding to the water flow rate are input to the CNTR1 pin and counted using the event count mode of timer Y.
- The contents of timer Y are read in the timer X interrupt service routine generated after 100 ms from the start of counting pulses, and the water flow rate during 100 ms is calculated.

Figure 2.2.3 shows a peripheral circuit example.
Figure 2.2.4 shows the method of measuring water flow rate.
Figure 2.2.5 shows the control procedure example of measuring water flow rate.
For the setting example of division ratio from timer X, refer to **Figure 2.2.1**.

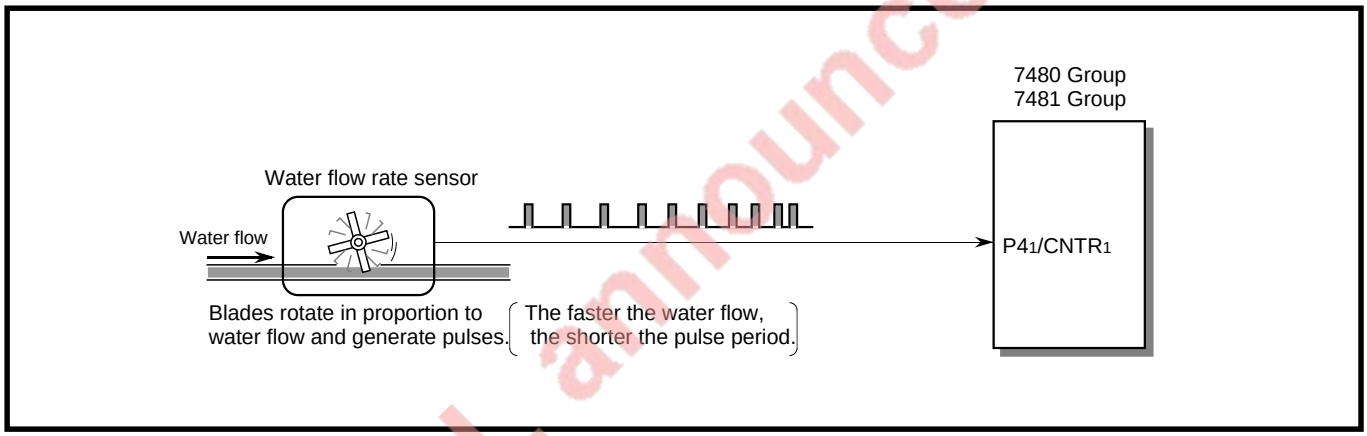


Figure 2.2.3 Peripheral Circuit Example

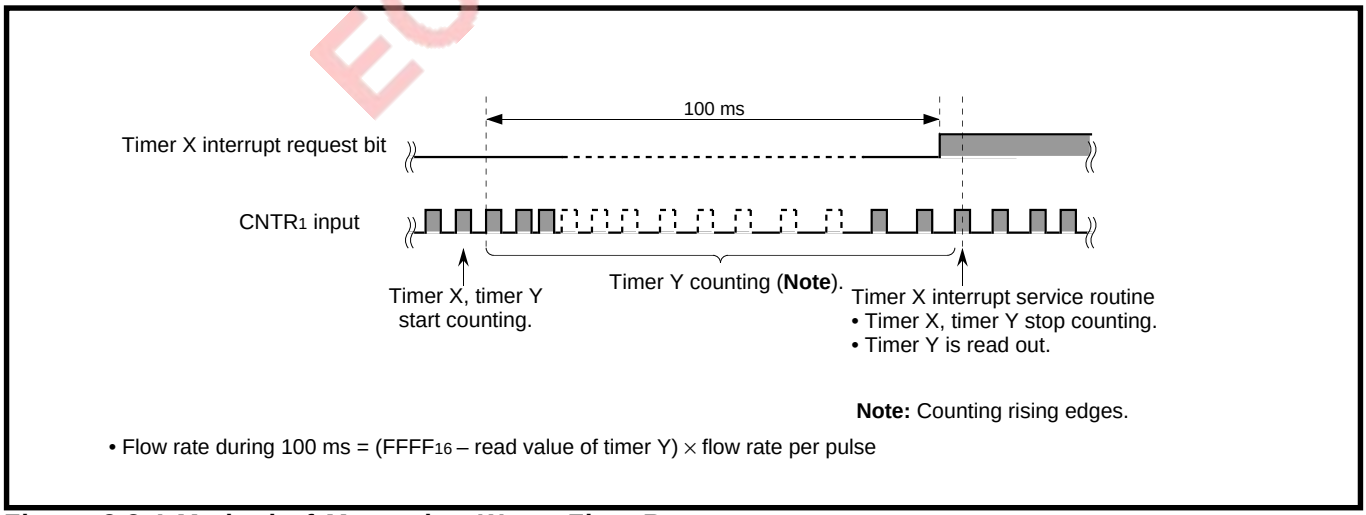
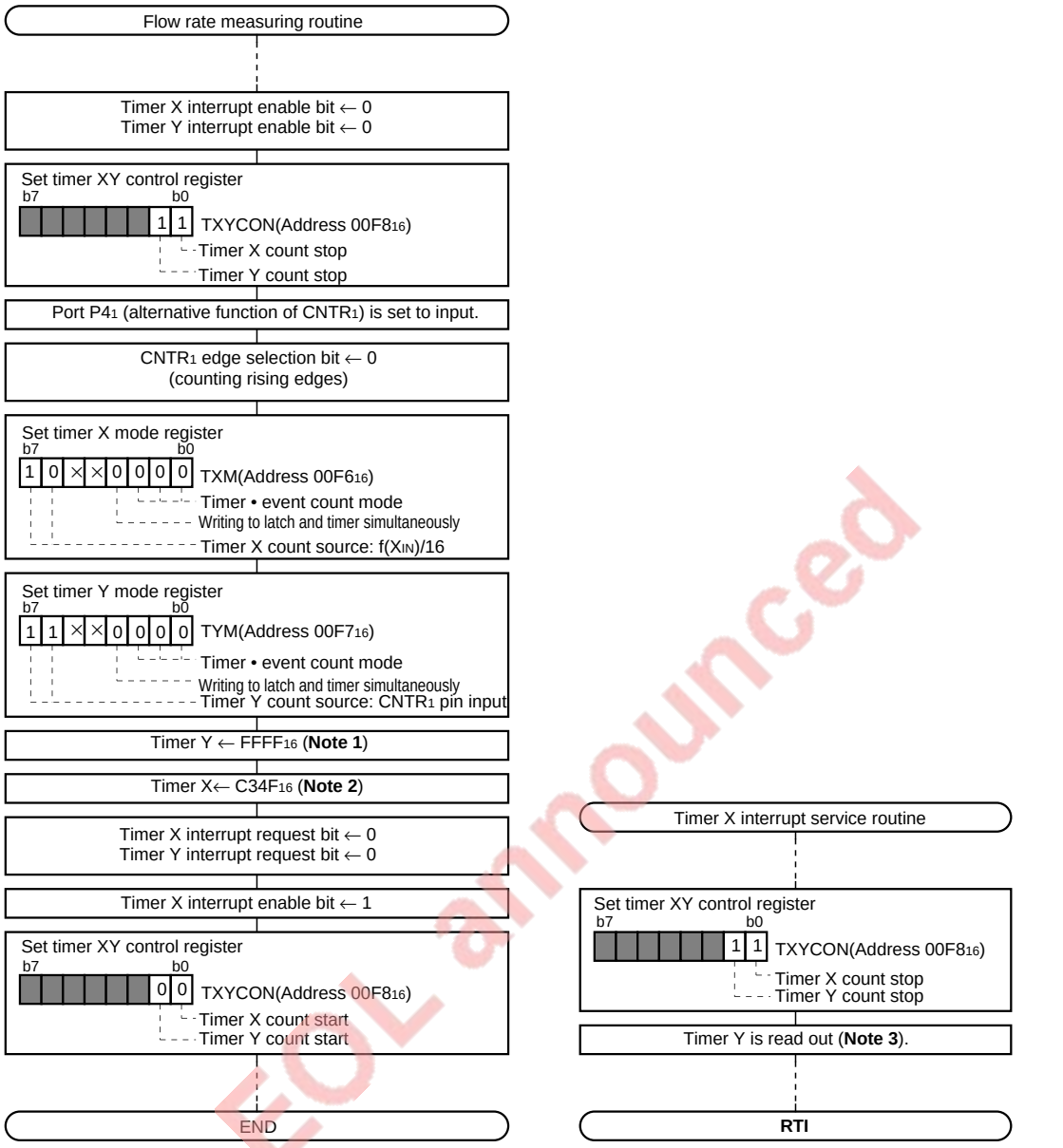


Figure 2.2.4 Method of Measuring Water Flow Rate



Note 1: • Initial value.
• Write to timer Y in order of low-order to high-order byte.
2: • $C34F_{16}$ = 50000 – 1.
• Write to timer X in order of low-order to high-order byte.
3: • ($FFFF_{16}$ – the read value of Timer Y) = the number of $CNTR_1$ edges detected during 100 ms.
• Read from timer Y in order of high-order to low-order byte.

Figure 2.2.5 Control Procedure Example of Measuring Water Flow Rate

2.2.3 Application Example of Pulse Output Mode

Piezoelectric Buzzer Output

POINT: The pulse output mode of a 16-bit timer is used for buzzer output.

SPECIFICATIONS:
- Clock: f(XIN) = 8 MHz
- 4 kHz pulses are output from the CNTR0 pin using the pulse output mode of timer X.
- CNTR0 pin output level is fixed to HIGH while the buzzer output is stopped.

Figure 2.2.6 shows a peripheral circuit example.
Figure 2.2.7 shows a setting example of the division ratio.
Figure 2.2.8 shows a control procedure example of buzzer output.

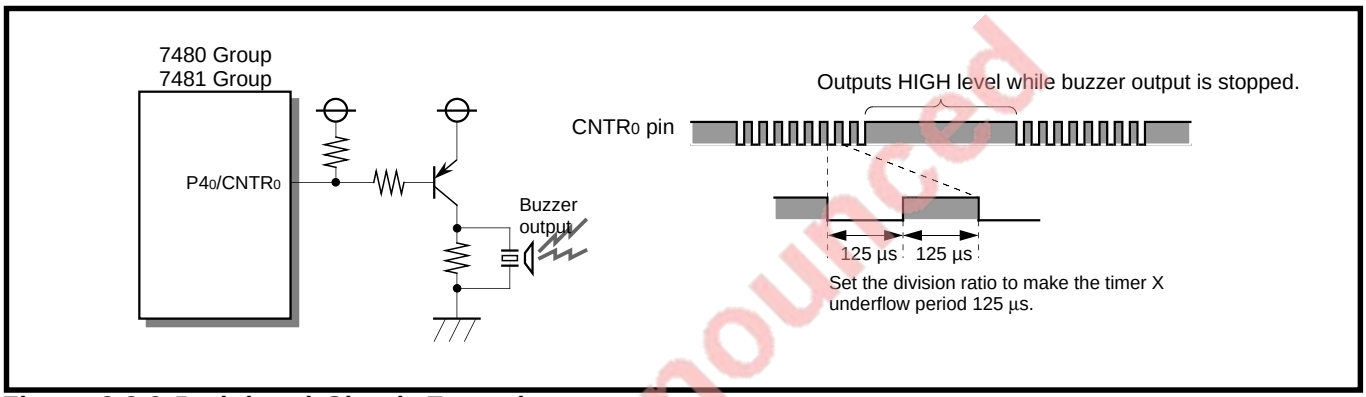


Figure 2.2.6 Peripheral Circuit Example

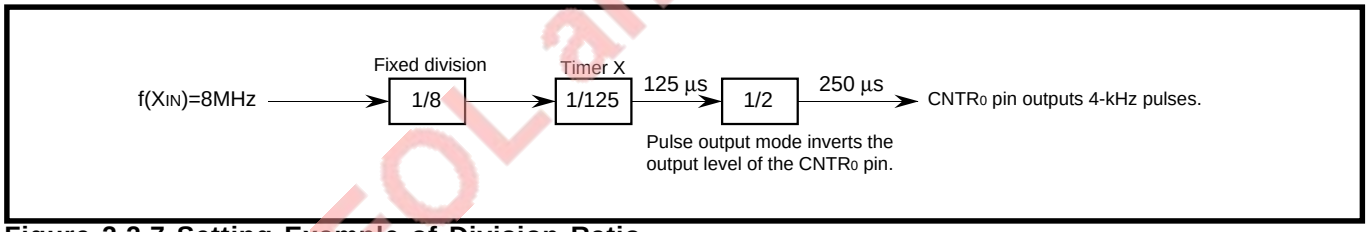
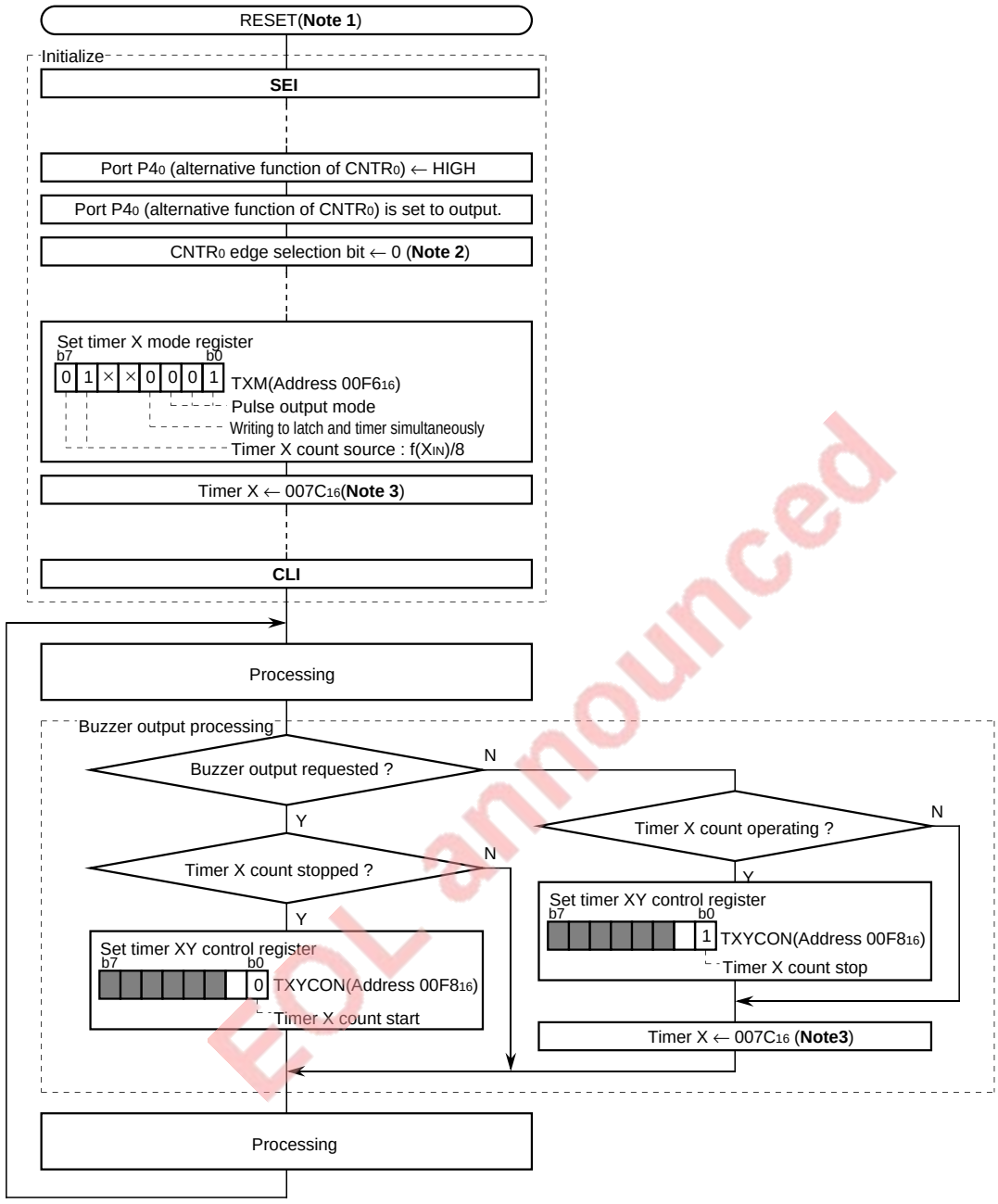


Figure 2.2.7 Setting Example of Division Ratio



Notes 1: State after system is released from reset
- Timer X stop control bit = 1 (count stop)

2: In pulse output mode, the output level of CNTR0 pin is initialized to HIGH by writing to timer X (writing to latch and timer simultaneously).

3:
- $007C_{16}$ = 125–1
- Write to timer X in order of low-order to high-order byte.
- The output level of CNTR0 pin is initialized by writing to timer X (writing to latch and timer simultaneously).

Figure 2.2.8 Control Procedure Example of Buzzer Output

2.2.4 Application Example of Pulse Period Measurement Mode

Phase Control of Load (Measuring Period of Feedback Signal)

POINT: The period of the feedback signal input from the load is measured using the pulse period measurement mode of a 16-bit timer.

SPECIFICATIONS:
- Clock: f(XIN) = 8 MHz
- Phase control signal is output to the load and controls the load's phase.
- The period of the feedback signal input to the CNTR0 pin from the load is measured using the pulse period measurement mode of timer X.
  - Count source: f(XIN)/16
- The period of the feedback signal is analyzed to adjust the phase control signal input to the load.

For the output of the phase control signal, refer to **Section 2.2.7 Application Example of Programmable One-shot Output Mode**.
Figure 2.2.9 shows a peripheral circuit example.
Figure 2.2.10 shows a phase control procedure example.

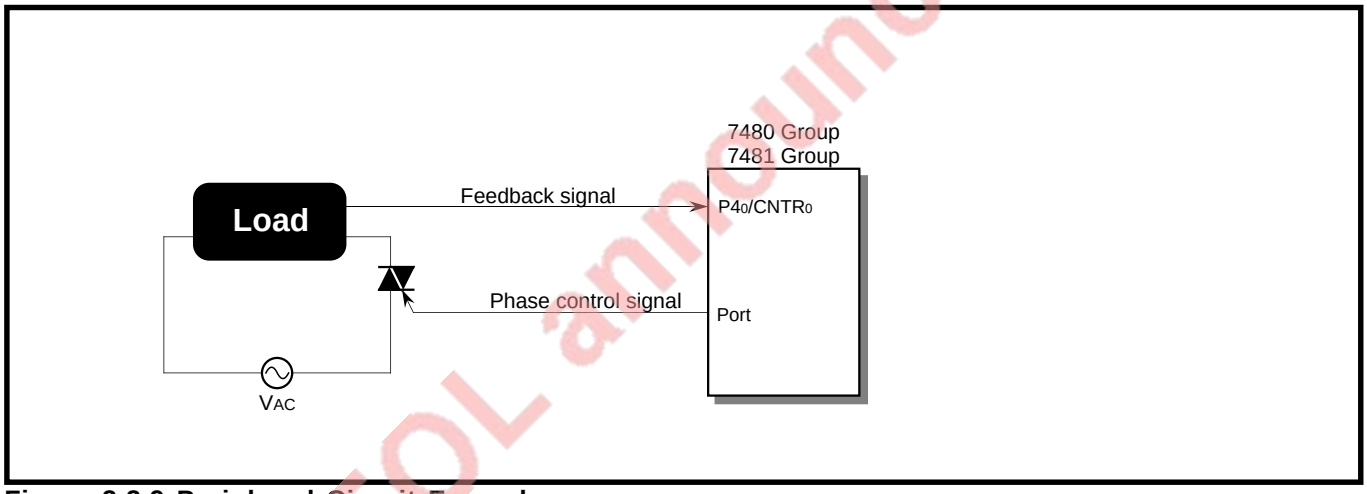
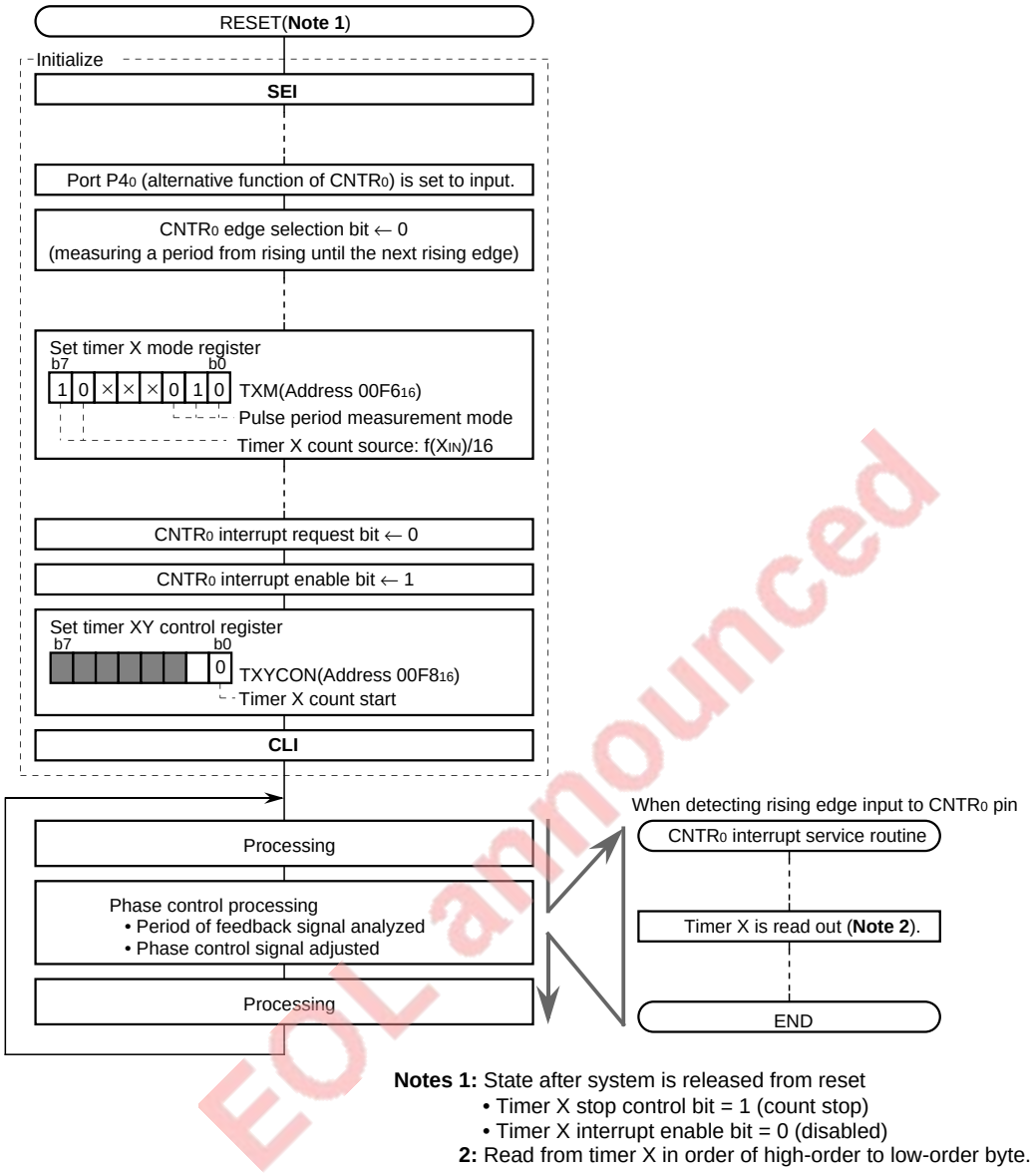


Figure 2.2.9 Peripheral Circuit Example



Notes 1: State after system is released from reset
- Timer X stop control bit = 1 (count stop)
- Timer X interrupt enable bit = 0 (disabled)

2: Read from timer X in order of high-order to low-order byte.

Figure 2.2.10 Phase Control Procedure Example

2.2.5 Application Example of Pulse Width Measurement Mode

Communications between Two Microcomputers (Reception)

POINT: 8-bit data is received by measuring each bit's HIGH-level width input to the CNTR pin and identifying each bit data.

SPECIFICATIONS:
- Clock: f(XIN) = 8 MHz
- The HIGH-level width of CNTR0 pin input is measured using the pulse width measurement mode of timer X.
  - Count source: f(XIN)/8
- The start, stop bits and each bit data of 8-bit receive data are identified by the measured values of the HIGH-level widths.

Figure 2.2.11 shows a peripheral circuit example.
Figure 2.2.12 shows a communication format example.
Figure 2.2.13 shows a communication control procedure example.

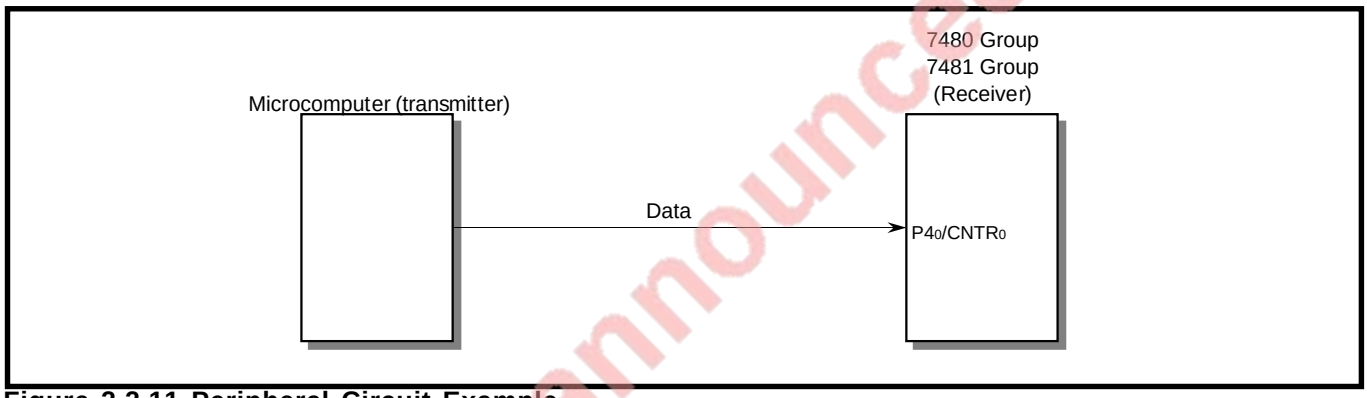


Figure 2.2.11 Peripheral Circuit Example

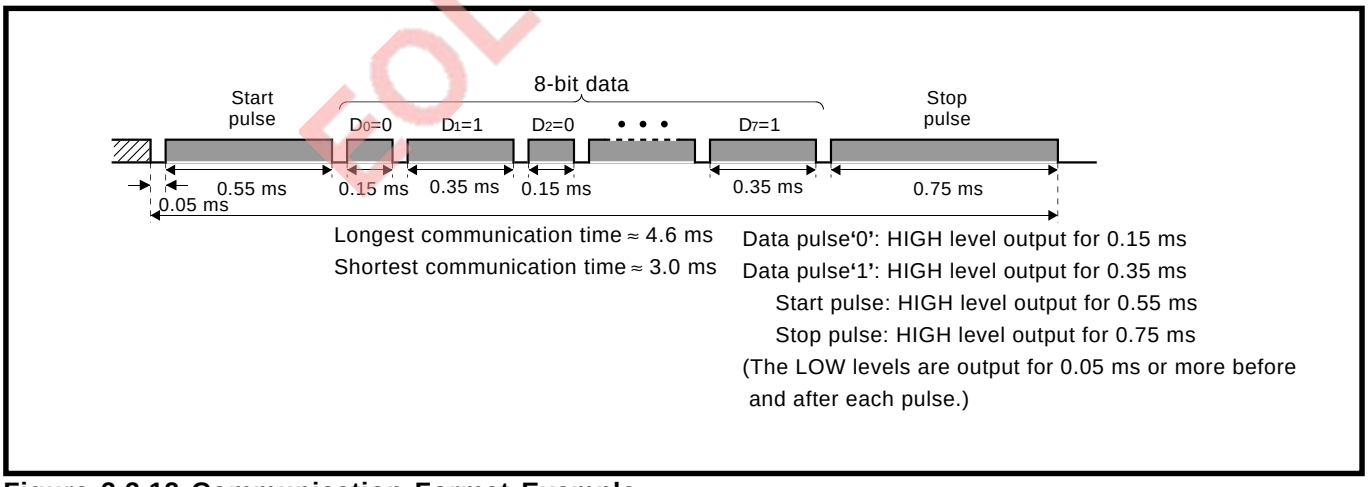


Figure 2.2.12 Communication Format Example

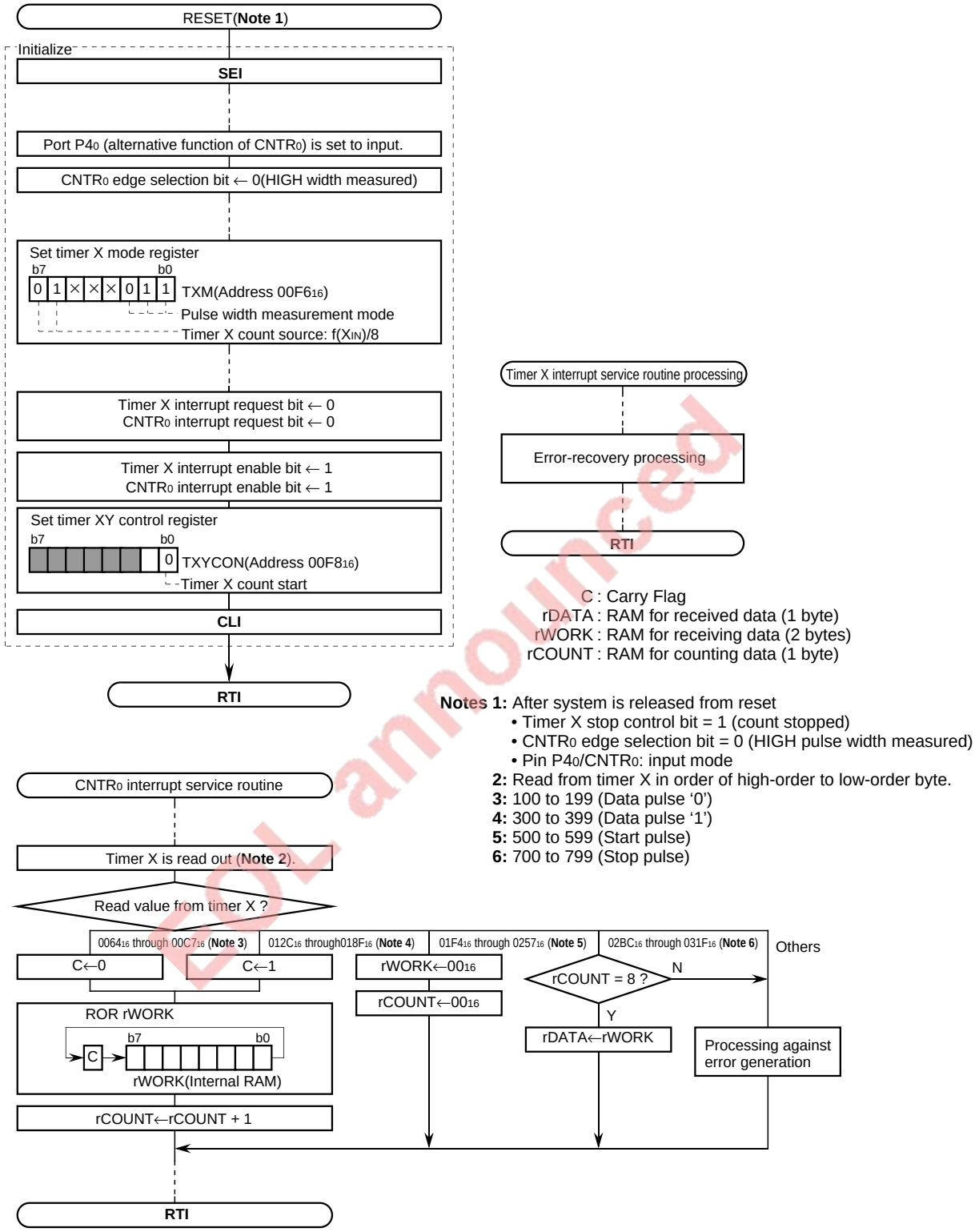


Figure 2.2.13 Communications Control Procedure Example

2.2.6 Application Example of Programmable Waveform Generation Mode

Control of Motorcycle Single-Cylinder Engine

POINT: A trigger input to the INT pin automatically starts a timer counting. This allows the CNTR pin output to be changed with a more accurate timing than counting is started in an INT interrupt service routine.

SPECIFICATIONS:
- Clock: f(XIN) = 8 MHz
- A rise-to-rise period of a crank angle signal input to the CNTR0 pin is measured using the pulse period measurement mode of timer X to determine the correction value of timer Y.
- The trigger of the crank angle signal input to the INT1 pin makes timer Y activated. Then, the control signal of the igniter is output from the CNTR1 pin using the programmable waveform generation mode.

For the pulse period measurement of the crank angle signal, refer to **Section 2.2.4 Application Example of Pulse Period Measurement Mode**.
Figure 2.2.14 shows a peripheral circuit example.
Figure 2.2.15 shows an operation timing example.
Figure 2.2.16 shows a control procedure example of motorcycle engine.

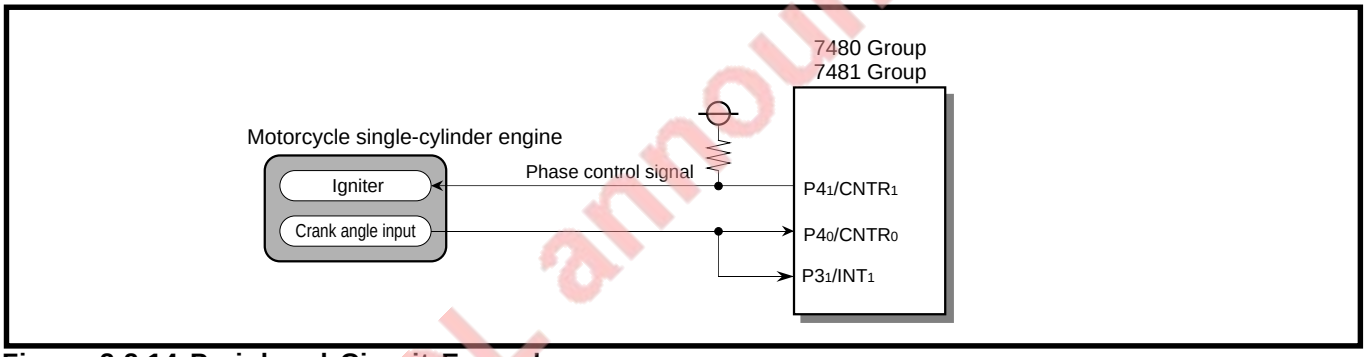


Figure 2.2.14 Peripheral Circuit Example

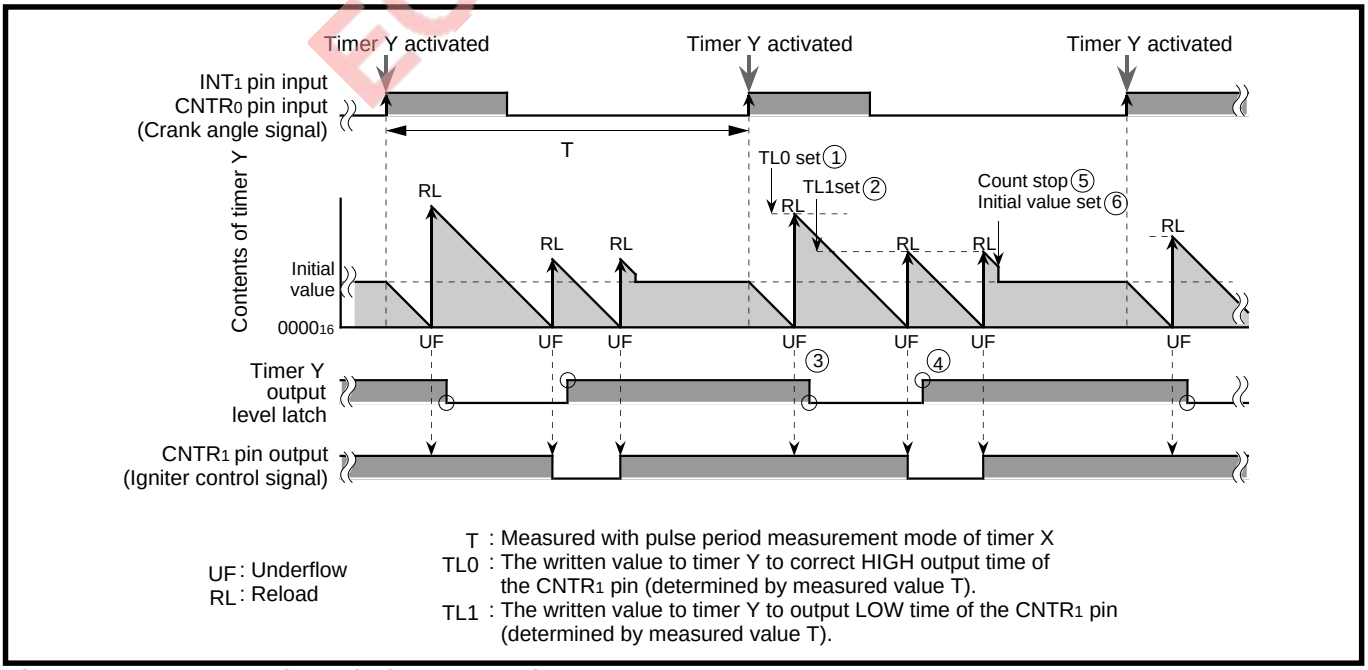
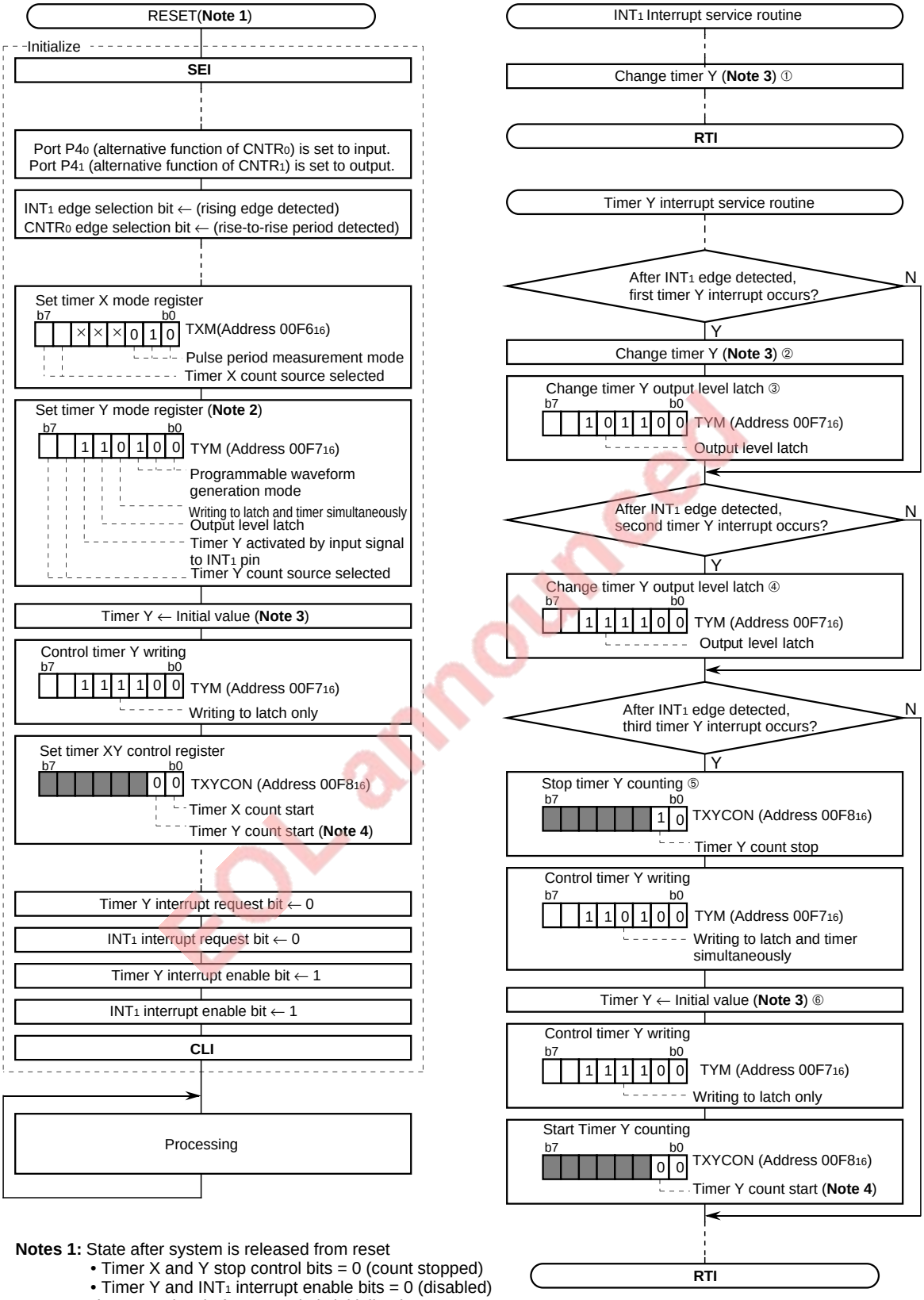


Figure 2.2.15 Operation Timing Example



Notes 1: State after system is released from reset
- Timer X and Y stop control bits = 0 (count stopped)
- Timer Y and INT1 interrupt enable bits = 0 (disabled)

2: The output level of CNTR1 pin is initialized to LOW.
3: Write to timer Y in order of low-order to high-order byte.
4: In this time, timer Y remains still stopped.

Figure 2.2.16 Control Procedure Example of Motorcycle Engine

2.2.7 Application Example of Programmable One-Shot Output Mode

Phase Control of Load (Output of Phase Control Signals)

POINT: The phase control signal to the load is output using the programmable one-shot output mode of a 16-bit timer.

SPECIFICATIONS:
- Clock: f(XIN) = 8 MHz
- The phase control signal to the load is output from the CNTR1 pin using the programmable one-shot output mode of timer Y.
  - Count source: f(XIN)/16
  - Rising edges of the signal input to the INT1 pin from the trigger detection circuit are detected.
  - A triac is turned on at the HIGH level.
- The period of the feedback signal input from the load is measured, analyzed, and used to adjust the phase control signal.

For the measurement of the period of the feedback signal, refer to **Section 2.2.4 Application Example of Pulse Period Measurement Mode**.
Figure 2.2.17 shows a peripheral circuit example.
Figure 2.2.18 shows an operation timing example.
Figure 2.2.19 shows a phase control procedure example.

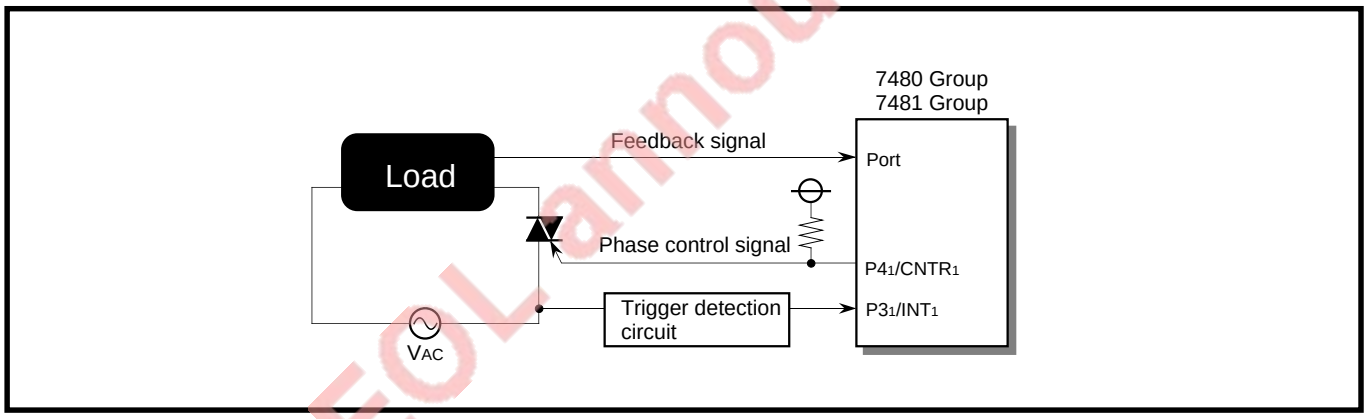


Figure 2.2.17 Peripheral Circuit Example

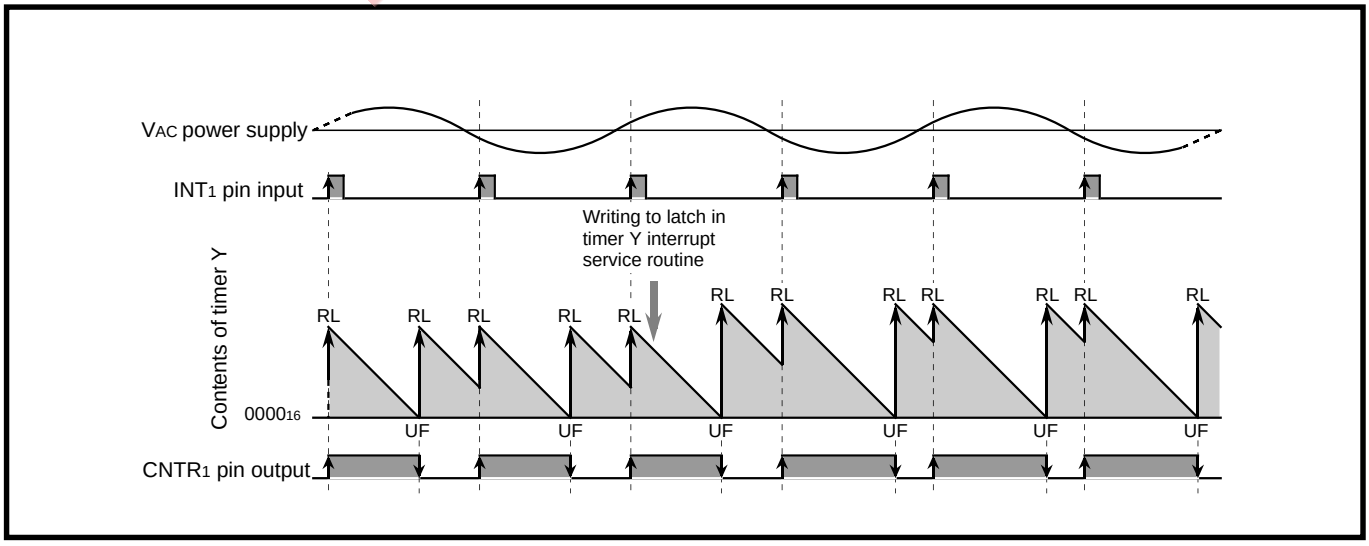
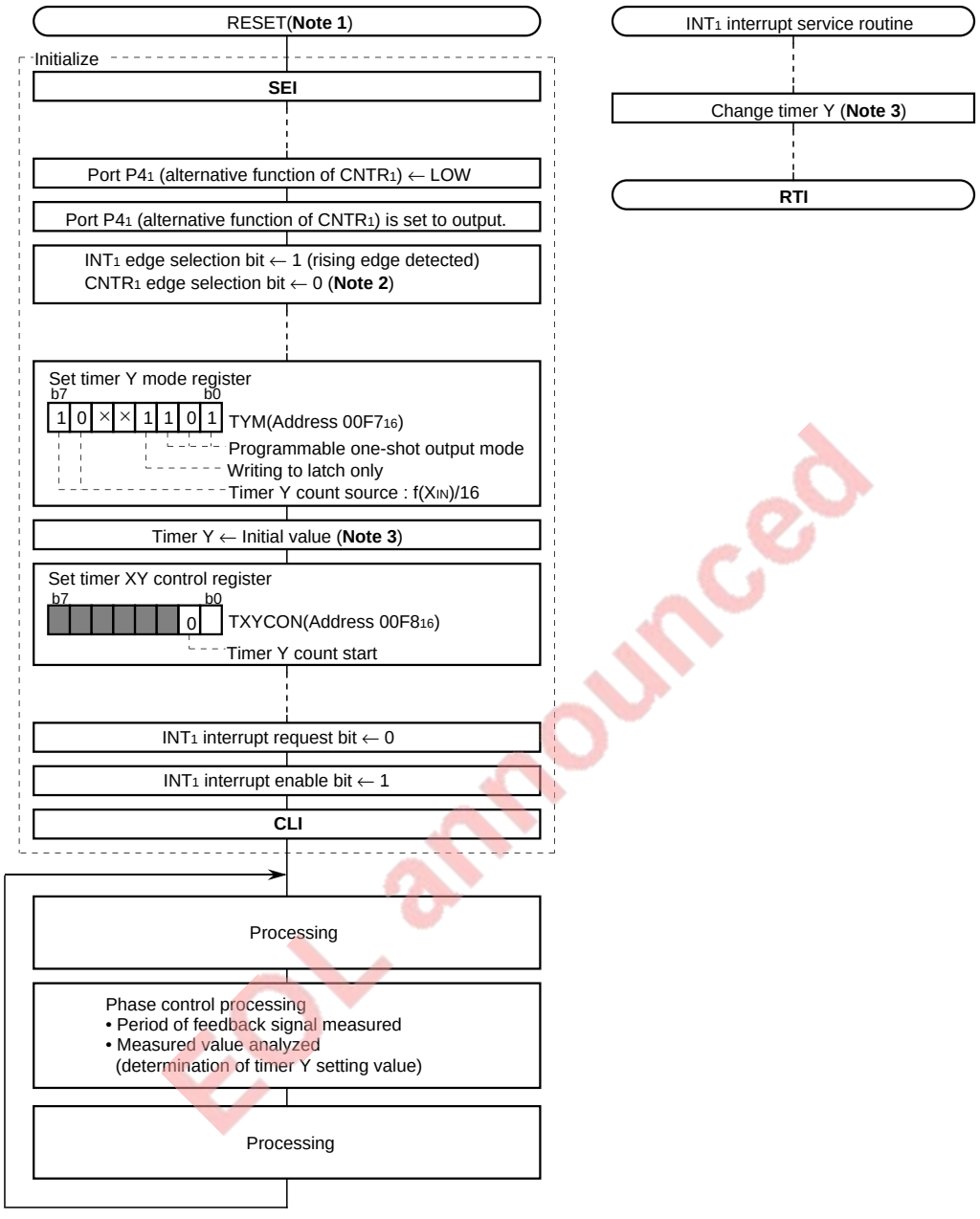


Figure 2.2.18 Operation Timing Example



Notes 1: State after system is released from reset
- Timer Y stop control bit = 0 (count stop)
- Timer Y interrupt enable bit = 0 (disabled)

2: HIGH level one-shot pulse is output.
3: Write to timer Y in order of low-order to high-order byte.

Figure 2.2.19 Phase Control Procedure Example

2.2.8 Application Example of PWM Mode

Output of Analog Voltage

POINT: Analog voltage is output using the PWM waveform.

SPECIFICATIONS:
- Clock: f(XIN) = 8 MHz
- PWM waveforms are output from the CNTR0 pin using the PWM mode of timer X.
  - Count source: f(XIN)/16
  - The duty cycle of PWM waveforms is determined depending on analog voltage output.
- PWM waveforms are converted into the analog voltage using the external circuit to the CNTR0 pin.

Figure 2.2.20 shows a peripheral circuit example.
Figure 2.2.21 shows a control procedure example of analog voltage output.

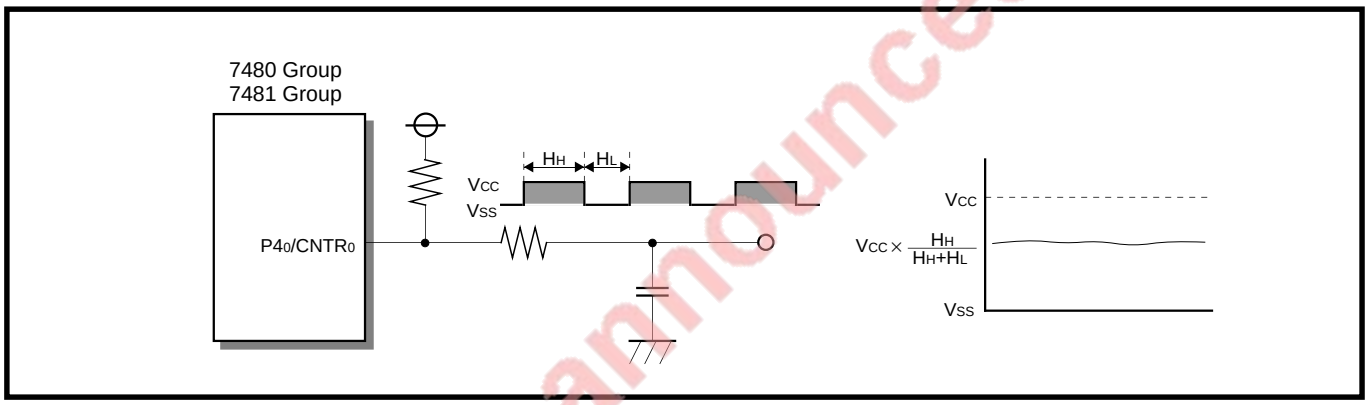
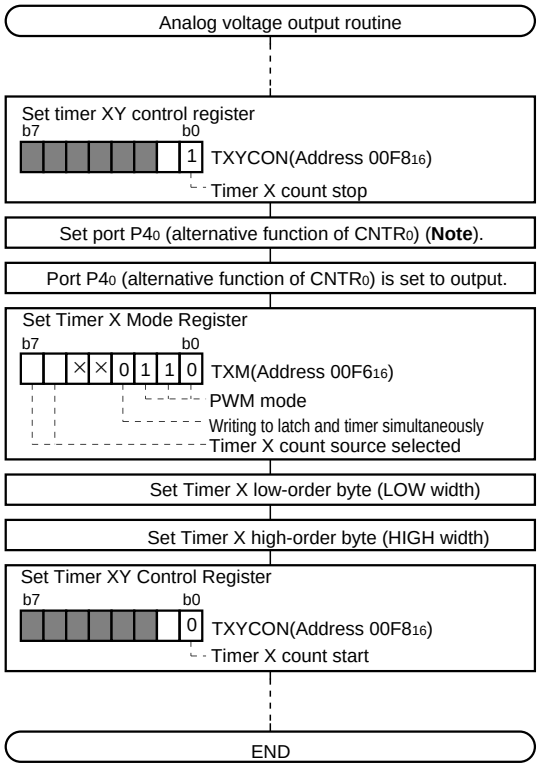


Figure 2.2.20 Peripheral Circuit Example



Note: The output analog voltage is initialized (V_{SS} or V_{CC}).
- The same potential as V_{SS} to be output : $P4_0$/$CNTR_0$ pin is set to '0'.
- The same potential as V_{CC} to be output : $P4_0$/$CNTR_0$ pin is set to '1'.

Figure 2.2.21 Control Procedure Example of Analog Voltage Output

2.3 Serial I/O

2.3.1 Application Example of Clock Synchronous Serial I/O Transmission

Successive Transmission

POINT: Successive transmission is performed by generating a serial I/O transmit interrupt when the transmit buffer register is emptied, as well as by generating a serial I/O transmit interrupt request when serial I/O transmission is initialized to 'enable' by using the serial I/O control register (Note).

Note: Refer to **Using Serial I/O Transmit Interrupt and Serial I/O Receive Interrupt** of **(5) Notes on Usage of Clock Synchronous Serial I/O** in **Section 1.14.2 Clock Synchronous Serial I/O.**

SPECIFICATIONS: ●Clock: f(XIN) = 7.9872 MHz

●5-byte successive transmission using clock synchronous serial I/O

- Baud rate: 2400 bps
- Synchronous clock: a frequency of 2.4 kHz, obtained from dividing f(XIN) is output from the SCLK pin.
- The completion of communication preparation at the receiver is recognized using port pin P17 as the $\overline{\text{SRDY}}$ signal input pin.

Figure 2.3.1 shows a connections example. Figure 2.3.2 shows a setting example of the synchronous clock.
Figure 2.3.3 shows the timing of interrupt control. Figure 2.3.4 shows a control procedure example of serial I/O transmit.

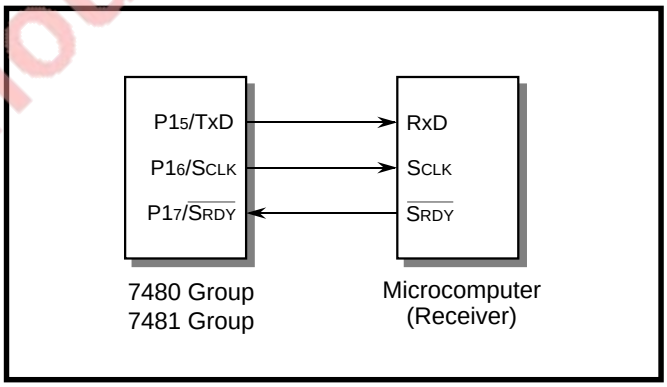


Figure 2.3.1 Connection Example

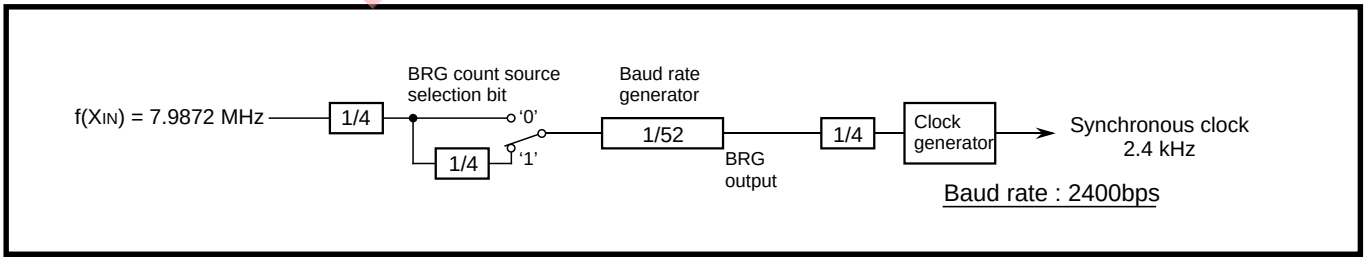


Figure 2.3.2 Setting Example of Synchronous Clock

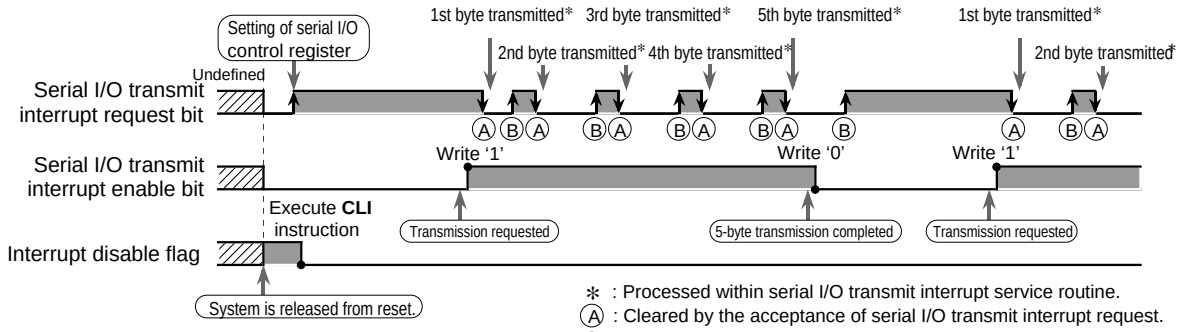
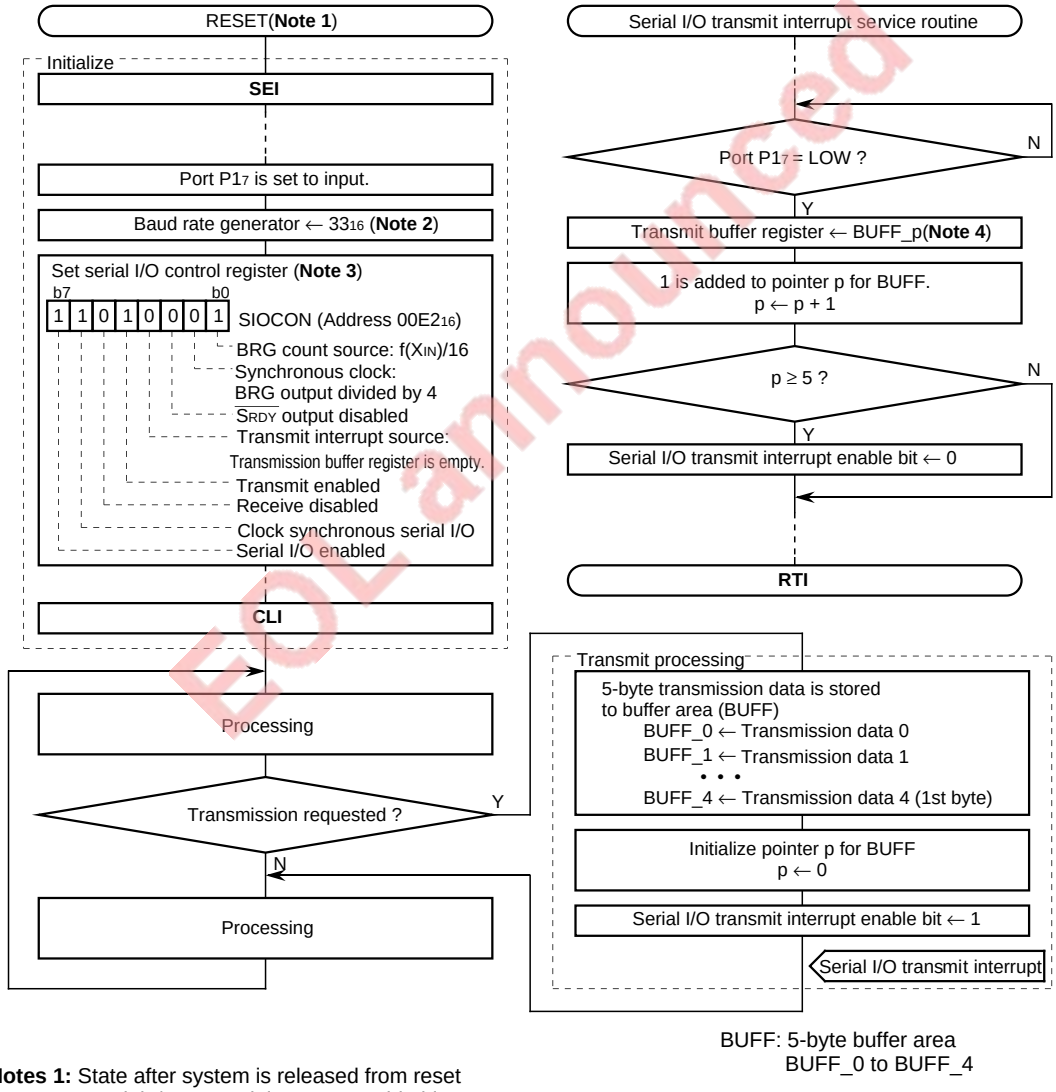


Figure 2.3.3 Timing of Interrupt Control



Notes 1: State after system is released from reset
- Serial I/O transmit interrupt enable bit = 0
- Serial I/O receive interrupt enable bit = 0

2: 33_{16} = 52–1
3: In this time, serial I/O transmit interrupt request bit is 1.
4: Transmit/receive is started by writing to transmit buffer register.

Figure 2.3.4 Control Procedure Example of Serial I/O Transmit

2.3.2 Application Example of Clock Asynchronous Serial I/O (UART) Reception

Processing of Received Data Bytes as a Packet

POINT: RAM area is secured by adding the several bytes to the maximum number of bytes necessary for data processing, and the received data is stored in increasing order of address in the interrupt service routine. If the data overflows the RAM area, the overflow data is stored at the start address of the RAM.
When the received data whose byte number satisfies the requirement of data processing is stored completely in the buffer area, the data processing is performed in the main routine. As a result, the received data can be stored without losing any bits of data in process even when the subsequent received data is stored completely during the data processing.

SPECIFICATIONS: ●Clock: f(XIN) = 7.9872 MHz
●UART reception
- Baud rate : 9600 bps
- Synchronous clock : f(XIN) is divided into 9.6 kHz
- Communication format: 1ST-8DATA-1SP

●Processing received data as a packet
The head data of every packet consists of the code characteristic for the head data and the code indicating the number of bytes of the packet.

Figure 2.3.5 shows a connection example.
Figure 2.3.6 shows the setting example of the synchronous clock.
Figure 2.3.7 shows a communication format.
Figure 2.3.8 shows a control procedure example of serial I/O receive.

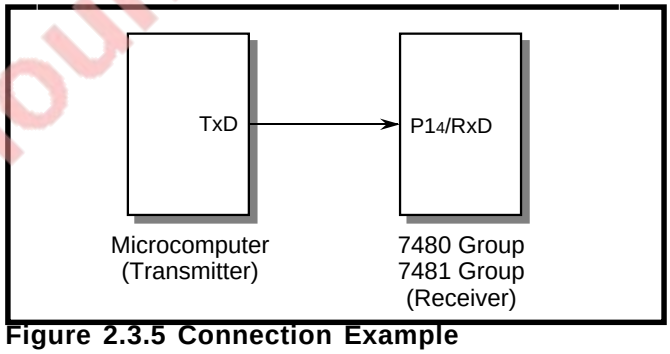


Figure 2.3.5 Connection Example

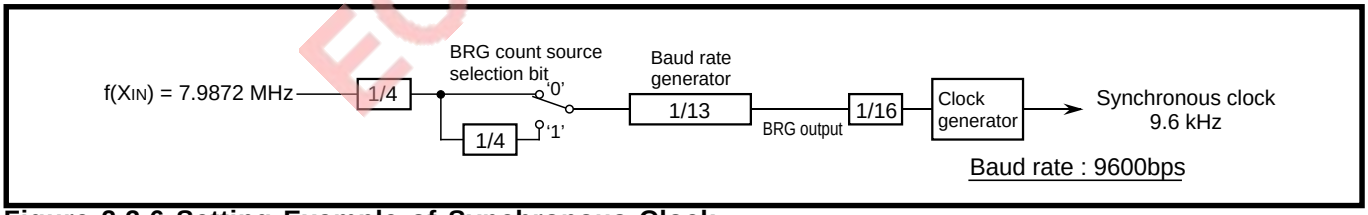


Figure 2.3.6 Setting Example of Synchronous Clock

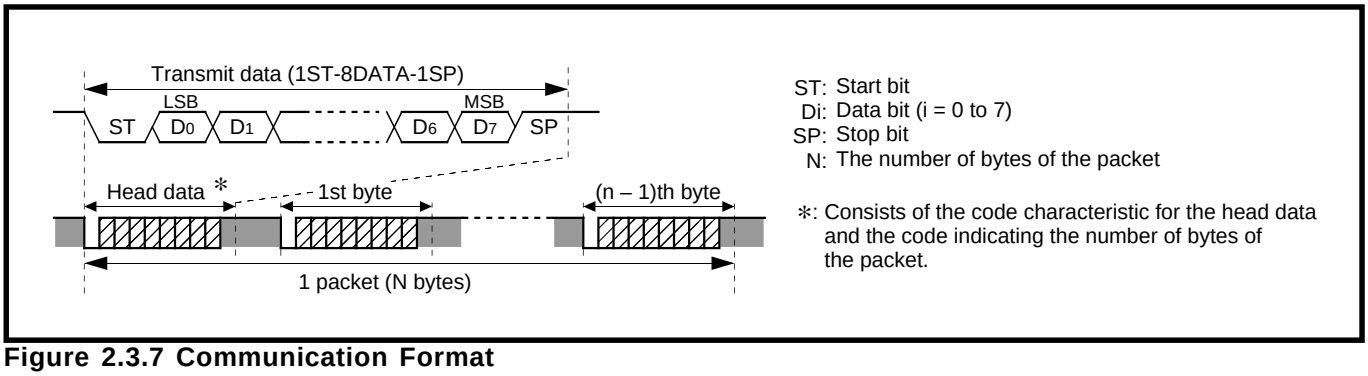
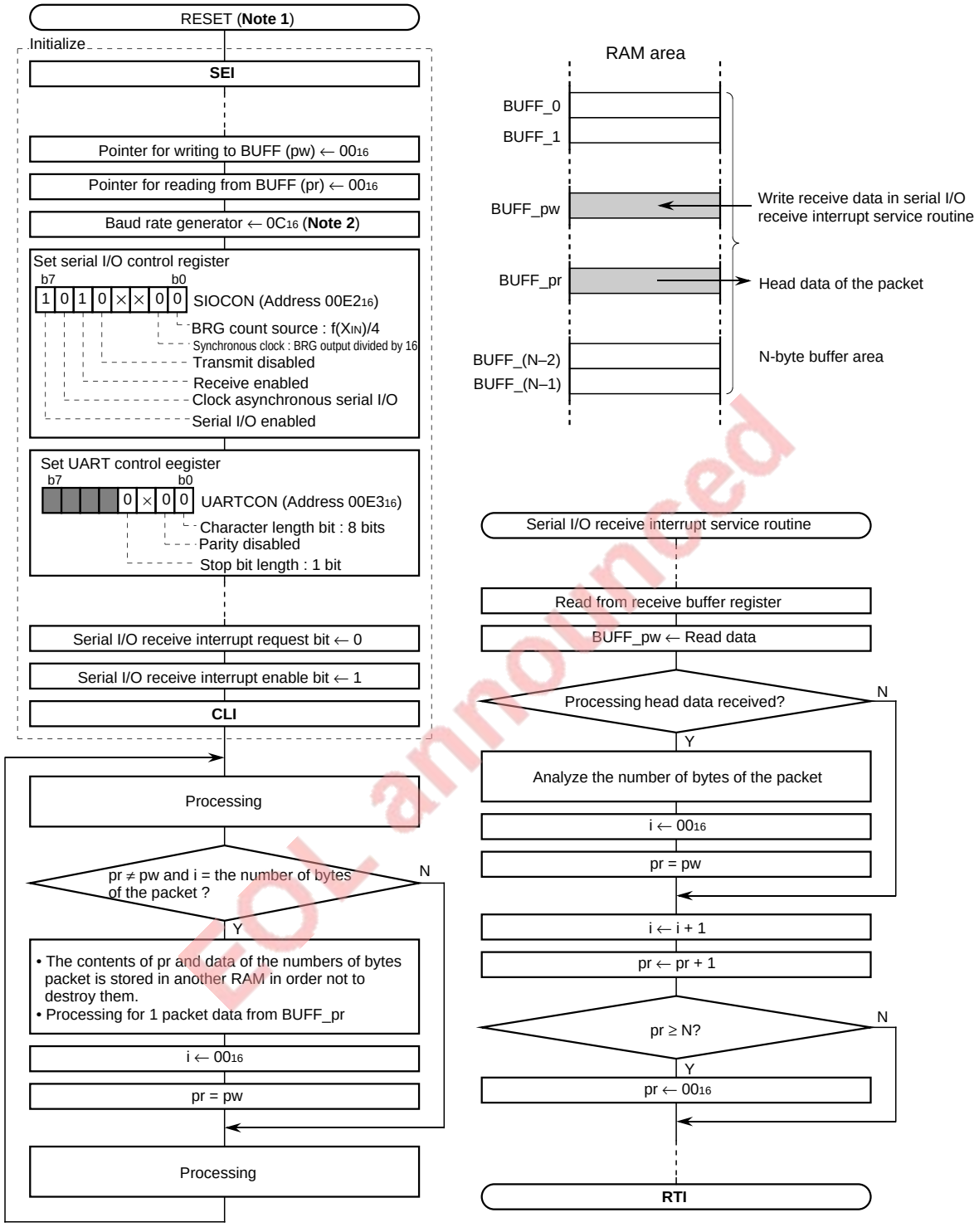


Figure 2.3.7 Communication Format



Notes 1: Processing after system is released from reset.
•Serial I/O receive interrupt enable bit = 0 (disabled)
2: 0C₁₆ = 13–1

Figure 2.3.8 Control Procedure Example of Serial I/O Receive

2.3.3 Application Example of Bus Arbitration Interrupt

LAN Communications in Contention Bus System

POINT: In LAN communications with the contention bus system, the malfunction of transmission due to bus collision is detected with a bus arbitration interrupt.

SPECIFICATIONS:
- Clock: f(XIN) = 7.9872 MHz
- LAN communication format: Simplified SAE J1850 (PWM system)
- The CNTR0 pin is connected to the RxD pin, and SOF is detected with the rising edge of a CNTR0 pin input
- Data is transmitted and received by clock synchronous serial I/O communications.
  - Baud rate: 41600 bps
  - Synchronous clock: f(XIN) is divided into 41.6 kHz
  - Bus collision detected
- The HIGH level has priority on LAN communication line at bus collision

Figure 2.3.9 shows a connection example.
Figure 2.3.10 shows a setting example of the synchronous clock.
Figure 2.3.11 shows a communication format example of simple SAE J1850.
Figure 2.3.12 shows a communication timing example.
Figures 2.3.13 and 2.3.14 show control procedure examples.

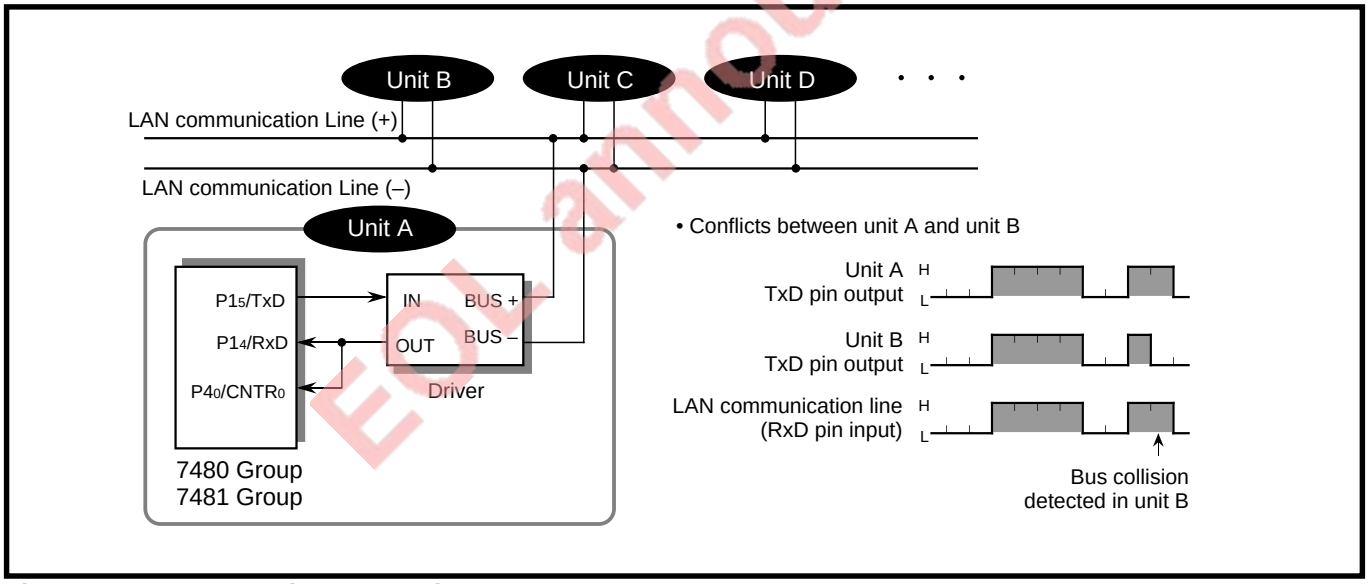


Figure 2.3.9 Connection Example

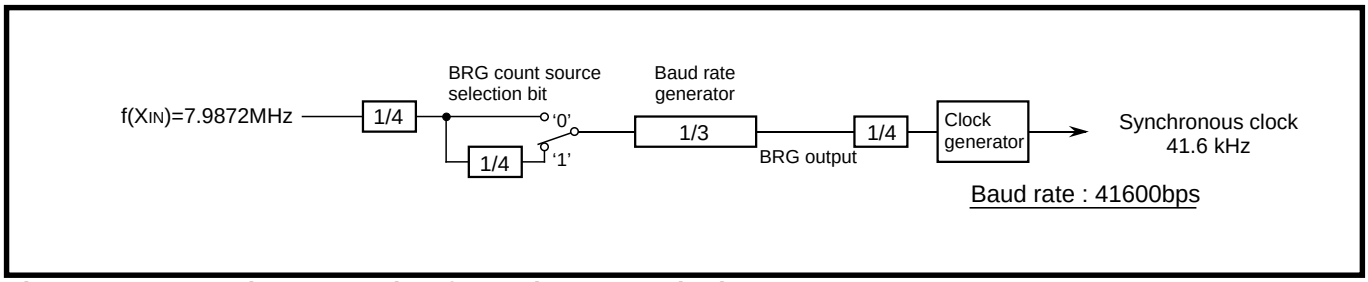
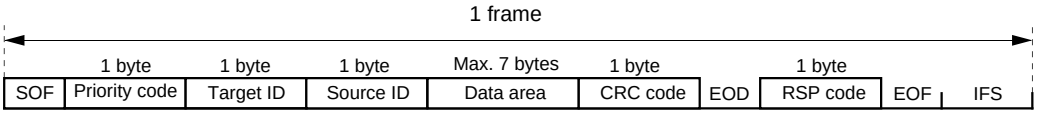


Figure 2.3.10 Setting Example of Synchronous Clock



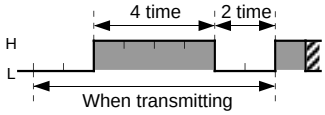
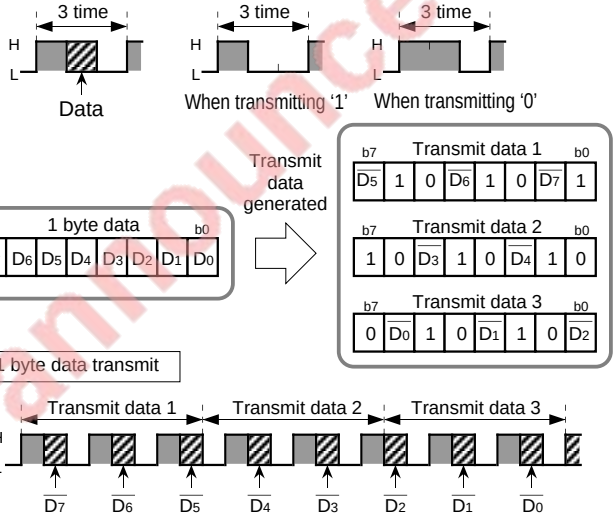
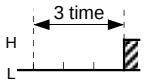
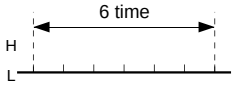
Transmit/Receive Data Name	Function	Transmission/Reception
SOF (Start of Frame)	Indicates the head of the frame	• Fixed format of 6 time. • Transmitter transmits '00111100_2' with serial transmission. • Receiver detects data with $CNTR_0$ interrupt and recognizes with pulse width measurement mode of timer X. 
Priority Code	Code for priority control in multi-unit transmission	• 1 bit represented by 3 time (1 byte represented by 3 bytes). • Both transmitter and receiver communicate with serial I/O. • Priority code: '00_{16}' (high priority) through 'FF_{16}' (low priority). 
Target ID	ID number of target unit	
Source ID	ID number of transmit unit	
Data Area	Transmit data	
CRC Code	Code for error detection	
RSP Code	Receiver transmits self-address if data is correctly received on error detection	
EOD (End of Data)	Indicates the end of data	• Fixed format of 3 time. • Both transmitter and receiver generate a wait time with timer 1.
EOF (End of Frame)	Indicates the end of frame	
IFS (Inter-Frame Separation)	Indicates the separation between frames	• Fixed format of 6 time. • Both transmitter and receiver generate a wait time with timer 1. 

Figure 2.3.11 Communication Format Example of Simplified SAE J1850

• When unit A, unit B, and unit C start transmitting simultaneously

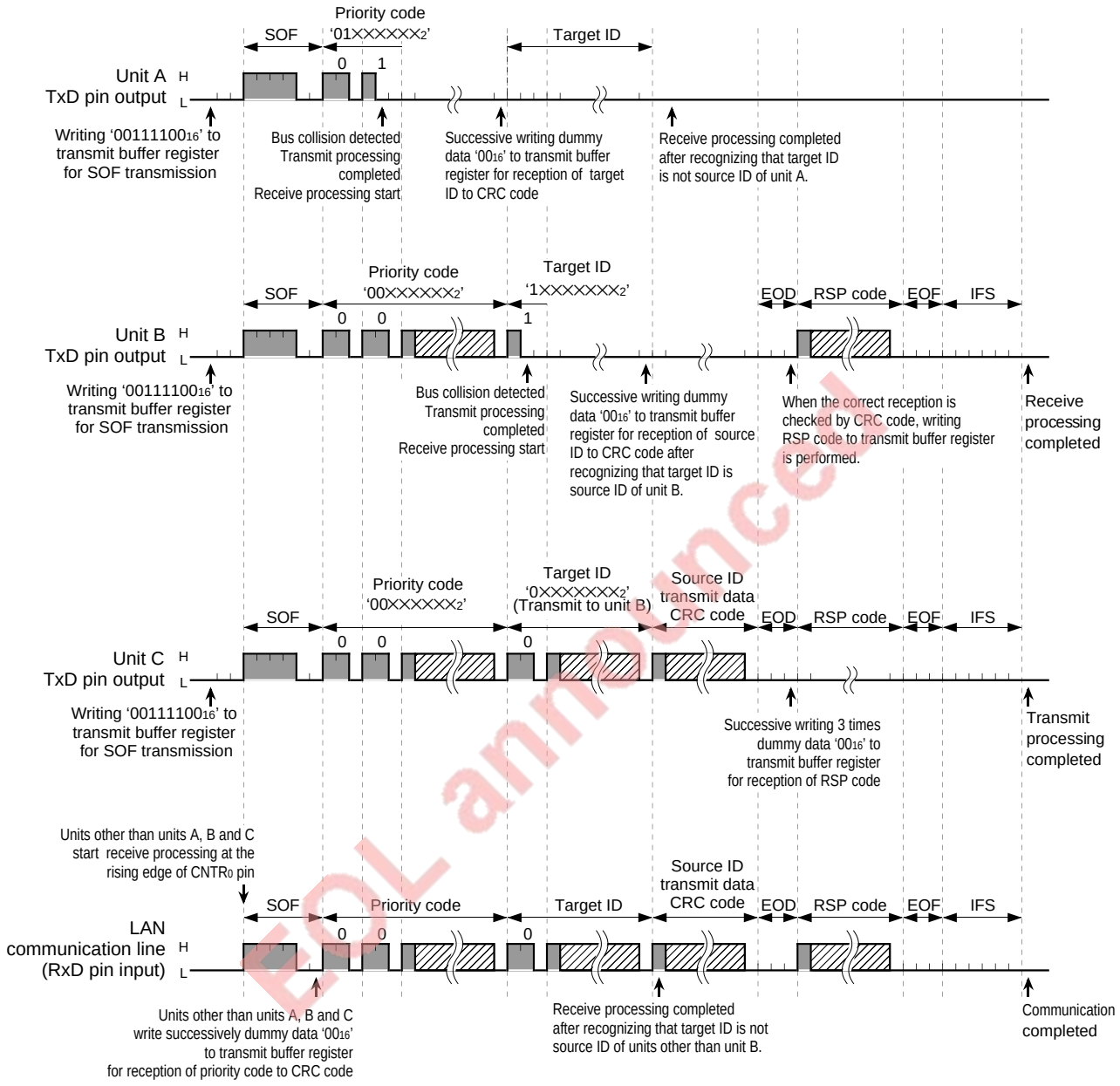


Figure 2.3.12 Communication Timing Example

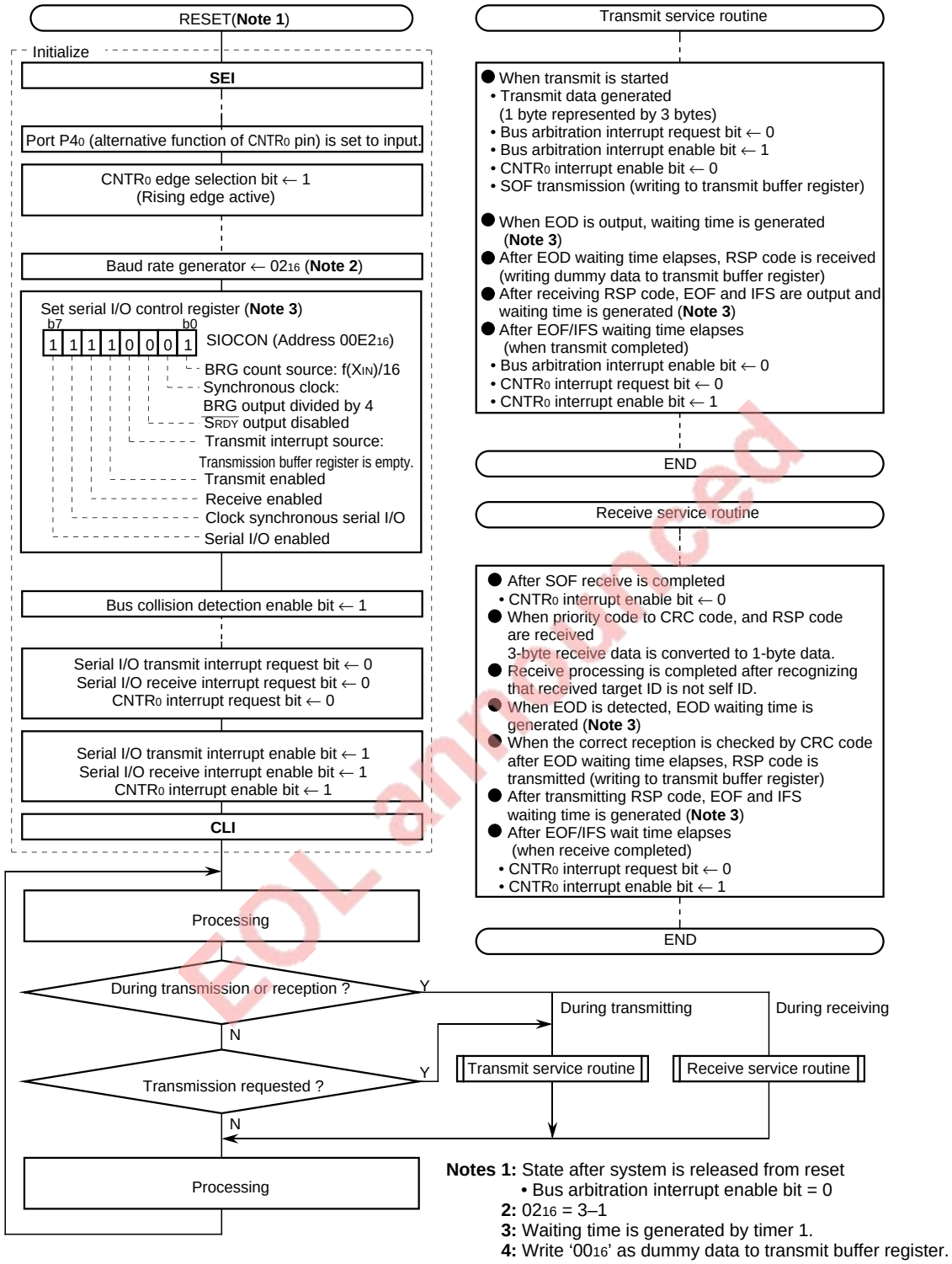
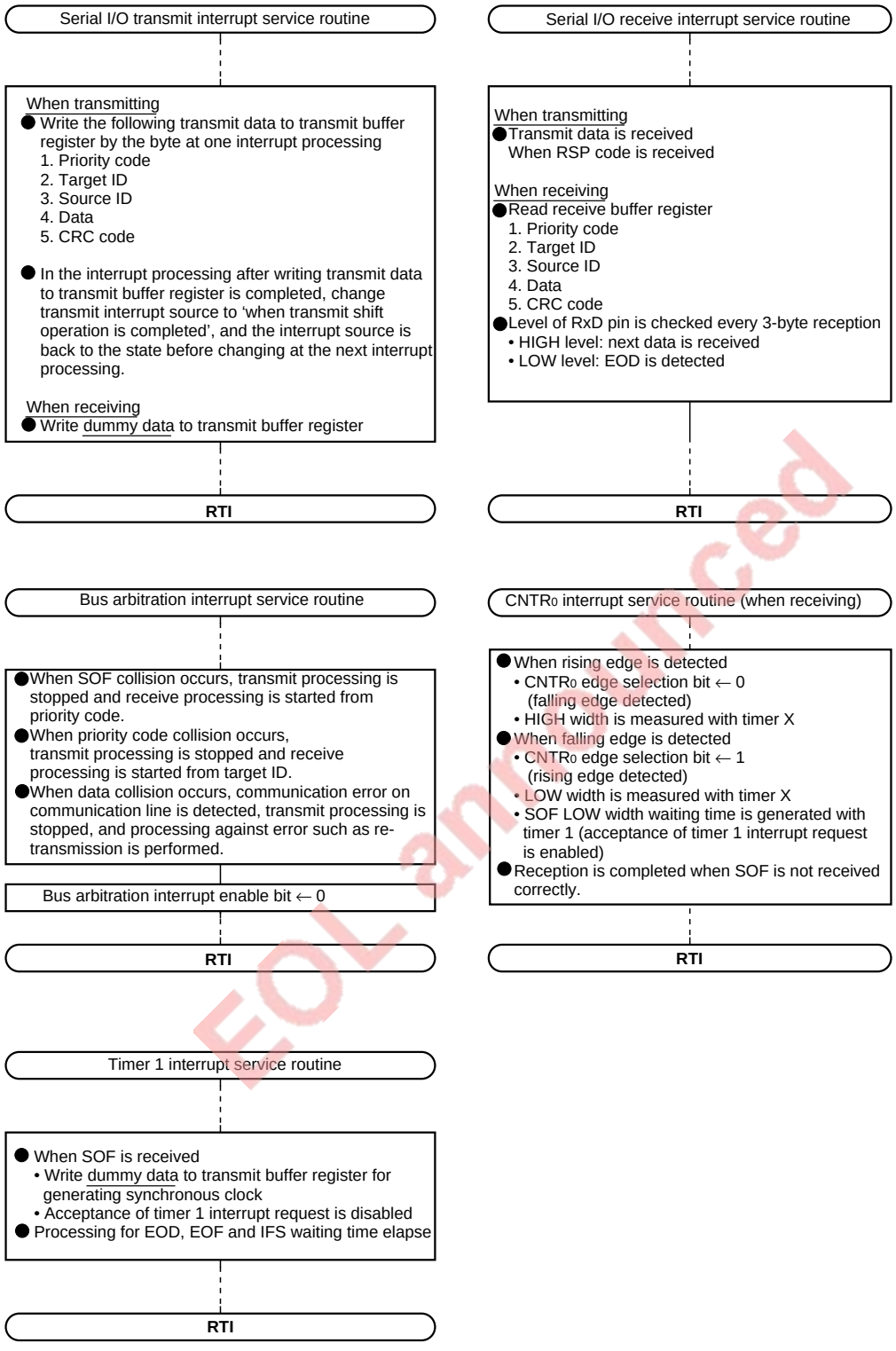


Figure 2.3.13 Control Procedure Example (1) of LAN Communications

2.3 Serial I/O



Note: Write '00_{16}' as dummy data to transmit buffer register.

Figure 2.3.14 Control Procedure Example (2) of LAN Communications

2.4 A-D Converter

2.4.1 Determination of A-D Conversion Values

In A-D conversion, it is recommended to determine conversion values using several samplings to improve the accuracy of A-D conversion. Methods for sampling and determining A-D conversion values are described below.

Sampling Methods

EXAMPLES: ① Sampling 2^n times
② Running sampling 2^n times
③ Sampling $(2^n + 2)$ times
Notes 1: 'n' is a positive integer according to the specifications.

Determining Methods

EXAMPLES: ① The sum of the sampling result is divided by the number of times of the sampling.
② Except the minimum and the maximum value, the sum of the sampling (or running sampling) results of $(2^n + 2)$ times is divided by 2^n.
③ The average value calculated by ① or ② is updated unless the difference between the previous and the newest value is 'm' or more.
Notes 2: 'm' and 'n' are positive integers according to the specifications.

A method derived from these examples of sampling and determining is explained in **Section 2.4.2 Application Example of A-D Converter**.

2.4.2 Application Example of A-D Converter

POINT: To improve the accuracy of A-D conversion, A-D conversion values are determined by **Sampling Methods** ② and ③, and **Determining Methods** ② and ③ of **Section 2.4.1 Determination of A-D Conversion Values**.

SPECIFICATIONS: After the running sampling has been taken 6 times, the sum of the sampling results, except the minimum and maximum values, is divided by 4.
When the difference between the new average value and the previous updated value is less than 5, the value is updated to the new value, when 5 or more, the value is not updated.

Figure 2.4.1 shows the example of determining A-D conversion values.
Figure 2.4.2 shows the control procedure example of determining of A-D conversion values.

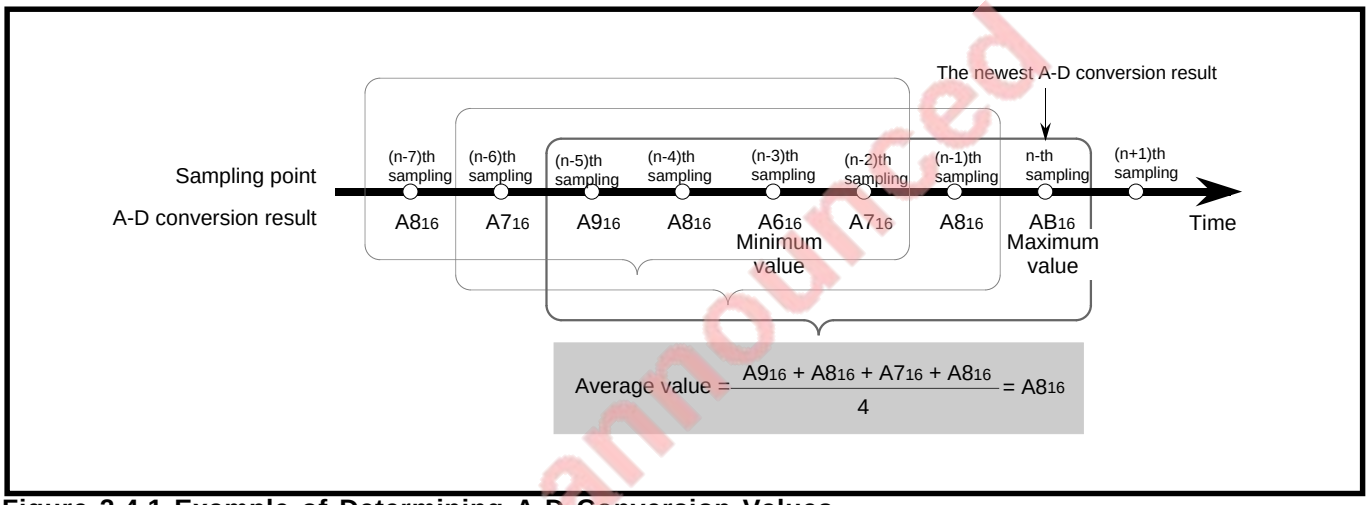
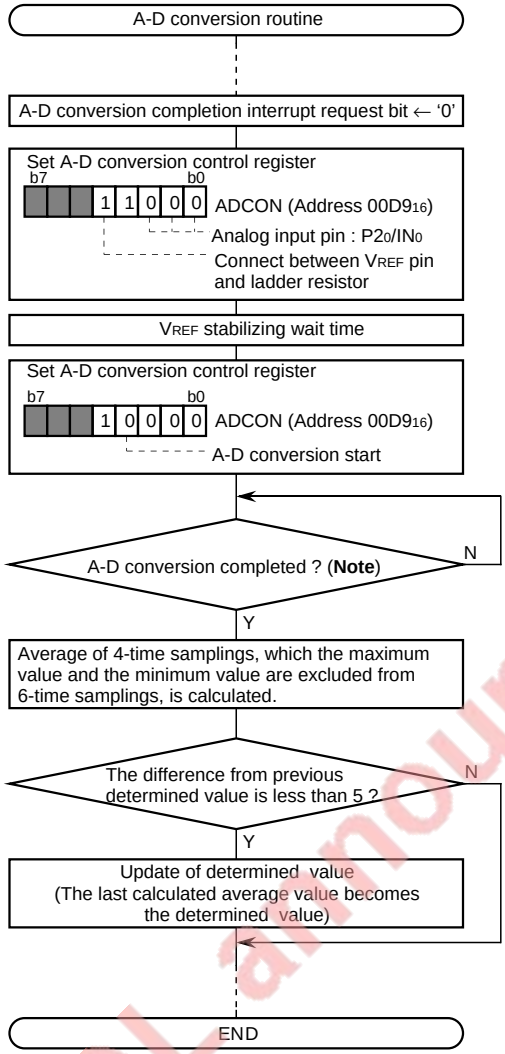


Figure 2.4.1 Example of Determining A-D Conversion Values

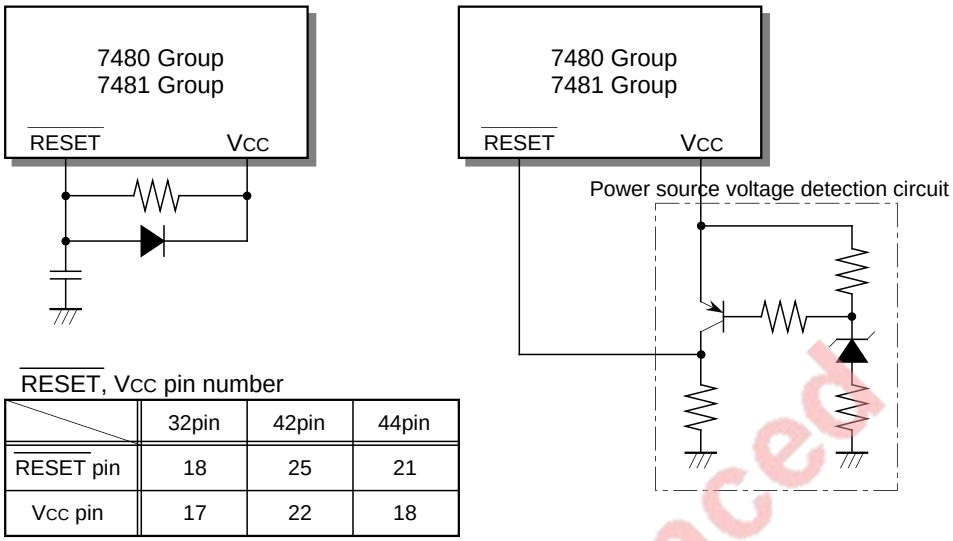


Note: The completion of A-D conversion is examined by the following.
- A-D conversion completion bit of A-D conversion control register is '1'.
- A-D conversion completion interrupt request bit of interrupt request register 1 is '1'.
- Branch to A-D conversion completion interrupt service routine is performed. (When A-D conversion completion interrupt is enabled)

Figure 2.4.2 Control Procedure Example of Determining A-D Conversion Values

2.5 Reset

Figure 2.5.1 shows reset circuit examples.



RESET, Vcc pin number

	32pin	42pin	44pin
RESET pin	18	25	21
Vcc pin	17	22	18

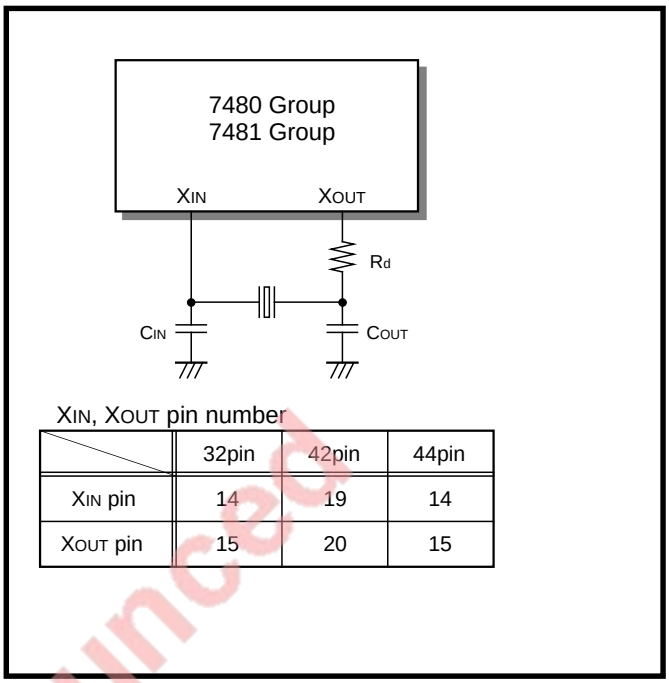
Figure 2.5.1 Reset Circuit Examples

2.6 Oscillation Circuit

2.6.1 Oscillation Circuit with Ceramic Resonator

An oscillation circuit can be formed by connecting a ceramic resonator between the XIN and XOUT pins. Figure 2.6.1 shows an oscillation circuit example with a ceramic resonator.

Note: Set oscillation circuit parameters, such as Rd, CIN, and COUT, to the values recommended by the manufacturer of the resonator.



XIN, XOUT pin number

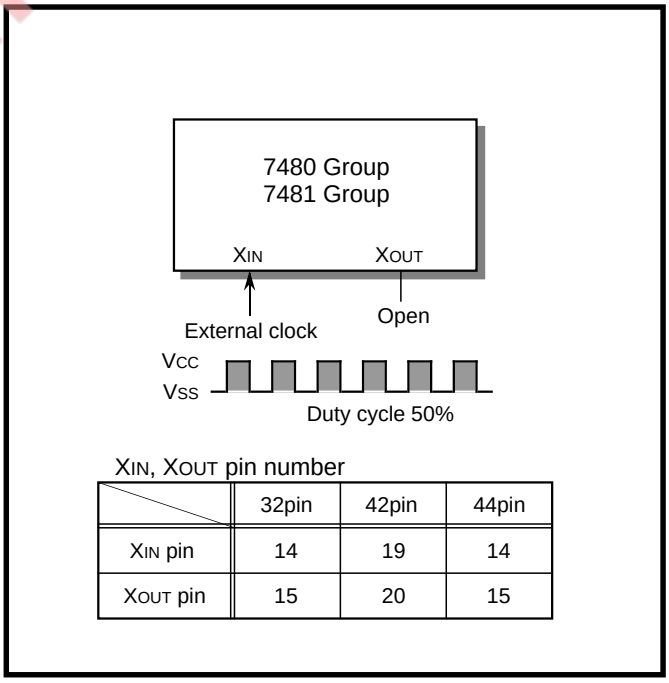
	32pin	42pin	44pin
XIN pin	14	19	14
XOUT pin	15	20	15

Figure 2.6.1 Oscillation Circuit Example with Ceramic Resonator

2.6.2 External Clock Input to XIN

An external clock input to the XIN pin can be supplied to the built-in clock generator.
Figure 2.6.2 shows the external clock circuit example.

Notes 1: Leave the XOUT pin open when an external clock is input to the XIN pin.
2: Use a 50% duty cycle pulse signal as the external clock input to the XIN pin.



XIN, XOUT pin number

	32pin	42pin	44pin
XIN pin	14	19	14
XOUT pin	15	20	15

Figure 2.6.2 External Clock Circuit Example

2.7 Power-Saving Function

2.7.1 Application Example of Stop Mode

Power-Saving in Key-Input Waiting State

POINT: When the CPU has no key input for the specified period in its key-input waiting state, the CPU enters the stop mode and reduces power dissipation by halting itself and its peripherals. Any key input, thereafter, generates a key-on wakeup interrupt request, and the CPU returns to the normal mode by accepting the request.

SPECIFICATIONS:
- Port pin $P0_0$ is used as a key-input pin.
- When having no key input for the specified period, the CPU executes the **STP** instruction to enter the stop mode.
- Any key input generates a key-on wakeup interrupt request in the stop mode, and the CPU returns to the normal mode by accepting the request.

Figure 2.7.1 shows a connection example.
Figure 2.7.2 shows an operation example in the key-input waiting state.
Figure 2.7.3 shows a control procedure example of power-saving in key-input waiting state.

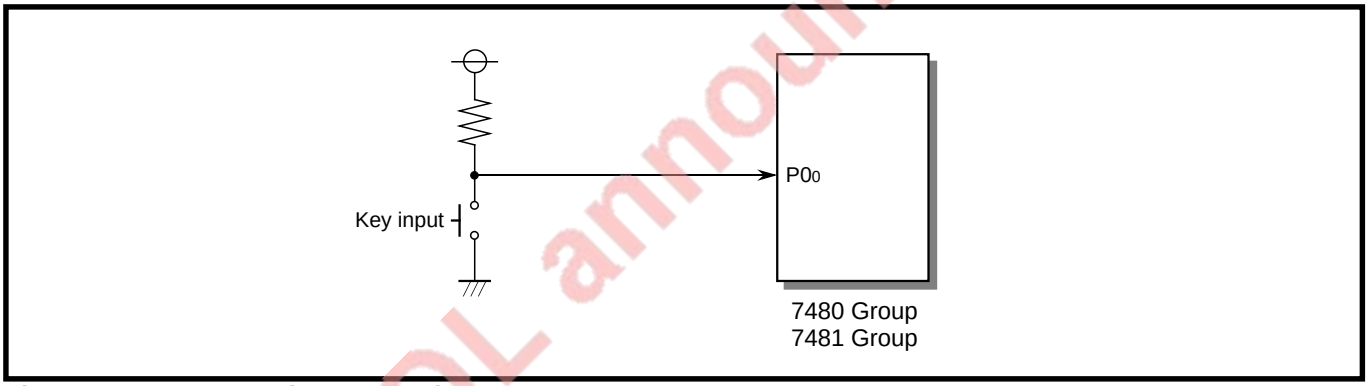


Figure 2.7.1 Connection Example

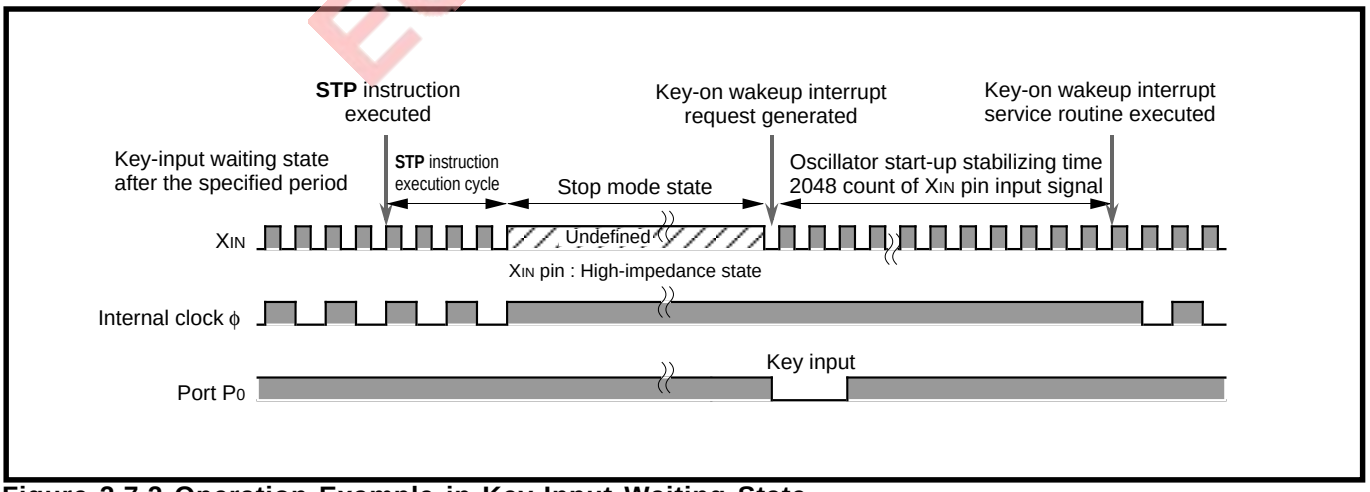


Figure 2.7.2 Operation Example in Key-Input Waiting State

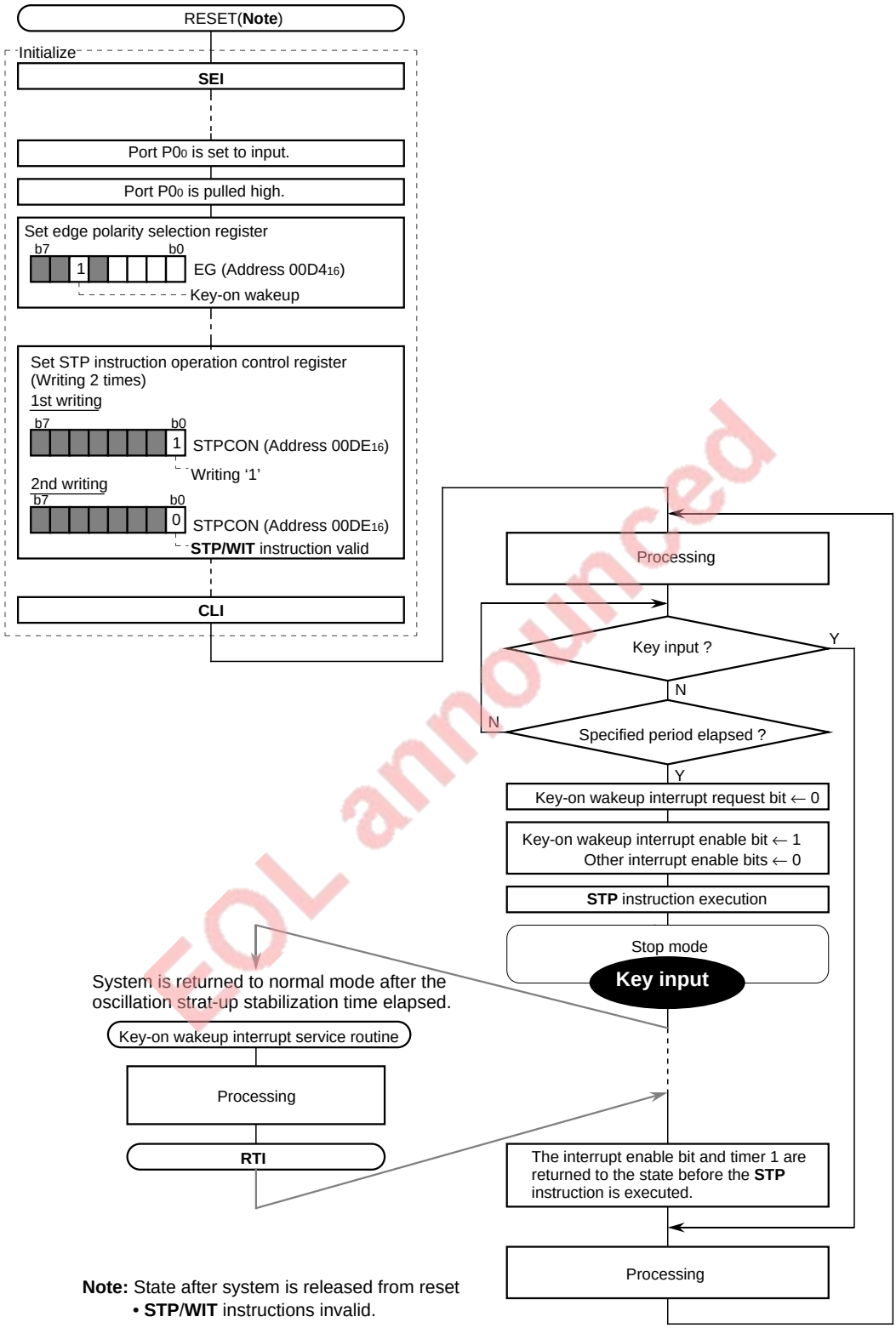


Figure 2.7.3 Control Procedure Example of Power-Saving in Key-Input Waiting State

2.7.2 Application Example of Wait Mode

Power-Saving in Serial I/O Receive Waiting State

POINT: When serial I/O reception is not started in serial I/O receive enabled state, the CPU enters the wait mode and reduces power dissipation by halting itself. Then, when a serial I/O receive interrupt or a timer X interrupt is accepted after the specified period, and the CPU returns to the normal mode and terminates the communications.

SPECIFICATIONS:
- Clock synchronous serial I/O reception
  - Synchronous clock: external clock input
  - $\overline{\text{SRDY}}$ signal output
- When serial I/O reception is not started in serial I/O receive enabled state, the CPU executes the **WIT** instruction to enter the wait mode.
- The CPU returns to the normal mode and terminates communications by either of the acceptance of the following interrupt sources:
  - Serial I/O receive interrupt
  - Timer X interrupt: receive wait time is counted in the timer mode

Figure 2.7.4 shows a connection example.
Figure 2.7.5 shows an operation example in the serial I/O receive waiting state.
Figure 2.7.6 shows a control procedure of power-saving.

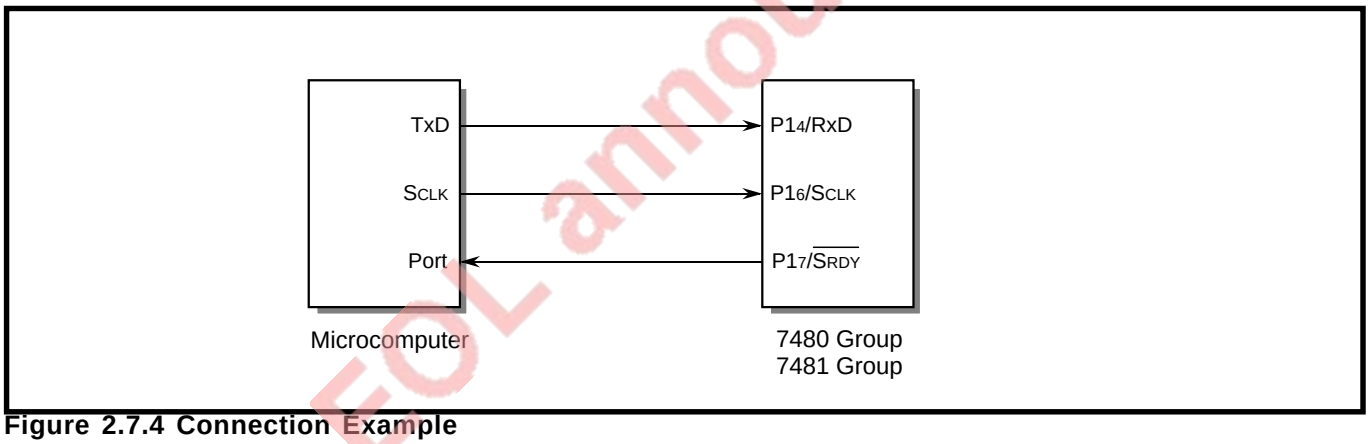


Figure 2.7.4 Connection Example

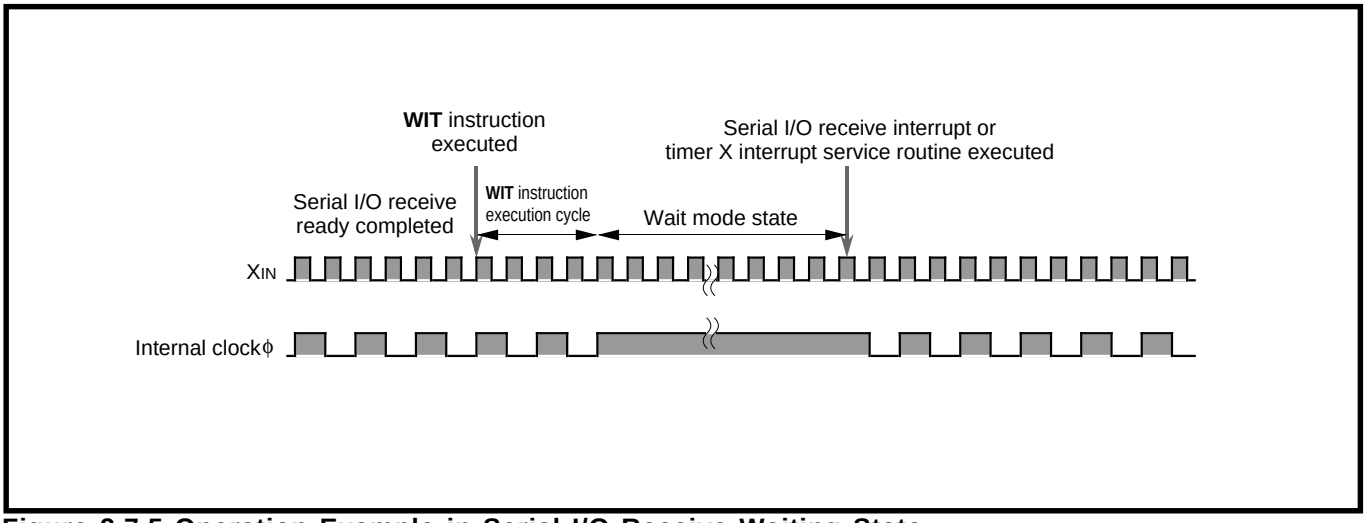
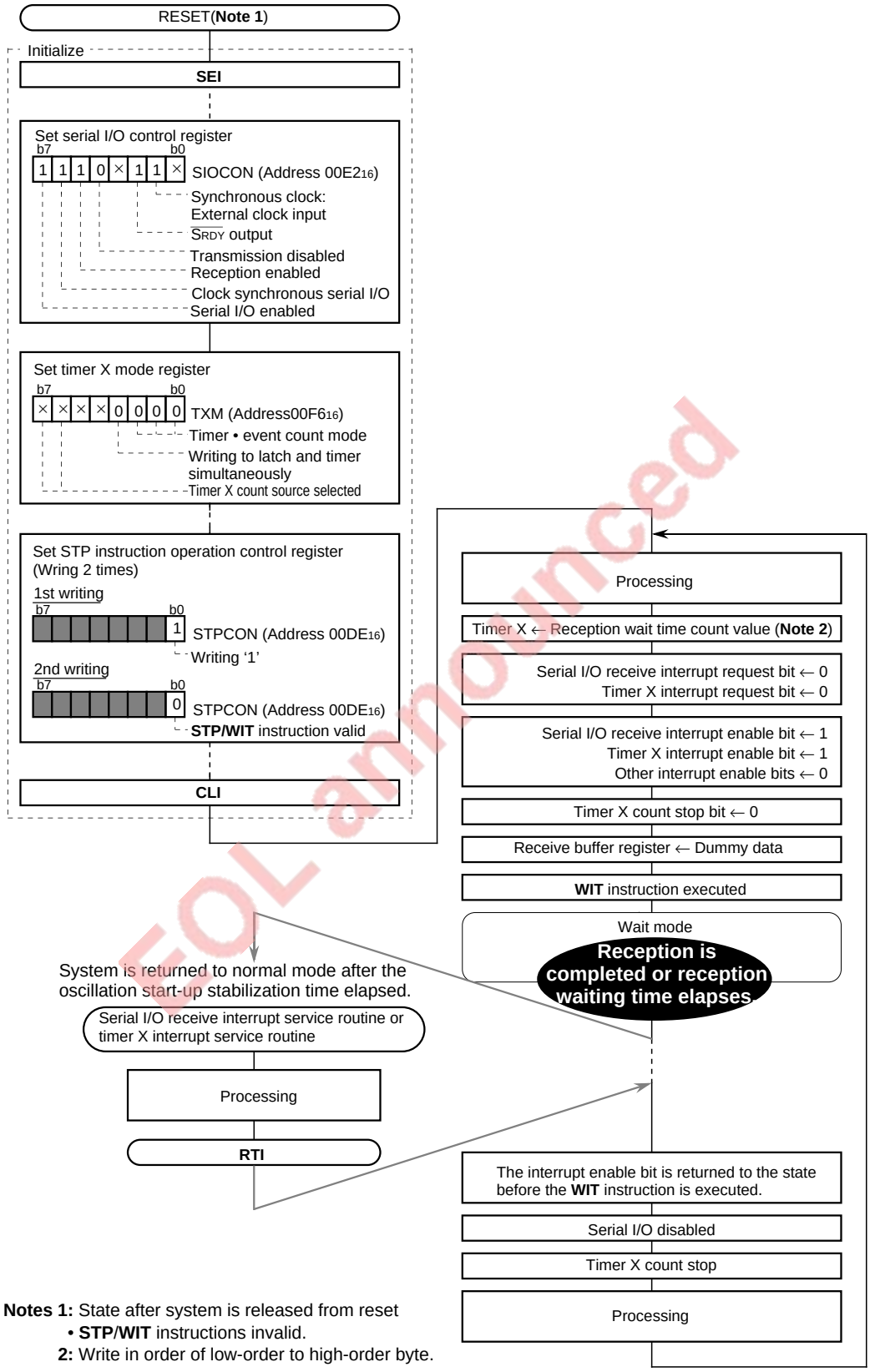


Figure 2.7.5 Operation Example in Serial I/O Receive Waiting State



Notes 1: State after system is released from reset
• **STP**/**WIT** instructions invalid.
2: Write in order of low-order to high-order byte.

Figure 2.7.6 Control Procedure Example of Power-Saving

2.8 Countermeasures against Noise

Countermeasures against noise are described below. The following countermeasures are effective against noise in theory, however, it is necessary not only to take measures as follows but to evaluate before actual use.

2.8.1 Shortest Wiring Length

The wiring on a printed circuit board can be as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

(1) Wiring for $\overline{\text{RESET}}$ Pin

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

REASON

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

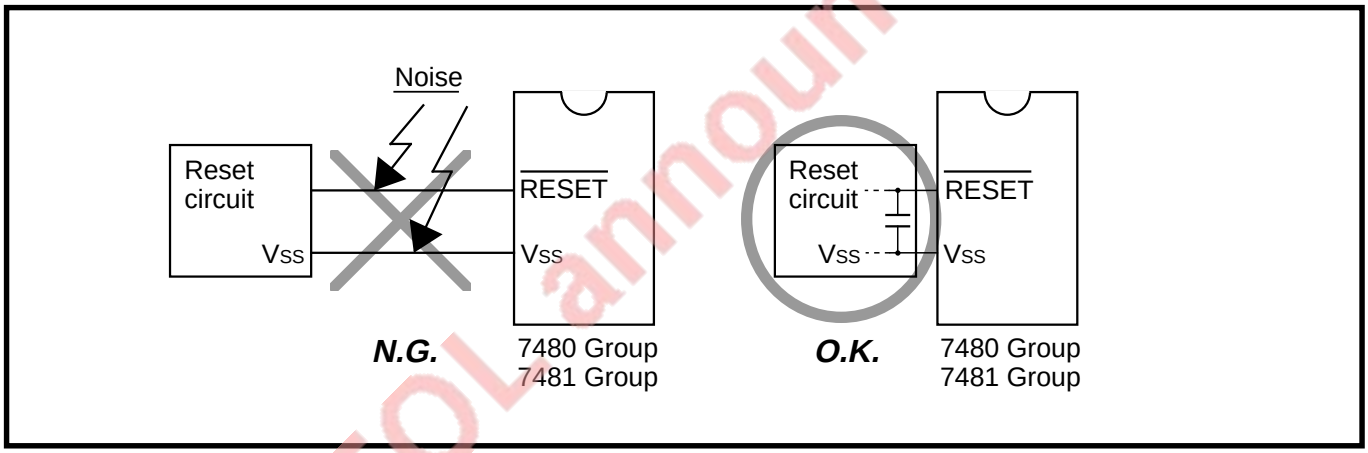


Figure 2.8.1 Wiring for $\overline{\text{RESET}}$ Pin

(2) Wiring for Clock I/O Pins

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

REASON

A microcomputer's operation synchronizes with a clock generated by the oscillator (circuit). If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a malfunction or program runaway.
Also, if a potential difference is caused by the noise between the VSS level of a microcomputer and the VSS level of an oscillator, the correct clock will not be input in the microcomputer.

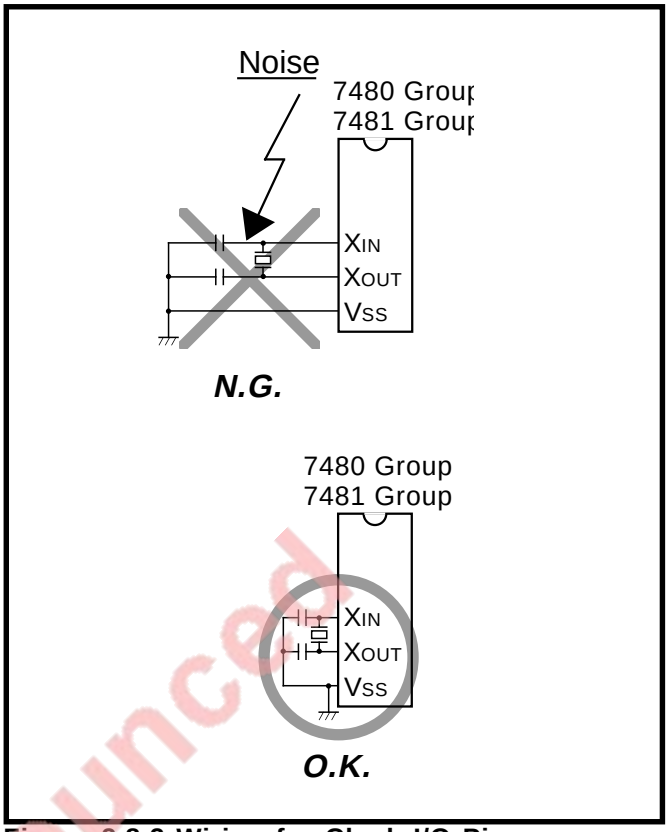


Figure 2.8.2 Wiring for Clock I/O Pins

(3) Wiring for VPP Pin of One Time PROM Version and EPROM Version

(In the 7480 Group and 7481 Group, the VPP pin is also used as the P33 pin)
Connect an approximately 5 kΩ resistor to the VPP pin the shortest possible in series.

Note: Even when a circuit which included an approximately 5 kΩ resistor is used in the Mask ROM version, the microcomputer operates correctly.

REASON

The VPP pin of the One Time PROM and the EPROM version is the power source input pin for the built-in PROM. When programming in the built-in PROM, the impedance of the VPP pin is low to allow the electric current for writing flow into the PROM. Because of this, noise can enter easily. If noise enters the VPP pin, abnormal instruction codes or data are read from the built-in PROM, which may cause a program runaway.

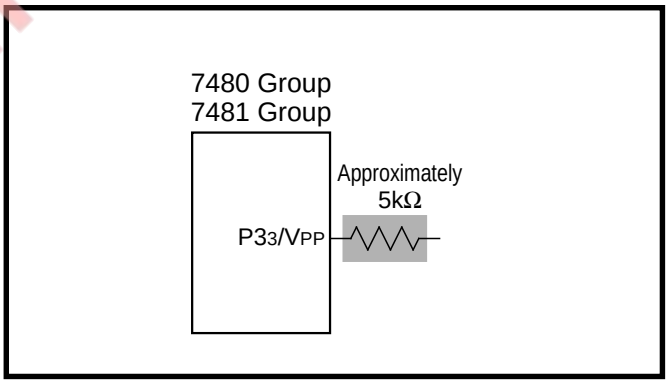


Figure 2.8.3 Wiring for VPP Pin of One Time PROM and EPROM Version

2.8 Countermeasures against Noise

2.8.2 Connection of Bypass Capacitor across Vss Line and Vcc Line

Connect an approximately 0.1 μF bypass capacitor across the Vss line and the Vcc line as follows:

- Connect a bypass capacitor across the Vss pin and the Vcc pin at equal length.
- Connect a bypass capacitor across the Vss pin and the Vcc pin with the shortest possible wiring.
- Use lines with a larger diameter than other signal lines for Vss line and Vcc line.
- Connect the power source line to Vss pin and Vcc pin through a bypass capacitor.

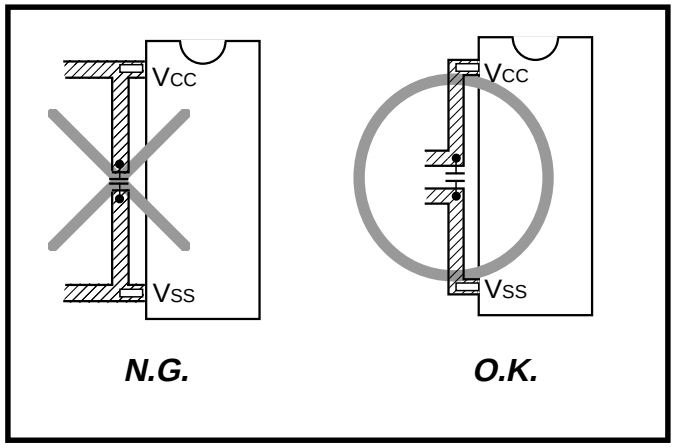


Figure 2.8.4 Bypass Capacitor across Vss Line and Vcc Line

2.8.3 Connection of Bypass Capacitor across Vss Line and VREF Line

Connect an approximately 0.01 μF bypass capacitor across the Vss line and the VREF line as follows:

- Connect a bypass capacitor across the Vss pin and the VREF pin at equal length.
- Connect a bypass capacitor across the Vss pin and the VREF pin with the shortest possible wiring.
- Use lines with a larger diameter than other signal lines for Vss line and VREF line.

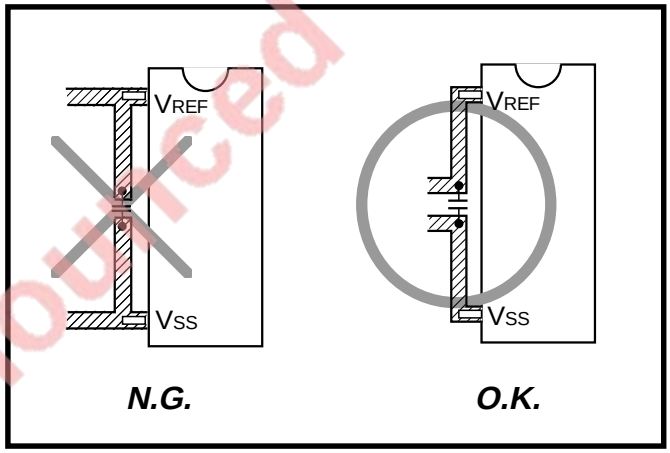


Figure 2.8.5 Bypass Capacitor across Vss Line and VREF Line

2.8.4 Wiring to Analog Input Pins

- Connect an approximately 100 Ω to 1 kΩ resistor to an analog signal line which is connected to an analog input pin in series. Besides, connect the resistor to the microcomputer as close as possible.
- Connect an approximately 0.1 to 1 μF capacitor across the Vss pin and the analog input pin. Besides, connect the capacitor to the Vss pin as close as possible. Also, connect the capacitor across the analog input pin and the Vss pin at equal length.

REASON

Signals which is input in an analog input pin (such as an A-D converter input pin) are usually output signals from sensor. The sensor which detects a change of event is installed far from the printed circuit board with a microcomputer, the wiring to an analog input pin is longer necessarily. This long wiring functions as an antenna which feeds noise into the microcomputer, which causes noise to an analog input pin.

If a capacitor between an analog input pin and the Vss pin is grounded at a position far away from the Vss pin, noise on the GND line may enter a microcomputer through the capacitor.

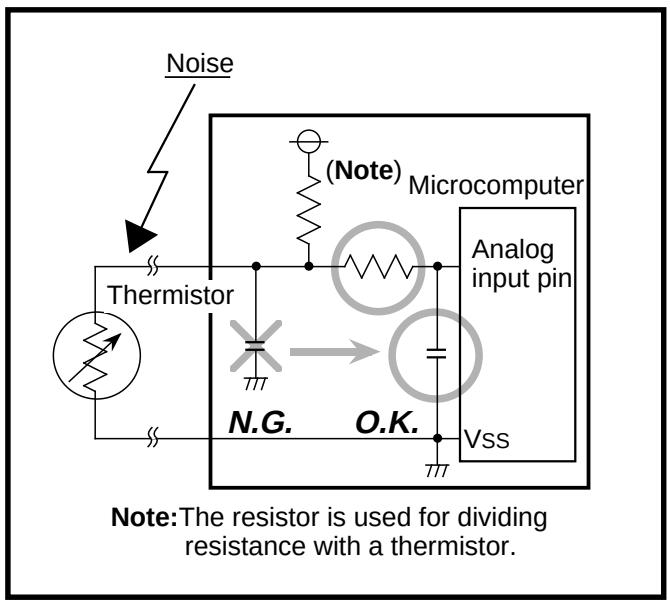


Figure 2.8.6 Analog Signal Line and Resistor and Capacitor

2.8.5 Consideration for Oscillator

Take care to prevent an oscillator that generates clocks for a microcomputer operation from being affected by other signals.

(1) Keeping Oscillator Away from Large Current Signal Lines

Install a microcomputer (and especially an oscillator) as far as possible from signal lines where a current larger than the tolerance of current value flows.

REASON

In the system using a microcomputer, there are signal lines for controlling motors, LEDs, and thermal heads or others. When a large current flows through those signal lines, strong noise occurs because of mutual inductance.

(2) Keeping Oscillator Away from Signal Lines Where Potential Levels Change Frequently

Install an oscillator and a connecting pattern of an oscillator away from signal lines where potential levels change frequently. Also, do not cross such signal lines over the clock lines or the signal lines which are sensitive to noise.

REASON

Signal lines where potential levels change frequently (such as the CNTR pin line) may affect other lines at signal rising or falling edge. If such lines cross over a clock line, clock waveforms may be deformed, which causes a microcomputer failure or a program runaway.

(3) Oscillator Protection Using Vss Pattern

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

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

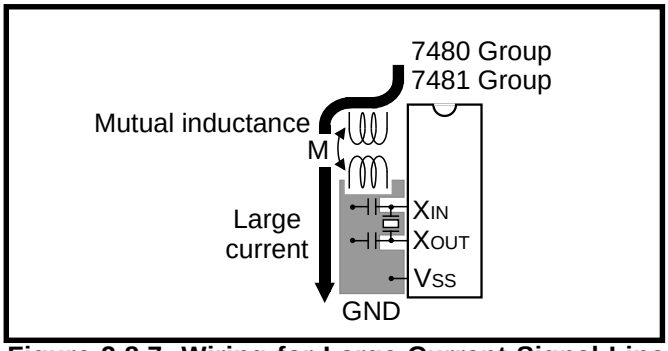


Figure 2.8.7 Wiring for Large Current Signal Line

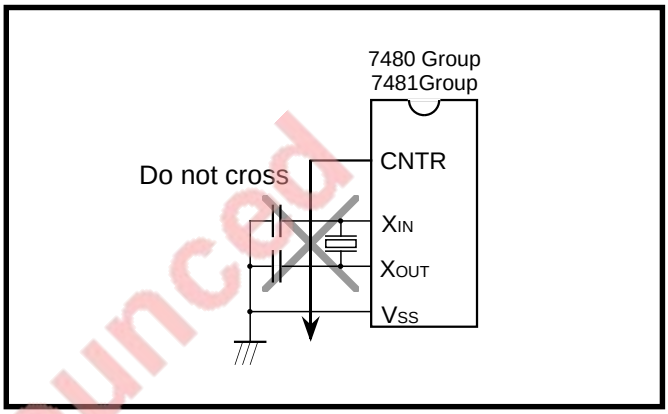


Figure 2.8.8 Wiring to Signal Line Where Potential Levels Change Frequently

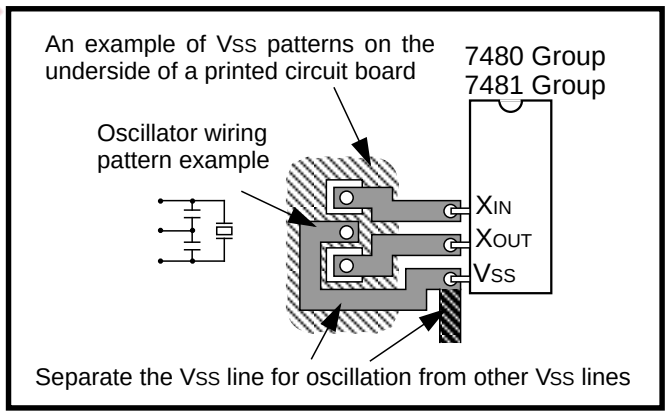


Figure 2.8.9 Vss Patterns on Underside of Oscillator

2.8 Countermeasures against Noise

2.8.6 Setup for I/O Ports

Setup I/O ports using hardware and software as follows:

<Hardware>

- Connect a resistor of 100 Ω or more to an I/O port in series.

<Software>

- As for an input port, read data several times by a program for checking whether input levels are equal or not.
- As for an output port, since the output data may reverse because of noise, rewrite data to its port latch at fixed periods.
- Rewrite data to direction registers, and if necessary, pull-up control registers and port P4P5 input control register at fixed periods.

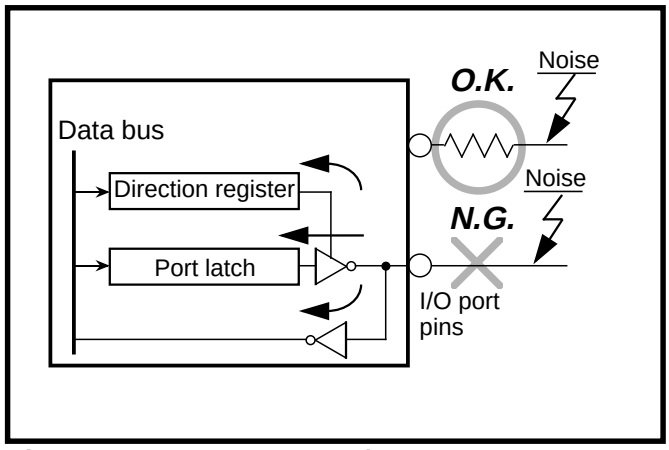


Figure 2.8.10 Setup For I/O Ports

2.8.7 Providing of Watchdog Timer Function by Software

If a microcomputer runs away because of noise or others, it can be detected by a software watchdog timer and the microcomputer can be reset to normal operation. This is equal to or more effective than program runaway detection by a hardware watchdog timer. The following shows an example of a watchdog timer provided by software.

In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt service routine and the interrupt service routine detects errors of the main routine.

This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

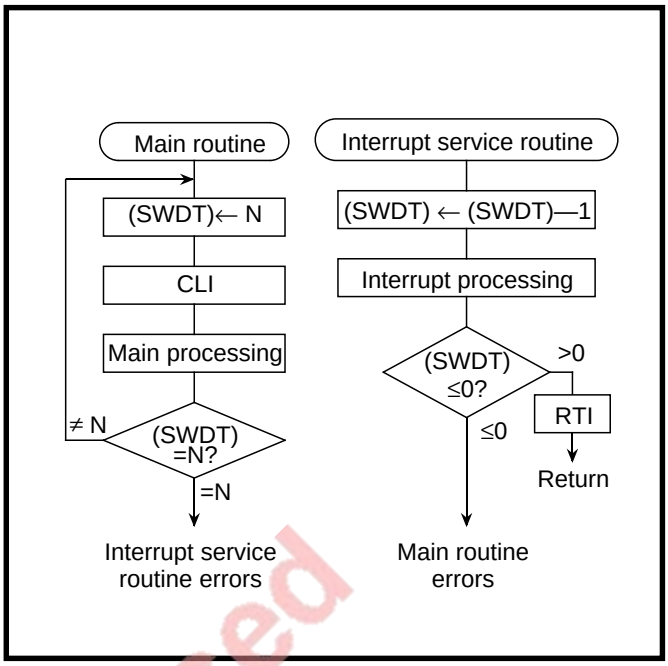


Figure 2.8.11 Watchdog Timer by Software

<Main Routine>

- Assigns a single byte of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:

  N+1 ≥ (Counts of interrupt processing executed in each main routine)

  As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.

- Watches the operation of the interrupt service routine by comparing the SWDT contents with counts of interrupt processing count after the initial value N has been set.
- Detects that the interrupt service routine has failed and determines to branch to the program initialization routine for recovery processing in the following cases:
  If the SWDT contents do not change after interrupt processing

<Interrupt Service Routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery when the contents of the SWDT reach 0 or less by continuative decrement without initializing to the initial value N.

2.9 Notes on Programming

2.9.1 Notes on Processor Status Register

■ **Initialization of Processor Status Register**

After system is released from reset, the contents of processor status register (PS) are undefined except for the I flag which is '1'. Therefore, flags which affect program execution must be initialized after system is released from reset.

In particular, it is essential to initialize the T and D flags because they have an important effect on calculations.

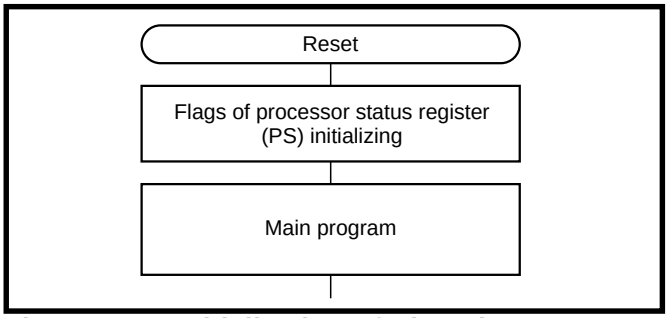


Figure 2.9.1 Initialization of Flags in PS

■ **How to Refer Processor Status Register**

To refer the contents of the processor status register (PS), execute the **PHP** instruction once then read the contents of (S + 1). If necessary, execute the **PLP** instruction to return the PS to its original status.

The **NOP** instruction should be executed after every **PLP** instruction.

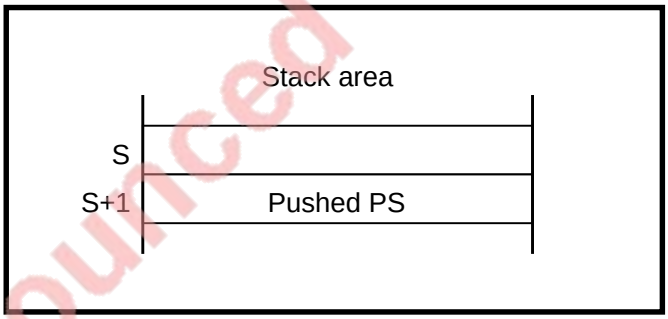


Figure 2.9.2 Stack Memory Contents after PHP Instruction Execution

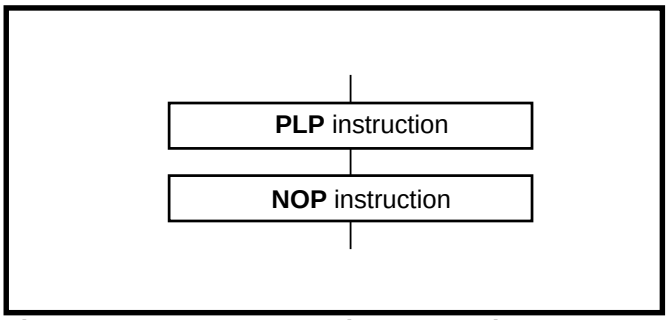


Figure 2.9.3 PLP Instruction Execution

2.9.2 Notes Concerning Decimal Operation

■ **Execution of Decimal Calculations**

The **ADC** and **SBC** are the only instructions which will yield proper decimal results in decimal mode. To calculate in decimal notation, set the decimal mode flag (D) to '1' with the **SED** instruction. After executing the **ADC** or **SBC** instruction, execute another instruction before executing the **SEC**, **CLC**, **SED** or **CLD** instruction.

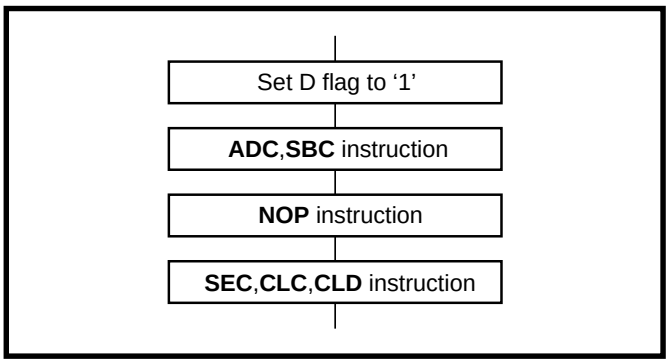


Figure 2.9.4 Execution of Decimal Operation

■ **Note on Flags in Decimal Mode**

When decimal mode is selected, the values of three of the flags in the status register (the N, V, and Z flags) are invalid after the **ADC** or **SBC** instruction is executed.

The carry flag (C) is set to '1' if a carry is generated as a result of the calculation, or is cleared to '0' if a borrow is generated. To determine whether a calculation has generated a carry, the C flag must be initialized to '1' before each calculation.

2.9.3 Notes on JMP Instruction

When using the **JMP** instruction in indirect addressing mode, do not specify the last address on a page as an indirect address.

Pages are sectioned every 256 addresses from address 0.

■ **For other notes, refer to each section.**

2.10 Differences between 7480 and 7481 Group, and 7477 and 7478 Group

The 7480 Group and 7481 Group have almost the same functions as the 7477 Group and 7478 Group. However, take the following differences into consideration when using the former to replace the latter.

2.10.1 Functions Added to 7480 Group and 7481 Group

Table 2.10.1 lists the functions added to the 7480 Group and 7481 Group.

Table 2.10.1 Functions added to the 7480 Group and 7481 Group

Added Function	Description
Bus Arbitration	• In serial I/O communication of the bus-contention system, the level of the TxD pin is compared with that of the RxD pin. When there is a mismatch, a bus arbitration interrupt is generated to detect bus collision.
STP and **WIT** Instruction	• The valid/invalid of the **STP** and **WIT** instructions is selectable by writing 2 times to the **STP** instruction operation control register.
Watchdog Timer	• Watchdog timer returns program to the reset state if a runaway occurs.
Built-in Clamping Diode	• Each pin of ports P4 and P5 has a built-in clamping diode, by which voltages Vcc or more can be applied. **Note:** In the 7480 Group, port P5 is not implemented.

2.10.2 Functions Revised from 7477 Group and 7478 Group

The functions of 7480 Group and 7481 Group whose specifications are revised from those of the 7477 Group and 7478 Group are listed in Table 2.10.2.

Table 2.10.2 Functions Revised from 7477 Group and 7478 Group

	7477 Group and 7478 Group	7480 Group and 7481 Group
RAM Sizes	M3747xM4 ..... 192 bytes M3747xM8/E8 ..... 384 bytes	M3748xM4 ..... 256 bytes M3748xM8/E8 ..... 448 bytes
Port P4	• I/O port • CMOS outputs • In input mode, pull-up transistors connectable.	• I/O port • N-channel open-drain outputs • Built-in clamping diodes • In input mode, $P4_0$–$P4_3$, $P5_0$–$P5_3$ pins have schmidt circuits.
Port P5	• Input port • Pull-up transistors connectable • $P5_0$, $P5_1$ serve as input pins for clock generator. **Note**: The 7477 Group does not have port P5.	**Note:** The 7480 Group does not have $P4_2$, $P4_3$, and $P5_0$–$P5_3$ pins. The 7480 Group and 7481 Group do not have the clock generator for timers.
CNTR Pins	$CNTR_0$ and $CNTR_1$ pins have the alternative functions of $P3_2$ and $P3_3$.	$CNTR_0$ and $CNTR_1$ pins have the alternative functions of $P4_0$ and $P4_1$.
Timer	Four 8-bit timers	Two 8-bit timers Two 16-bit timers
Noise Margin	V_{IL}: 0.25 V_{CC} (Max.) V_{IH}: 0.7 V_{CC} (Min.)	On port P3, P4, and P5 only. V_{IL}: 0.4 V_{CC} (Max.) V_{IH}: 0.8 V_{CC} (Min.) (at V_{CC} = 4.5 V to 5.5 V)
A-D Converter	No V_{REF}-off circuit	To connect/not to connect between V_{REF} and ladder resistors is selectable with a V_{REF}-off circuit.
Package	The M37478Mx/E8-XXXFP and M37478MxT/E8T-XXXFP packaged in 56P6N-A packages.	The M37481Mx/E8-XXXFP and M37481MxT/E8T-XXXFP are packaged in 44P6N-A packages.

2.11 Application Circuit Examples

Figures 2.11.1 and 2.11.2 show two examples of application circuits using the 7480 Group and 7481 Group.

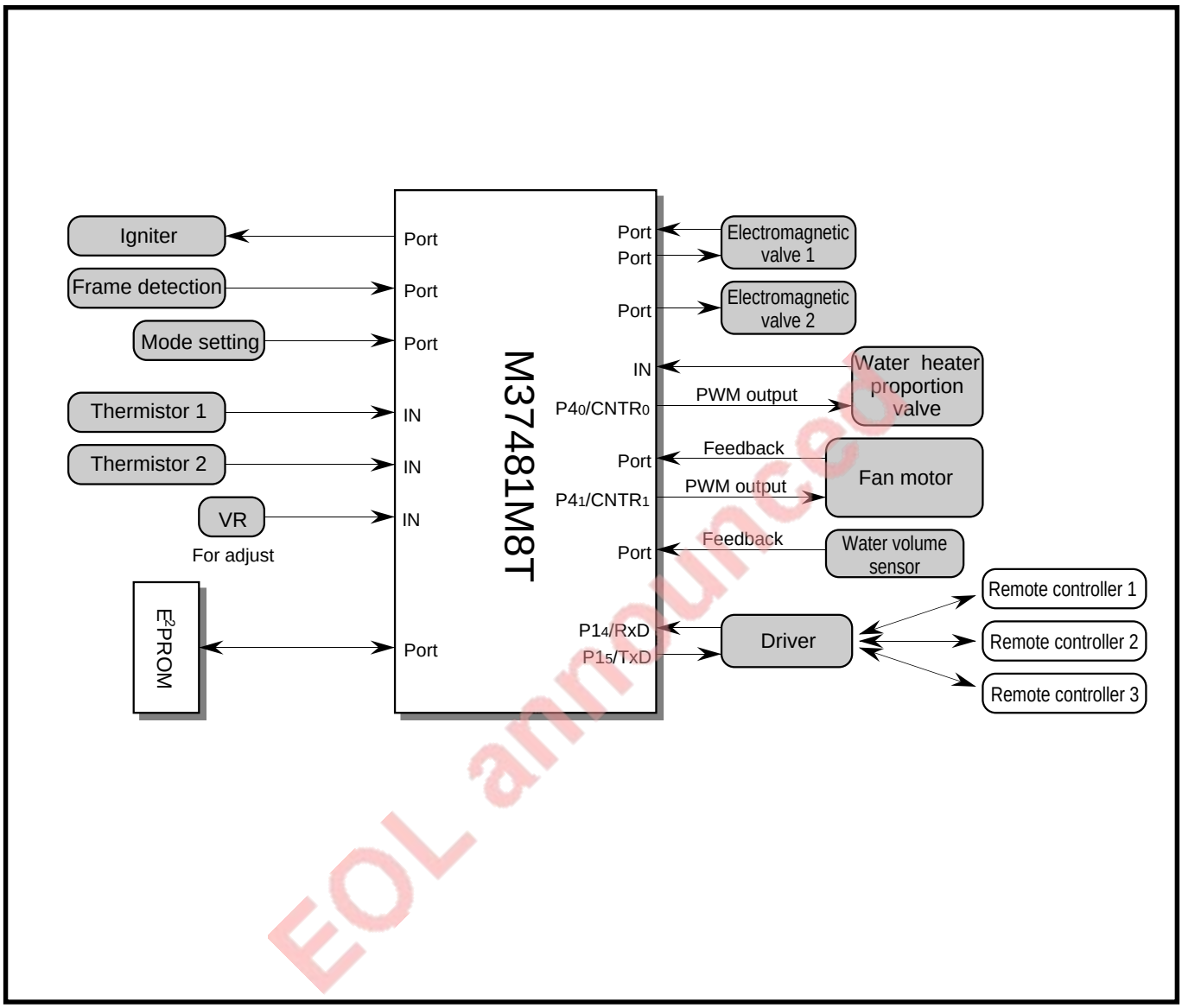


Figure 2.11.1 Application Circuit to Hot-Water Supply Equipment

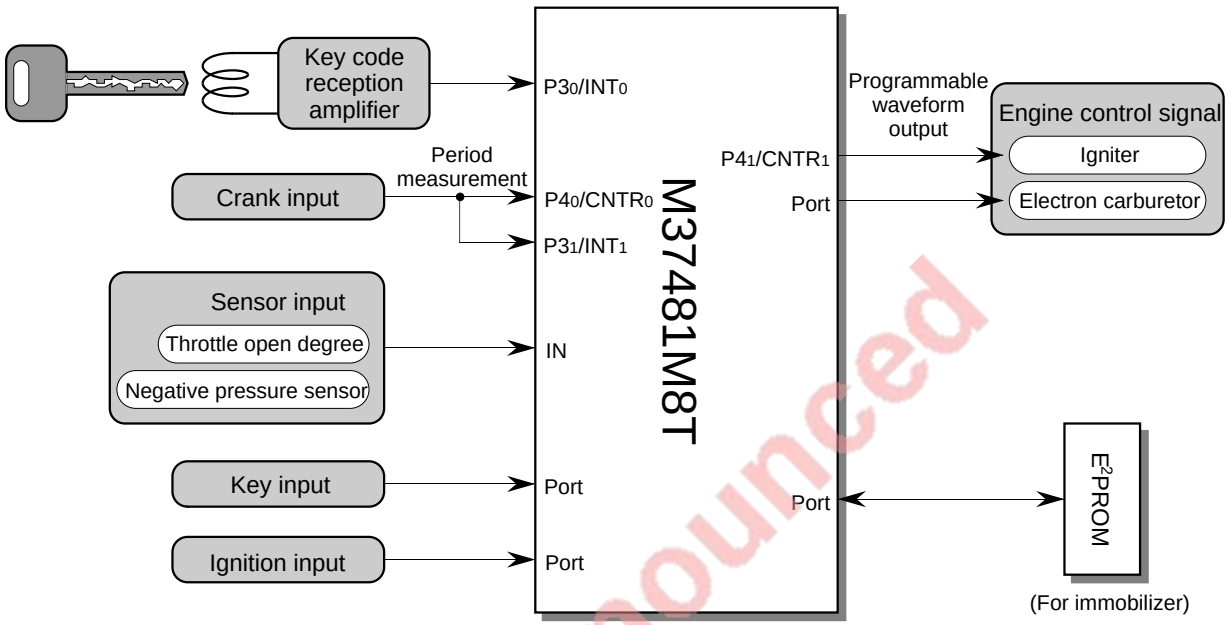


Figure 2.11.2 Application Circuit to Motorcycle Single-Cylinder Engine

This page left blank intentionally.

CHAPTER 3
APPENDICES

3.1 Control Registers

Port Pi (i=0 to 5)

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (Pi, i=0 to 5) [Addresses $00C0_{16}$,$00C2_{16}$,$00C4_{16}$,$00C6_{16}$,$00C8_{16}$,$00CA_{16}$]

b	Function	At reset	R	W
0	■ When used as input ports (Ports P0 to P5) • At reading, input level of pin is read. • At writing, writing to port latch is performed and the pin state is not affected. ■ When used as output ports (Ports P0, P1, P4, P5) • At reading, the last written value into the port latch is read. • At writing, the written value is output externally through a transistor.	Undefined	O	O
1		Undefined	O	O
2		Undefined	O	O
3		Undefined	O	O
4		Undefined	O	O
5		Undefined	O	O
6		Undefined	O	O
7		Undefined	O	O

Note: • In the 7480 Group, port P2 has only bits 0 to 3. The other bits are not implemented. (undefined at reading).
• Port P3 has only bits 0 to 3. The other bits are not implemented ('0' at reading).
• In the 7480 Group, port P4 has only bits 0 and 1. In the 7481 Group, port P4 has only bits 0 to 3. The other bits are not implemented. (bits 4 to 7: '0' at reading, bits 2 and 3 in the 7480 Group: undefined).
• The 7480 Group does not have port P5. In the 7481 Group, port P5 has only bits 0 to 3. The other bits are not implemented. ('0' at reading).

Figure 3.1.1 Port Pi Registers (i = 0 to 5)

Port Pi direction register (i=0,1,4,5)

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (PiD, i=0,1,4,5) [Addresses $00C1_{16}$,$00C3_{16}$,$00C9_{16}$,$00CB_{16}$]

b	Name	Function	At reset	R	W
0	Port Pi direction register (**Note**)	0 : Input mode 1 : Output mode	0	O	O
1		0 : Input mode 1 : Output mode	0	O	O
2		0 : Input mode 1 : Output mode	0	O	O
3		0 : Input mode 1 : Output mode	0	O	O
4		0 : Input mode 1 : Output mode	0	O	O
5		0 : Input mode 1 : Output mode	0	O	O
6		0 : Input mode 1 : Output mode	0	O	O
7		0 : Input mode 1 : Output mode	0	O	O

Note: • In the 7480 Group, port P4 direction register has only bits 0 and 1. In the 7481 Group, port P4 direction register has only bits 0 to 3. The other bits are not implemented (bits 4 to 7: '0' at reading, bits 2 and 3 in the 7480 Group: undefined).
• The 7480 Group does not have port P5 direction register. In the 7481 Group, port P5 has only bits 0 to 3. The other bits are not implemented ('0' at reading).

Figure 3.1.2 Port Pi Direction Registers (i = 0, 1, 4, 5)

Port P0 pull-up control register

b7 b6 b5 b4 b3 b2 b1 b0

Port P0 pull-up control register (P0PCON) [Address $00D0_{16}$]

b	Name	Function	At reset	R	W
0	$P0_0$ pull-up control bit	0 : $P0_0$ No pull-up 1 : $P0_0$ Pull-up	0	O	O
1	$P0_1$ pull-up control bit	0 : $P0_1$ No pull-up 1 : $P0_1$ Pull-up	0	O	O
2	$P0_2$ pull-up control bit	0 : $P0_2$ No pull-up 1 : $P0_2$ Pull-up	0	O	O
3	$P0_3$ pull-up control bit	0 : $P0_3$ No pull-up 1 : $P0_3$ Pull-up	0	O	O
4	$P0_4$ pull-up control bit	0 : $P0_4$ No pull-up 1 : $P0_4$ Pull-up	0	O	O
5	$P0_5$ pull-up control bit	0 : $P0_5$ No pull-up 1 : $P0_5$ Pull-up	0	O	O
6	$P0_6$ pull-up control bit	0 : $P0_6$ No pull-up 1 : $P0_6$ Pull-up	0	O	O
7	$P0_7$ pull-up control bit	0 : $P0_7$ No pull-up 1 : $P0_7$ Pull-up	0	O	O

Note: Pull-up control is valid when the corresponding port is set to the input mode.

Figure 3.1.3 Port P0 Pull-up Control Register

Port P1 pull-up control register

b7 b6 b5 b4 b3 b2 b1 b0

Port P1 pull-up control register (P1PCON) [Address $00D1_{16}$]

b	Name	Function	At reset	R	W
0	$P1_3$–$P1_0$ pull-up control bit	0 : $P1_0$–$P1_3$ No pull-up 1 : $P1_0$–$P1_3$ Pull-up	0	O	O
1	$P1_7$–$P1_4$ pull-up control bit	0 : $P1_4$–$P1_7$ No pull-up 1 : $P1_4$–$P1_7$ Pull-up	0	O	O
2	Not implemented. Writing to these bits is disabled. These bits are undefined at reading.		Undefined	Undefined	×
3			Undefined	Undefined	×
4			Undefined	Undefined	×
5			Undefined	Undefined	×
6			Undefined	Undefined	×
7			Undefined	Undefined	×

Note: Pull-up control is valid only when the corresponding port is set to the input mode.
When port pins $P1_5$–$P1_7$ are used as serial I/O pins, pull-up control of the corresponding port pins is invalid.

Figure 3.1.4 Port P1 Pull-up Control Register

3.1 Control Registers

Port P4P5 input control register

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

Port P4P5 input control register (P4P5CON) [Address $00D2_{16}$]

b	Name	Function	At reset	R	W
0	$P4_2$, $P4_3$ input control bit	When the $P4_2$, $P4_3$ are used as the input port, set this bit to '1'.	0	O	O
1	P5 input control bit	When the P5 is used as the input port, set this bit to '1'.	0	O	O
2	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
3			0	0	×
4			0	0	×
5			0	0	×
6			0	0	×
7			0	0	×

Note: 7480 Group does not have port pins $P4_2$, $P4_3$ and P5, so set bits 0 and 1 to '0'.

Figure 3.1.5 Port P4P5 Input Control Register

Edge polarity selection register

Edge polarity selection register (EG) [Address $00D4_{16}$]

b	Name	Function	At reset	R	W
0	INT_0 edge selection bit	0 : Falling edge 1 : Rising edge	0	O	O
1	INT_1 edge selection bit	0 : Falling edge 1 : Rising edge	0	O	O
2	$CNTR_0$ edge selection bit	0: In event count mode, rising edge counted. : In pulse output mode, operation started at HIGH level output. : In pulse period measurement mode, a period from falling edge until falling edge measured. : In pulse width measurement mode, HIGH-level period measured. : In programmable one-shot output mode, one-shot HIGH pulse generated after operation started at LOW level output. : Interrupt request is generated by detecting falling edge. 1: In event count mode, falling edge counted : In pulse output mode, operation started at LOW level output. : In pulse period measurement mode, a period from rising edge until rising edge measured. : In pulse width measurement mode, LOW-level period measured. : In programmable one-shot output mode, one-shot LOW pulse generated after operation started at HIGH level output. : Interrupt request is generated by detecting rising edge.	0	O	O
3	$CNTR_1$ edge selection bit		0	O	O
4	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
5	INT_1 source selection bit at STP or WIT	0 : $P3_1/INT_1$ 1 : $P0_0$–$P0_7$ LOW level (for key-on wake-up)	0	O	O
6	Not implemented. Writing to these bits are disabled. These bits are undefined at reading.		Undefined	Undefined	×
7			Undefined	Undefined	×

Note: When setting bits 0 to 3, the interrupt request bit may be set to '1'. After setting the following, enable the interrupt.
① Disable interrupts
② Set the edge polarity selection register
③ Set the interrupt request bit to '0'

Figure 3.1.6 Edge Polarity Selection Register

3.1 Control Registers

A-D control register

b7 b6 b5 b4 b3 b2 b1 b0

A-D control register (ADCON) [Address $00D9_{16}$]

b	Name	Function	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0 : $P2_0/IN_0$ 0 0 1 : $P2_1/IN_1$ 0 1 0 : $P2_2/IN_2$ 0 1 1 : $P2_3/IN_3$ 1 0 0 : $P2_4/IN_4$ 1 0 1 : $P2_5/IN_5$ 1 1 0 : $P2_6/IN_6$ 1 1 1 : $P2_7/IN_7$ (**Note**)	0	O	O
1			0	O	O
2			0	O	O
3	A-D conversion completion bit	0 : Conversion in progress 1 : Conversion completed	1	O	*
4	V_{REF} connection selection bit	0 : Disconnect between V_{REF} pin and ladder resistor 1 : Connect between V_{REF} pin and ladder resistor	0	O	O
5	Not implemented. Writing to these bits is disabled. These bits are undefined at reading.		Undefined	Undefined	×
6			Undefined	Undefined	×
7			Undefined	Undefined	×

*: The bit can be set to '0' by software, but cannot be set to '1'.

Note: Do not perform setting in the 7480 Group.

Figure 3.1.7 A-D Control Register

A-D conversion register

b7 b6 b5 b4 b3 b2 b1 b0

A-D conversion register (AD) [Address $00DA_{16}$]

b	Function	At reset	R	W
0	Read-only register to store the A-D conversion result.	Undefined	O	×
1		Undefined	O	×
2		Undefined	O	×
3		Undefined	O	×
4		Undefined	O	×
5		Undefined	O	×
6		Undefined	O	×
7		Undefined	O	×

Figure 3.1.8 A-D Conversion Register

STP instruction operation control register

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

STP instruction operation control register (STPCON) [Address $00DE_{16}$]

b	Name	Function	At reset	R	W
0	**STP** and **WIT** valid/invalid selection bit (**Note**)	0 : **STP/WIT** instruction valid 1 : **STP/WIT** instruction invalid	1	O	O
1	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
2			0	0	×
3			0	0	×
4			0	0	×
5			0	0	×
6			0	0	×
7			0	0	×

Note: The **STP** and **WIT** instructions are invalid after the system is released from reset. When using these instructions, set **STP** and **WIT** valid/invalid selection bit of the **STP** instruction operation control register to '1', then set this bit to '0'. (Writing twice successively)
When not using the **STP** and **WIT** instructions, set this bit to '1' either once or twice.

Figure 3.1.9 STP Instruction Operation Control Register

3.1 Control Registers

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

Transmit/Receive buffer register (TB/RB) [Address $00E0_{16}$]

b	Function	At reset	R	W
0	In transmission: Transmit data is transferred to the transmit shift register by writing transmit data. In reception: When data is stored completely in the receive shift register, the receive data is transferred to this register.	Undefined	Note	
1		Undefined		
2		Undefined		
3		Undefined		
4		Undefined		
5		Undefined		
6		Undefined		
7		Undefined		

Note: In transmission, this register is a write-only register.
In reception, this register is a read-only register.

Figure 3.1.10 Transmit/Receive Buffer Register

Serial I/O status register

b7 b6 b5 b4 b3 b2 b1 b0

1

Serial I/O status register (SIOSTS) [Address $00E1_{16}$]

b	Name	Function	At reset	R	W
0	Transmit buffer empty flag (TBE)	0 : Buffer full 1 : Buffer empty	0	O	×
1	Receive buffer full flag (RBF)	0 : Buffer empty 1 : Buffer full	0	O	×
2	Transmit shift completion flag (TSC)	0 : Transmit shift in progress 1 : Transmit shift completed	0	O	×
3	Overrun error flag (OE)	0 : No error 1 : Overrun error	0	O	×
4	Parity error flag (PE)	0 : No error 1 : Parity error	0	O	×
5	Framing error flag (FE)	0 : No error 1 : Framing error	0	O	×
6	Summing error flag (SE)	0 : OE U PE U FE=0 1 : OE U PE U FE=1	0	O	×
7	This bit is fixed to '1'.		1	1	×

Note: b4 and b5 are valid only in UART

Figure 3.1.11 Serial I/O Status Register

Serial I/O control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O control register (SIOCON) [Address 00E2$_{16}$]

b	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : f(XIN)/4 1 : f(XIN)/16	0	O	O
1	Serial I/O synchronous clock selection bit (SCS)	when clock synchronous serial I/O is selected 0 : BRG output/4 1 : External clock input when UART is selected 0 : BRG output/16 1 : External clock input/16	0	O	O
2	$\overline{\text{SRDY}}$ output enable bit (SRDY)	0 : P1$_7$ pin 1 : $\overline{\text{SRDY}}$ output pin	0	O	O
3	Transmit interrupt source selection bit (TIC)	0 : Interrupt occurs when transmit buffer is empty. 1 : Interrupt occurs when transmit shift operation is completed.	0	O	O
4	Transmit enable bit (TE)	0 : Transmit disabled 1 : Transmit enabled	0	O	O
5	Receive enable bit (RE)	0 : Receive disabled 1 : Receive enabled	0	O	O
6	Serial I/O mode selection bit (SIOM)	0 : Asynchronous serial I/O (UART) 1 : Clock synchronous serial I/O	0	O	O
7	Serial I/O enable bit (SIOE)	0 : Serial I/O disabled 1 : Serial I/O enabled	0	O	O

Figure 3.1.12 Serial I/O Control Register

UART control register

b7 b6 b5 b4 b3 b2 b1 b0

1 1 1 1

UART control register (UARTCON) [Address 00E3$_{16}$]

b	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	O	O
1	Parity enable bit (PARE)	0 : Parity disabled 1 : Parity enabled	0	O	O
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	O	O
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	O	O
4	These bits are fixed to '1'.		1	1	×
5			1	1	×
6			1	1	×
7			1	1	×

Figure 3.1.13 UART Control Register

3.1 Control Registers

Baud rate generator

b7 b6 b5 b4 b3 b2 b1 b0

Baud rate generator (BRG) [Address $00E4_{16}$]

b	Function	At reset	R	W
0	• 8-bit timer for baud rate generation of serial I/O. • Valid only when BRG output divided by 4 or 16 is selected as synchronous clock.	Undefined	O	O
1		Undefined	O	O
2		Undefined	O	O
3		Undefined	O	O
4		Undefined	O	O
5		Undefined	O	O
6		Undefined	O	O
7		Undefined	O	O

Figure 3.1.14 Baud Rate Generator

Bus collision detection control register

b7 b6 b5 b4 b3 b2 b1 b0

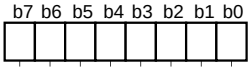
0 0 0 0 0 0 0

Bus collision detection control register (BUSARBCON) [Address $00E5_{16}$]

b	Name	Function	At reset	R	W
0	Bus collision detection enable bit	0 : Collision detection invalid 1 : Collision detection valid	0	O	O
1	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
2			0	0	×
3			0	0	×
4			0	0	×
5			0	0	×
6			0	0	×
7			0	0	×

Figure 3.1.15 Bus Collision Detection Control Register

Watchdog timer H



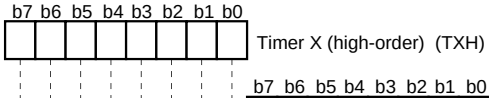
Watchdog timer H (WDTH) [Address 00EF16]

b	Function	At reset	R	W
0	The high-order count value of watchdog timer is indicated.	1	O	Note
1		1	O	
2		1	O	
3		1	O	
4		1	O	
5		1	O	
6		1	O	
7		1	O	

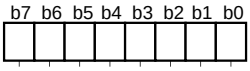
Note: The following value is set by writing arbitrary data.
- watchdog timer L ← '7F16'
- watchdog timer H ← 'FF16'

Figure 3.1.16 Watchdog Timer H

Timer X (Timer X latch)



Timer X (high-order) (TXH) [Address $00F1_{16}$]



Timer X (low-order) (TXL) [Address $00F0_{16}$]

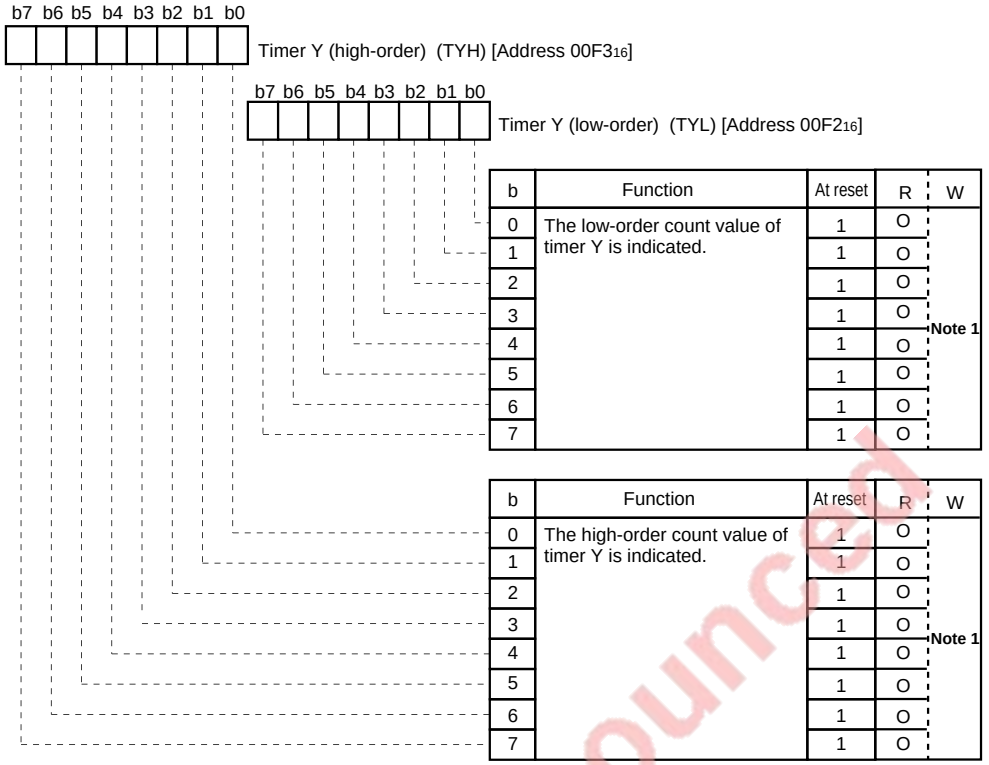
b	Function	At reset	R	W
0	The low-order count value of timer X is indicated.	1	O	Note 1
1		1	O	
2		1	O	
3		1	O	
4		1	O	
5		1	O	
6		1	O	
7		1	O	

b	Function	At reset	R	W
0	The high-order count value of timer X is indicated.	1	O	Note 1
1		1	O	
2		1	O	
3		1	O	
4		1	O	
5		1	O	
6		1	O	
7		1	O	

Notes 1: Do not write to these bits in pulse period measurement mode or pulse width measurement mode.
2: When b3 of timer X mode register is '0', writing to latch and timer is simultaneously performed. When b3 is '1', writing to only latch is performed.
3: When reading/writing from/to timer X, read/write from/to both high-order and low-order bytes. At reading, read the high-order byte and the low-order byte in this order. At writing, write the low-order byte and the high-order byte in this order.

Figure 3.1.17 Timer X

Timer Y (Timer Y latch)



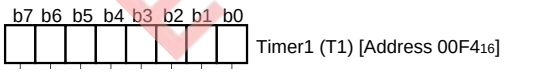
b	Function	At reset	R	W
0	The low-order count value of timer Y is indicated.	1	O	Note 1
1		1	O	
2		1	O	
3		1	O	
4		1	O	
5		1	O	
6		1	O	
7		1	O	

b	Function	At reset	R	W
0	The high-order count value of timer Y is indicated.	1	O	Note 1
1		1	O	
2		1	O	
3		1	O	
4		1	O	
5		1	O	
6		1	O	
7		1	O	

Notes 1: Do not write to these bits in pulse period measurement mode or pulse width measurement mode.
2: When b3 of timer Y mode register is '0', writing to latch and timer is simultaneously performed. When b3 is '1', writing to only latch is performed.
3: When reading/writing from/to timer Y, read/write from/to both high-order and low-order bytes. At reading, read the high-order byte and the low-order byte in this order. At writing, write the low-order byte and the high-order byte in this order.

Figure 3.1.18 Timer Y

Timer 1 (Timer 1 latch)

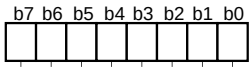


b	Function	At reset	R	W
0	The timer 1 count value is indicated.	1	O	O
1		1	O	O
2		1	O	O
3		1	O	O
4		1	O	O
5		1	O	O
6		1	O	O
7		1	O	O

Figure 3.1.19 Timer 1

3.1 Control Registers

Timer 2 (Timer 2 latch)

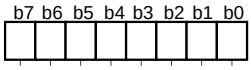


Timer 2 (T2) [Address $00F5_{16}$]

b	Function	At reset	R	W
0	The timer 2 count value is indicated.	Undefined	O	O
1		Undefined	O	O
2		Undefined	O	O
3		Undefined	O	O
4		Undefined	O	O
5		Undefined	O	O
6		Undefined	O	O
7		Undefined	O	O

Figure 3.1.20 Timer 2

Timer X mode register



Timer X mode register (TXM) [Address $00F6_{16}$]

b	Name	Function	At reset	R	W
0	Timer X operating mode bits	b2 b1 b0 0 0 0 : Timer • event count mode 0 0 1 : Pulse output mode 0 1 0 : Pulse period measurement mode 0 1 1 : Pulse width measurement mode 1 0 0 : Programmable waveform generation mode 1 0 1 : Programmable one-shot output mode 1 1 0 : PWM mode 1 1 1 : Not available	0	O	O
1			0	O	O
2			0	O	O
3	Timer X write control bit	0 : Writing to both latch and timer 1 : Writing to latch only	0	O	O
4	Output level latch	0 : LOW output from $CNTR_0$ pin 1 : HIGH output from $CNTR_0$ pin	0	O	O
5	Timer X trigger selection bit	0 : Timer X free run in programmable waveform generation mode 1 : Trigger occurrence (input signal of INT_0 pin) and timer X start in programmable waveform generation mode	0	O	O
6	Timer X count source selection bits	b7 b6 0 0 : $f(X_{IN})/2$ 0 1 : $f(X_{IN})/8$ 1 0 : $f(X_{IN})/16$ 1 1 : Input from $CNTR_0$ pin	0	O	O
7			0	O	O

Figure 3.1.21 Timer X Mode Register

Timer Y mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer Y mode register (TYM) [Address $00F7_{16}$]

b	Name	Function	At reset	R	W
0	Timer Y operating mode bits	b2 b1 b0 0 0 0 : Timer • event count mode 0 0 1 : Pulse output mode 0 1 0 : Pulse period measurement mode 0 1 1 : Pulse width measurement mode 1 0 0 : Programmable waveform generation mode 1 0 1 : Programmable one-shot output mode 1 1 0 : PWM mode 1 1 1 : Not available	0	O	O
1			0	O	O
2			0	O	O
3	Timer Y write control bit	0 : Writing to both latch and timer 1 : Writing to latch only	0	O	O
4	Output level latch	0 : LOW output from $CNTR_1$ pin 1 : HIGH output from $CNTR_1$ pin	0	O	O
5	Timer Y trigger selection bit	0 : Timer Y free run in programmable waveform generation mode 1 : Trigger occurrence (input signal of INT_1 pin) and timer Y start in programmable waveform generation mode	0	O	O
6	Timer Y count source selection bits	b7 b6 0 0 : $f(X_{IN})/2$ 0 1 : $f(X_{IN})/8$ 1 0 : $f(X_{IN})/16$ 1 1 : Input from $CNTR_1$ pin	0	O	O
7			0	O	O

Figure 3.1.22 Timer Y Mode Register

Timer XY control register

b7 b6 b5 b4 b3 b2 b1 b0

0 0 0 0 0 0

Timer XY control register (TXYCON) [Address $00F8_{16}$]

b	Name	Function	At reset	R	W
0	Timer X stop control bit	0 : Count operation 1 : Count stop	1	O	O
1	Timer Y stop control bit	0 : Count operation 1 : Count stop	1	O	O
2	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
3			0	0	×
4			0	0	×
5			0	0	×
6			0	0	×
7			0	0	×

Figure 3.1.23 Timer XY Control Register

3.1 Control Registers

Timer 1 mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0		0	0		

Timer 1 mode register (T1M) [Address $00F9_{16}$]

b	Name	Function	At reset	R	W
0	Timer 1 stop control bit	0 : Count operation 1 : Count stop	0	O	O
1	Timer 1 operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode	0	O	O
2	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
3			0	0	×
4	Output level latch	0 : LOW output from T_0 pin 1 : HIGH output from T_0 pin	0	O	O
5	Not implemented. Writing to this bit is disabled. This bit is '0' at reading.		0	0	×
6	Timer 1 count source selection bits	b7 b6 0 0 : $f(X_{IN})/8$ 0 1 : $f(X_{IN})/64$ 1 0 : $f(X_{IN})/128$ 1 1 : $f(X_{IN})/256$	0	O	O
7			0	O	O

Figure 3.1.24 Timer 1 Mode Register

Timer 2 mode register

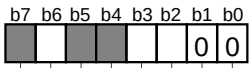
b7	b6	b5	b4	b3	b2	b1	b0
		0		0	0		

Timer 2 mode register (T2M) [Address $00FA_{16}$]

b	Name	Function	At reset	R	W
0	Timer 2 stop control bit	0 : Count operation 1 : Count stop	0	O	O
1	Timer 2 operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode	0	O	O
2	Not implemented. Writing to these bits is disabled. These bits are '0' at reading.		0	0	×
3			0	0	×
4	Output level latch	0 : LOW output from T_1 pin 1 : HIGH output from T_1 pin	0	O	O
5	Not implemented. Writing to this bit is disabled. This bit is '0' at reading.		0	0	×
6	Timer 2 count source selection bits	b7 b6 0 0 : $f(X_{IN})/8$ 0 1 : $f(X_{IN})/64$ 1 0 : $f(X_{IN})/128$ 1 1 : $f(X_{IN})/256$	0	O	O
7			0	O	O

Figure 3.1.25 Timer 2 Mode Register

CPU mode register



CPU mode register (CPUM) [Address $00FB_{16}$]

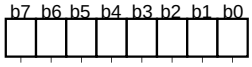
b	Name	Function	At reset	R	W
0	Fix these bits to '0'.		0	O	0
1			0	O	0
2	Stack page selection bit (**Note**)	0 : Zero page 1 : 1 page	0	O	O
3	Watchdog timer L count source selection bit	0 : $f(X_{IN})/8$ 1 : $f(X_{IN})/16$	0	O	O
4	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
5	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×
6	Clock division ratio selection bit	0 : $f(X_{IN})/2$ (high-speed mode) 1 : $f(X_{IN})/8$ (medium-speed mode)	0	O	O
7	Not implemented. Writing to this bit is disabled. This bit is undefined at reading.		Undefined	Undefined	×

Note: In the products whose RAM size is 192 bytes or less, set this bit to '0'.

Figure 3.1.26 CPU Mode Register

3.1 Control Registers

Interrupt request register 1



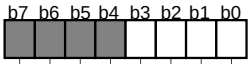
Interrupt request register 1 (IREQ1) [Address 00FC16]

b	Name	Function	At reset	R	W
0	Timer X interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
1	Timer Y interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
2	Timer 1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
3	Timer 2 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
4	Serial I/O receive interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
5	Serial I/O transmit interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
6	Bus arbitration interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
7	A-D conversion completion interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*

*: The bit can be set to '0' by software, but cannot be set to '1'.

Figure 3.1.27 Interrupt Request Register 1

Interrupt request register 2



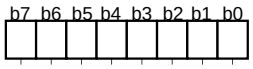
Interrupt request register 2 (IREQ2) [Address 00FD16]

b	Name	Function	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
1	INT_1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
2	$CNTR_0$ interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
3	$CNTR_1$ interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	O	*
4	Not implemented. Writing to these bits is disabled. These bits are undefined at reading.		Undefined	Undefined	×
5			Undefined	Undefined	×
6			Undefined	Undefined	×
7			Undefined	Undefined	×

*: The bit can be set to '0' by software, but cannot be set to '1'.

Figure 3.1.28 Interrupt Request Register 2

Interrupt control register 1

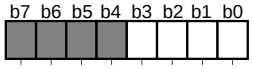


Interrupt control register 1 (ICON1) [Address 00FE16]

b	Name	Function	At reset	R	W
0	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
1	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
2	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
3	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
4	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
5	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
6	Bus arbitration interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
7	A-D conversion completion interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O

Figure 3.1.29 Interrupt Control Register 1

Interrupt control register 2



Interrupt control register 2 (ICON2) [Address 00FF16]

b	Name	Function	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
1	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
2	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
3	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	O	O
4	Not implemented. Writing to these bits are disabled. These bits are undefined at reading.		Undefined	Undefined	×
5			Undefined	Undefined	×
6			Undefined	Undefined	×
7			Undefined	Undefined	×

Figure 3.1.30 Interrupt Control Register 2

3.2 Mask ROM Confirmation Forms

GZZ-SH09-84B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M2T-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation
Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37480M2T-XXXSP ☐ M37480M2T-XXXFP

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

EPROM type (indicate the type used)

☐ 27128	☐ 27256	☐ 27512
EPROM address	EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37480M2T–'	0000_{16} Area for ASCII codes of the name of the product 'M37480M2T-'	0000_{16} Area for ASCII codes of the name of the product 'M37480M2T–'
$000F_{16}$ 0010_{16}	$000F_{16}$ 0010_{16}	$000F_{16}$ 0010_{16}
$2FFF_{16}$ 3000_{16} ROM (4K)	$6FFF_{16}$ 7000_{16} ROM (4K)	$EFFF_{16}$ $F000_{16}$ ROM (4K)
$3FFF_{16}$	$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37480M2T–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37480M2T–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' T ' = 54_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	' – ' = $2D_{16}$
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'0' = 30_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'2' = 32_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-84B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M2T-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	✻=△$C000 △.BYTE△'M37480M2T–'	✻=△$8000 △.BYTE△'M37480M2T–'	✻=△$0000 △.BYTE△'M37480M2T–'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification
Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (32P4B for M37480M2T-XXXSP, 32P2W-A for M37480M2T-XXXFP) and attach to the mask ROM confirmation form.

✵ 3. Comments

GZZ-SH09-85B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M4-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37480M4-XXXSP ☐ M37480M4-XXXFP

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

EPROM type (indicate the type used)

☐ 27128	☐ 27256	☐ 27512
EPROM address	EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37480M4–'	0000_{16} Area for ASCII codes of the name of the product 'M37480M4-'	0000_{16} Area for ASCII codes of the name of the product 'M37480M4–'
$000F_{16}$	$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}	0010_{16}
$1FFF_{16}$	$5FFF_{16}$	$DFFF_{16}$
2000_{16} ROM (8K)	6000_{16} ROM (8K)	$E000_{16}$ ROM (8K)
$3FFF_{16}$	$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37480M4–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37480M4–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'0' = 30_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-85B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M4-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	✻=△$C000 △.BYTE△'M37480M4–'	✻=△$8000 △BYTE△'M37480M4–'	✻=△$0000 △.BYTE△'M37480M4–'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (32P4B for M37480M4-XXXSP, 32P2W-A for M37480M4-XXXFP) and attach to the mask ROM confirmation form.

✵ 3. Comments

GZZ-SH09-86B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M4T-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37480M4T-XXXSP ☐ M37480M4T-XXXFP

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

EPROM type (indicate the type used)

☐ 27128	☐ 27256	☐ 27512
EPROM address	EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37480M4T–'	0000_{16} Area for ASCII codes of the name of the product 'M37480M4T-'	0000_{16} Area for ASCII codes of the name of the product 'M37480M4T–'
$000F_{16}$	$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}	0010_{16}
$1FFF_{16}$	$5FFF_{16}$	$DFFF_{16}$
2000_{16} ROM (8K)	6000_{16} ROM (8K)	$E000_{16}$ ROM (8K)
$3FFF_{16}$	$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37480M4T–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37480M4T–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' T ' = 54_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	' – ' = $2D_{16}$
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'0' = 30_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-86B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M4T-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	✻= △$C000 △.BYTE△'M37480M4T–'	✻= △$8000 △.BYTE△'M37480M4T–'	✻= △$0000 △.BYTE△'M37480M4T–'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✺ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (32P4B for M37480M4T-XXXSP, 32P2W-A for M37480M4T-XXXFP) and attach to the mask ROM confirmation form.

✺ 3. Comments

GZZ-SH09-87B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M8-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37480M8-XXXSP ☐ M37480M8-XXXFP

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

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37480M8–'	0000_{16} Area for ASCII codes of the name of the product 'M37480M8–'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$3FFF_{16}$	$BFFF_{16}$
4000_{16} ROM (16K)	$C000_{16}$ ROM (16K)
$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37480M8–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37480M8–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'0' = 30_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-87B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M8-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27256	27512
The pseudo-command	✻=△$8000 △BYTE△'M37480M8-'	✻=△$0000 △BYTE△'M37480M8-'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (32P4B for M37480M8-XXXSP, 32P2W-A for M37480M8-XXXFP) and attach to the mask ROM confirmation form.

✵ 3. Comments

GZZ-SH09-88B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M8T-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

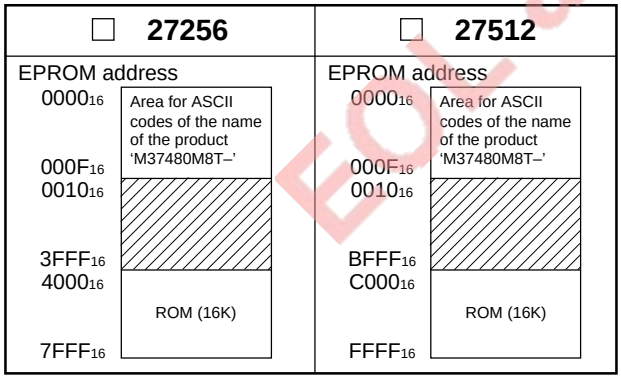
✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37480M8T-XXXSP ☐ M37480M8T-XXXFP

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

EPROM type (indicate the type used)



(1) Set "FF$_{16}$" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37480M8T–' to addresses 0000$_{16}$ to 000F$_{16}$. ASCII codes 'M37480M8T–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000$_{16}$	'M' = 4D$_{16}$	0008$_{16}$	' T ' = 54$_{16}$
0001$_{16}$	'3' = 33$_{16}$	0009$_{16}$	' – ' = 2D$_{16}$
0002$_{16}$	'7' = 37$_{16}$	000A$_{16}$	FF$_{16}$
0003$_{16}$	'4' = 34$_{16}$	000B$_{16}$	FF$_{16}$
0004$_{16}$	'8' = 38$_{16}$	000C$_{16}$	FF$_{16}$
0005$_{16}$	'0' = 30$_{16}$	000D$_{16}$	FF$_{16}$
0006$_{16}$	'M' = 4D$_{16}$	000E$_{16}$	FF$_{16}$
0007$_{16}$	'8' = 38$_{16}$	000F$_{16}$	FF$_{16}$

GZZ-SH09-88B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480M8T-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27256	27512
The pseudo-command	✻=△ $8000 △.BYTE △'M37480M8T-'	✻=△ $0000 △.BYTE △'M37480M8T-'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (32P4B for M37480M8T-XXXSP, 32P2W-A for M37480M8T-XXXFP) and attach to the mask ROM confirmation form.

✵ 3. Comments

GZZ-SH09-78B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M2T-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37481M2T-XXXSP ☐ M37481M2T-XXXFP

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

EPROM type (indicate the type used)

☐ 27128	☐ 27256	☐ 27512
EPROM address	EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37481M2T–'	0000_{16} Area for ASCII codes of the name of the product 'M37481M2T-'	0000_{16} Area for ASCII codes of the name of the product 'M37481M2T–'
$000F_{16}$ 0010_{16}	$000F_{16}$ 0010_{16}	$000F_{16}$ 0010_{16}
$2FFF_{16}$ 3000_{16} ROM (4K)	$6FFF_{16}$ 7000_{16} ROM (4K)	$EFFF_{16}$ $F000_{16}$ ROM (4K)
$3FFF_{16}$	$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37481M2T–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37481M2T–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' T ' = 54_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	' – ' = $2D_{16}$
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'2' = 32_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-78B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M2T-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	✻=△$C000 △BYTE△'M37481M2T–'	✻=△$8000 △BYTE△'M37481M2T–'	✻=△$0000 △BYTE△ 'M37481M2T–'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification
Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (42P4B for M37481M2T-XXXSP, 44P6N-A for M37481M2T-XXXFP) and attach to the mask ROM confirmation form.

✵ 3. Comments

GZZ-SH09-79B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M4-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37481M4-XXXSP ☐ M37481M4-XXXFP

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

EPROM type (indicate the type used)

☐ 27128	☐ 27256	☐ 27512
EPROM address	EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37481M4–'	0000_{16} Area for ASCII codes of the name of the product 'M37481M4-'	0000_{16} Area for ASCII codes of the name of the product 'M37481M4–'
$000F_{16}$	$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}	0010_{16}
$1FFF_{16}$	$5FFF_{16}$	$DFFF_{16}$
2000_{16} ROM (8K)	6000_{16} ROM (8K)	$E000_{16}$ ROM (8K)
$3FFF_{16}$	$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37481M4–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37481M4–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-79B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M4-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	✻=△$C000 △.BYTE△'M37481M4–'	✻=△$8000 △.BYTE△'M37481M4–'	✻=△$0000 △.BYTE△'M37481M4–'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (42P4B for M37481M4-XXXSP, 44P6N-A for M37481M4-XXXFP) and attach to the mask ROM confirmation form.

✵ 3. Comments

GZZ-SH09-80B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M4T-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37481M4T-XXXSP ☐ M37481M4T-XXXFP

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

EPROM type (indicate the type used)

☐ 27128	☐ 27256	☐ 27512
EPROM address	EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37481M4T–'	0000_{16} Area for ASCII codes of the name of the product 'M37481M4T-'	0000_{16} Area for ASCII codes of the name of the product 'M37481M4T–'
$000F_{16}$	$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}	0010_{16}
$1FFF_{16}$	$5FFF_{16}$	$DFFF_{16}$
2000_{16} ROM (8K)	6000_{16} ROM (8K)	$E000_{16}$ ROM (8K)
$3FFF_{16}$	$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37481M4T–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37481M4T–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' T ' = 54_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	' – ' = $2D_{16}$
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-80B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M4T-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	✻=△$C000 △.BYTE △'M37481M4T–'	✻=△$8000 △BYTE △'M37481M4T–'	✻=△$0000 △.BYTE△'M37481M4T–'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (42P4B for M37481M4T-XXXSP, 44P6N-A for M37481M4T-XXXFP) and attach to the mask ROM confirmation form.

✵ 3. Comments

GZZ-SH09-81B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M8-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37481M8-XXXSP ☐ M37481M8-XXXFP

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

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37481M8–'	0000_{16} Area for ASCII codes of the name of the product 'M37481M8–'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$3FFF_{16}$	$BFFF_{16}$
4000_{16} ROM (16K)	$C000_{16}$ ROM (16K)
$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37481M8–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37481M8–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-81B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M8-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27256	27512
The pseudo-command	✻=△$8000 △.BYTE△'M37481M8-'	✻=△$0000 △.BYTE△'M37481M8-'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✺ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (42P4B for M37481M8-XXXSP, 44P6N-A for M37481M8-XXXFP) and attach to the mask ROM confirmation form.

✺ 3. Comments

GZZ-SH09-82B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M8T-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37481M8T-XXXSP ☐ M37481M8T-XXXFP

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

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37481M8T–'	0000_{16} Area for ASCII codes of the name of the product 'M37481M8T–'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$3FFF_{16}$	$BFFF_{16}$
4000_{16} ROM (16K)	$C000_{16}$ ROM (16K)
$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37481M8T–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37481M8T–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' T ' = 54_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	' – ' = $2D_{16}$
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-82B<56A0>

Mask ROM number	

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481M8T-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27256	27512
The pseudo-command	✻=△ $8000 △.BYTE △'M37481M8T-'	✻=△ $0000 △.BYTE △ 'M37481M8T-'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (42P4B for M37481M8T-XXXSP, 44P6N-A for M37481M8T-XXXFP) and attach to the mask ROM confirmation form.

✵ 3. Comments

3.3 ROM Programming Confirmation Forms

GZZ-SH09-91B<56A0>

ROM number	

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480E8-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

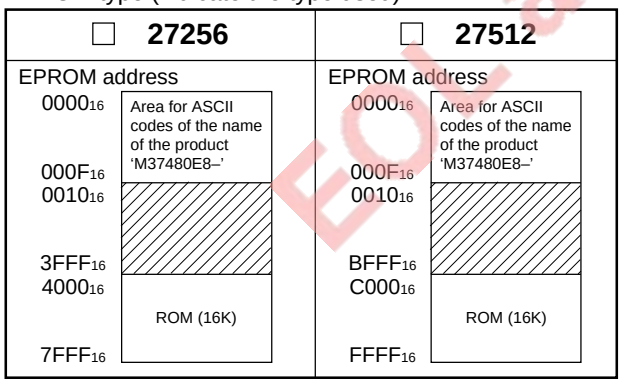
✵	Customer	Company name	TEL ( )	Issuance signature	Submitted by	Supervisor
		Date issued	Date:			

✵ 1. Confirmation
Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce ROM programming based on this data. We shall assume the responsibility for errors only if the ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37480E8-XXXSP ☐ M37480E8-XXXFP

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

EPROM type (indicate the type used)



(1) Set "FF$_{16}$" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37480E8–' to addresses 0000$_{16}$ to 000F$_{16}$. ASCII codes 'M37480E8–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000$_{16}$	'M' = 4D$_{16}$	0008$_{16}$	' – ' = 2D$_{16}$
0001$_{16}$	'3' = 33$_{16}$	0009$_{16}$	FF$_{16}$
0002$_{16}$	'7' = 37$_{16}$	000A$_{16}$	FF$_{16}$
0003$_{16}$	'4' = 34$_{16}$	000B$_{16}$	FF$_{16}$
0004$_{16}$	'8' = 38$_{16}$	000C$_{16}$	FF$_{16}$
0005$_{16}$	'0' = 30$_{16}$	000D$_{16}$	FF$_{16}$
0006$_{16}$	'E' = 45$_{16}$	000E$_{16}$	FF$_{16}$
0007$_{16}$	'8' = 38$_{16}$	000F$_{16}$	FF$_{16}$

GZZ-SH09-91B<56A0>

ROM number	

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480E8-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27256	27512
The pseudo-command	✻= △$8000 △BYTE△'M37480E8-'	✻=△$0000 △BYTE△'M37480E8-'

Note : If the name of the product written to the EPROMs does not match the name of the ROM programming confirmation form, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Please submit the shrink DIP package Mark Specification Form (only for built-in One Time PROM microcomputer) for the M37480E8-XXXSP or the 32P2W-A Mark Specification Form for the M37480E8-XXXFP.

✵ 3. Comments

GZZ-SH09-92B<56A0>

ROM number	

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480E8T-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce ROM programming based on this data. We shall assume the responsibility for errors only if the ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37480E8T-XXXSP ☐ M37480E8T-XXXFP

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

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37480E8T–'	0000_{16} Area for ASCII codes of the name of the product 'M37480E8T–'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$3FFF_{16}$	$BFFF_{16}$
4000_{16} ROM (16K)	$C000_{16}$ ROM (16K)
$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37480E8T–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37480E8T–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' T ' = 54_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	' – ' = $2D_{16}$
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'0' = 30_{16}	$000D_{16}$	FF_{16}
0006_{16}	'E' = 45_{16}	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-92B<56A0>

ROM number	

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37480E8T-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27256	27512
The pseudo-command	✻= △$8000 △.BYTE △'M37480E8T-'	✻= △$0000 △.BYTE △'M37480E8T-'

Note : If the name of the product written to the EPROMs does not match the name of the ROM programming confirmation form, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Please submit the shrink DIP package Mark Specification Form (only for built-in One Time PROM microcomputer) for the M37480E8T-XXXSP or the 32P2W-A Mark Specification Form for the M37480E8T-XXXFP.

✵ 3. Comments

GZZ-SH09-89B<56A0>

ROM number	

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481E8-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce ROM programming based on this data. We shall assume the responsibility for errors only if the ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37481E8-XXXSP ☐ M37481E8-XXXFP

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

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37481E8–'	0000_{16} Area for ASCII codes of the name of the product 'M37481E8–'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$3FFF_{16}$	$BFFF_{16}$
4000_{16} ROM (16K)	$C000_{16}$ ROM (16K)
$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37481E8–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37481E8–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'E' = 45_{16}	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-89B<56A0>

ROM number	

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481E8-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27256	27512
The pseudo-command	✻= △$8000 △BYTE△'M37481E8-'	✻=△$0000 △BYTE△'M37481E8-'

Note : If the name of the product written to the EPROMs does not match the name of the ROM programming confirmation form, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Please submit the shrink DIP package Mark Specification Form (only for built-in One Time PROM microcomputer) for the M37481E8-XXXSP or the 44P6N-A Mark Specification Form for the M37481E8-XXXFP.

✵ 3. Comments

GZZ-SH09-90B<56A0>

ROM number	

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481E8T-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain identical data, we will produce ROM programming based on this data. We shall assume the responsibility for errors only if the ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M37481E8T-XXXSP ☐ M37481E8T-XXXFP

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

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Area for ASCII codes of the name of the product 'M37481E8T–'	0000_{16} Area for ASCII codes of the name of the product 'M37481E8T–'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$3FFF_{16}$	$BFFF_{16}$
4000_{16} ROM (16K)	$C000_{16}$ ROM (16K)
$7FFF_{16}$	$FFFF_{16}$

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37481E8T–' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37481E8T–' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' T ' = 54_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	' – ' = $2D_{16}$
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'8' = 38_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'E' = 45_{16}	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

GZZ-SH09-90B<56A0>

ROM number	

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37481E8T-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27256	27512
The pseudo-command	✻= △$8000 △BYTE △'M37481E8T-'	✻=△$0000 △BYTE △'M37481E8T-'

Note : If the name of the product written to the EPROMs does not match the name of the ROM programming confirmation form, the ROM processing is disabled. Write the data correctly.

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Please submit the shrink DIP package Mark Specification Form (only for built-in One Time PROM microcomputer) for the M37481E8T-XXXSP or the 44P6N-A Mark Specification Form for the M37481E8T-XXXFP.

✵ 3. Comments

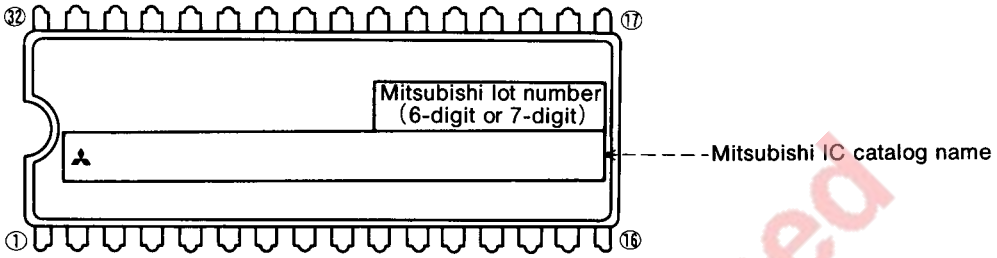
3.4 Mark Specification Forms

32P4B (32-PIN SHRINK DIP) MARK SPECIFICATION FORM

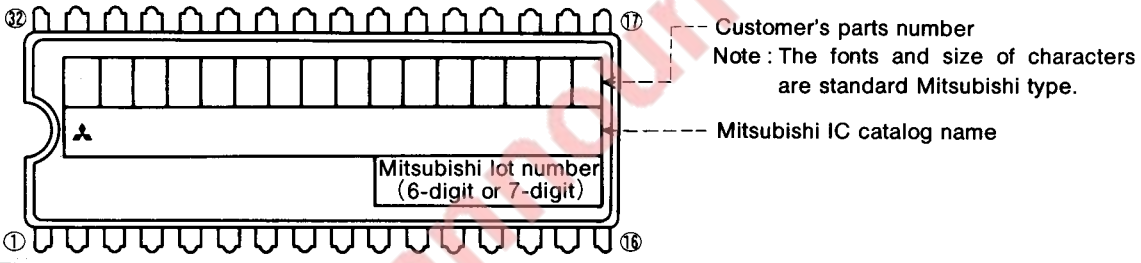
Mitsubishi IC catalog name

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

A. Standard Mitsubishi Mark

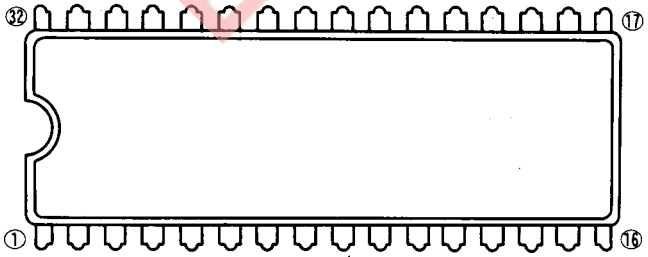


B. Customer's Parts Number + Mitsubishi Catalog Name



Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 16 characters :
Only 0~9, A~Z, +, −, /, (,), &, ©, . (period), and , (comma) are usable.
4 : If the Mitsubishi logo is not required, check the box on the right. Mitsubishi logo is not required

C. Special Mark Required



Note1 : If the special mark is to be printed, indicate the desired layout of the mark in the upper figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit or 7-digit) and mask ROM number (3-digit) are always marked.
2 : If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special logo required

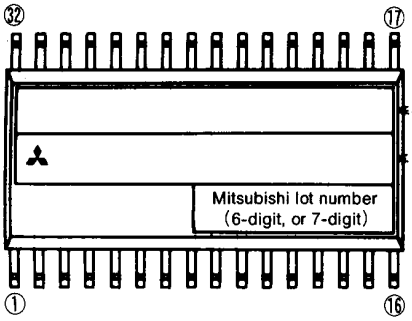
The standard Mitsubishi font is used for all characters except for a logo.

32P2W (32-PIN SOP) MARK SPECIFICATION FORM

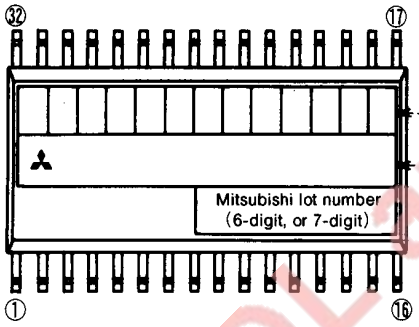
Mitsubishi IC catalog name

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

A. Standard Mitsubishi Mark



B. Customer's Parts Number + Mitsubishi catalog name



Customer's Parts Number

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

Mitsubishi IC catalog name

Note1 : The mark field should be written right aligned.

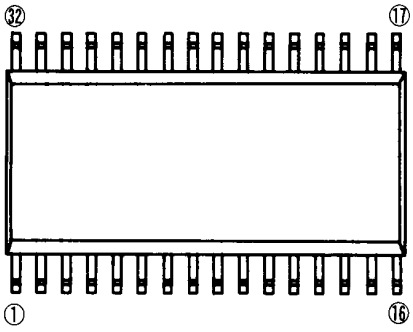
2 : The fonts and size of characters are standard Mitsubishi type.

3 : Customer's Parts Number can be up to 13 characters : Only 0~9, A~Z, +, −, /, (,), &, ©, ・ (periods), , (commas) are usable.

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

Mitsubishi logo is not required ☐

C. Special Mark Required



Note1 : If the Special Mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit, or 7-digit) and Mask ROM number (3-digit) are always marked.

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

Special logo required ☐

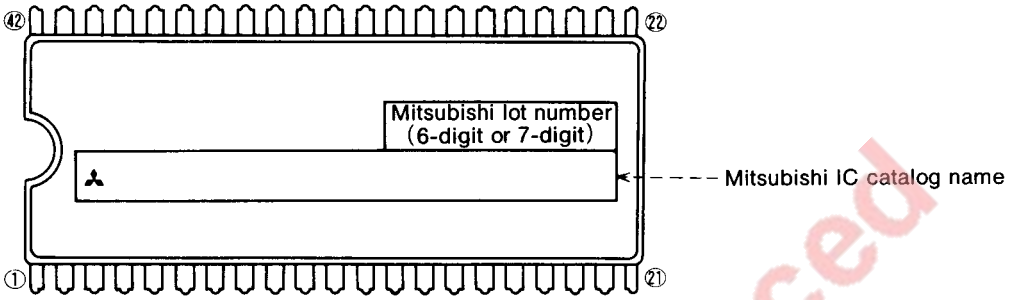
3 : The standard Mitsubishi font is used for all characters except for a logo.

42P4B (42-PIN SHRINK DIP) MARK SPECIFICATION FORM

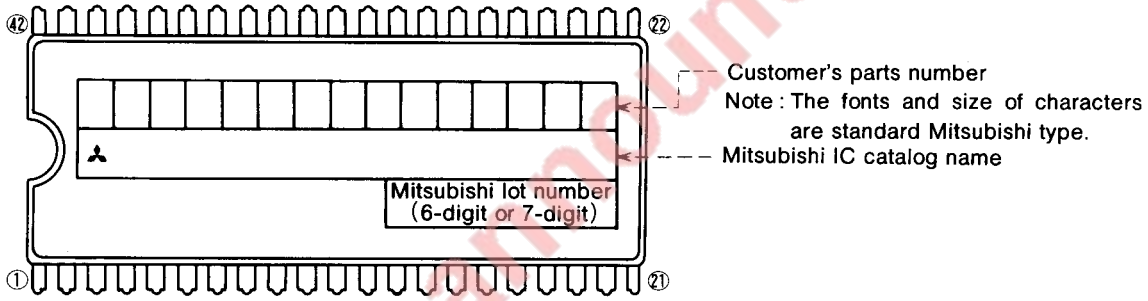
Mitsubishi IC catalog name

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

A. Standard Mitsubishi Mark

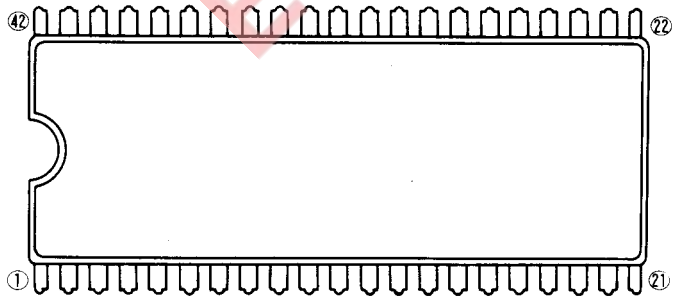


B. Customer's Parts Number + Mitsubishi Catalog Name



Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 15 characters :
Only 0~9, A~Z, +, −, /, (,), &, ©, . (period), and , (comma) are usable.
4 : If the Mitsubishi logo is not required, check the box on the right. Mitsubishi logo is not required

C. Special Mark Required



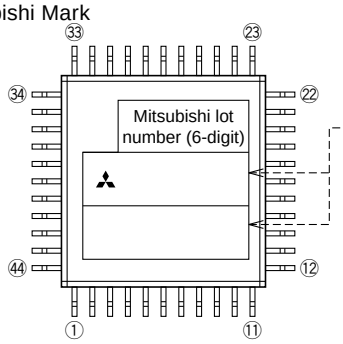
Note1 : If the special mark is to be printed, indicate the desired layout of the mark in the upper figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit or 7-digit) and mask ROM number (3-digit) are always marked.
2 : If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.
Special logo required
The standard Mitsubishi font is used for all characters except for a logo.

44P6N (44-PIN QFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

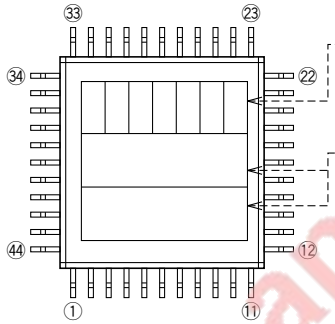
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

B. Customer's Parts Number + Mitsubishi Catalog Name



Customer's parts number
Note : The fonts and size of characters are standard Mitsubishi type.

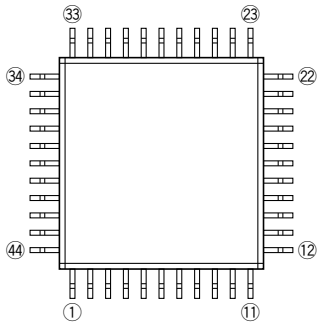
Mitsubishi IC catalog name and Mitsubishi lot number

Note4 : If the Mitsubishi logo is not required, check the box below.

Mitsubishi logo is not required. ☐

Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 7 characters : Only 0 ~ 9, A ~ Z,+,–,/, (,), &, ©, . (period), and , (comma) are usable.

C. Special Mark Required



Note1 : If the special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit) and mask ROM number (3-digit) are always marked.
2 : If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo. For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special logo required ☐

The standard Mitsubishi font is used for all characters except for a logo.

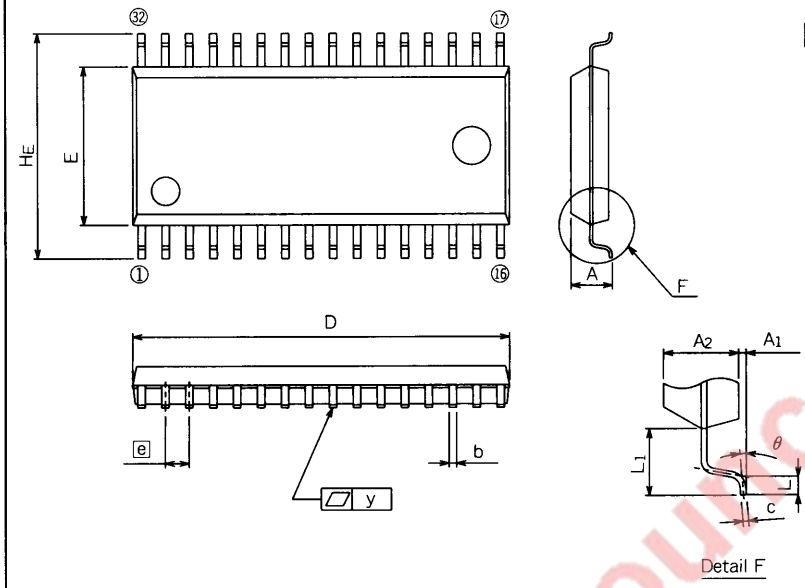
3.5 Package Outlines

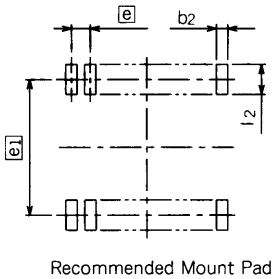
32P2W–A

Plastic 32pin 450mil SOP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SOP32-P-450-1.27	—	0.67	Alloy 42

Scale : 2.5/1





Recommended Mount Pad

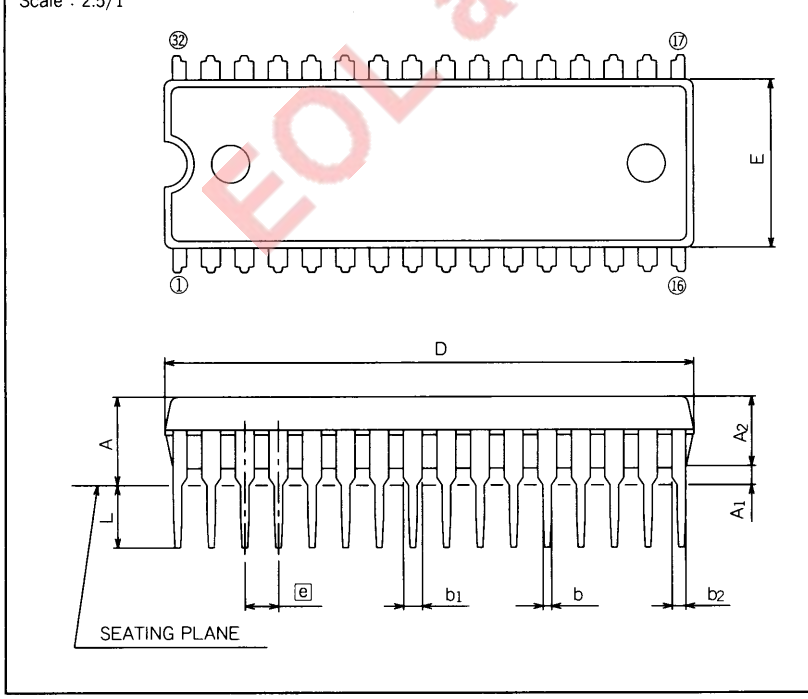
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	2.4
A_1	0.05	—	—
A_2	—	2.0	—
b	0.35	0.4	0.5
c	0.13	0.15	0.2
D	19.8	20.0	20.2
E	8.2	8.4	8.6
e	—	1.27	—
H_E	11.63	11.93	12.23
L	0.3	0.5	0.7
L_1	—	1.765	—
y	—	—	0.15
θ	0°	—	10°
b_2	—	0.76	—
e_1	—	11.43	—
l_2	1.27	—	—

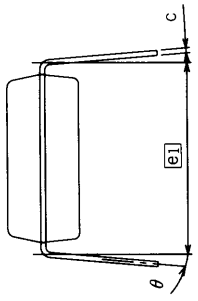
32P4B

Plastic 32pin 400mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP32-P-400-1.78	—	2.2	Alloy 42/Cu Alloy

Scale : 2.5/1





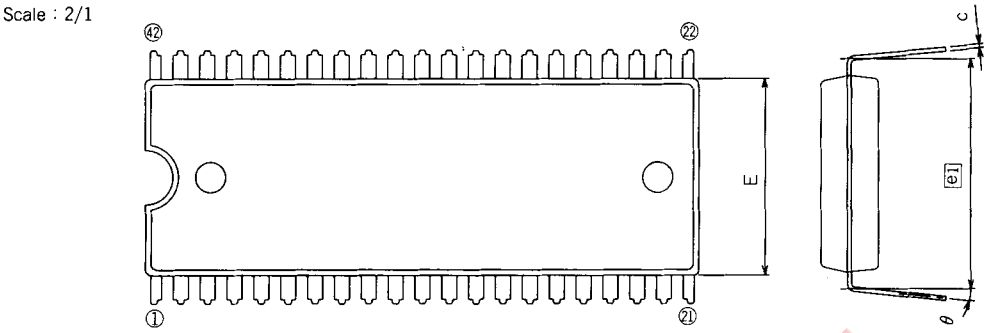
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	5.08
A_1	0.51	—	—
A_2	—	3.8	—
b	0.35	0.45	0.55
b_1	0.9	1.0	1.3
b_2	0.63	0.73	1.03
c	0.22	0.27	0.34
D	27.8	28.0	28.2
E	8.75	8.9	9.05
e	—	1.778	—
e_1	—	10.16	—
L	3.0	—	—
θ	0°	—	15°

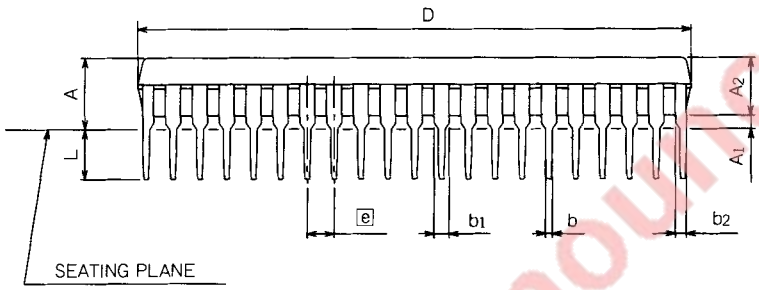
42P4B

Plastic 42pin 600mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP42-P-600-1.78	—	4.1	Alloy 42/Cu Alloy

Scale : 2/1





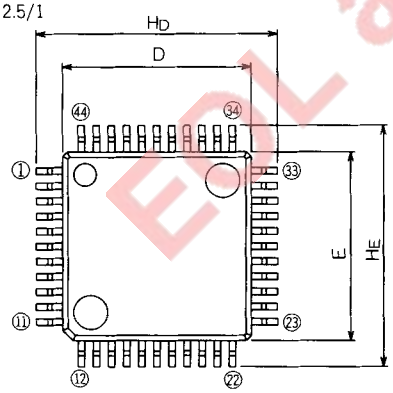
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	5.5
A_1	0.51	—	—
A_2	—	3.8	—
b	0.35	0.45	0.55
b_1	0.9	1.0	1.3
b_2	0.63	0.73	1.03
c	0.22	0.27	0.34
D	36.5	36.7	36.9
E	12.85	13.0	13.15
e	—	1.778	—
e1	—	15.24	—
L	3.0	—	—
θ	0°	—	15°

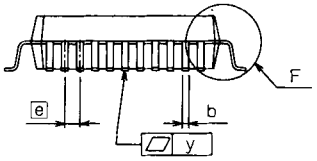
44P6N-A

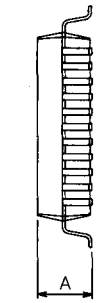
Plastic 44pin 10×10mm body QFP

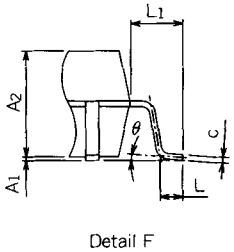
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP44-P-1010-0.80	—	0.59	Alloy 42

Scale : 2.5/1

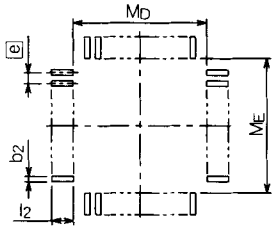








Detail F



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	3.05
A_1	0	0.1	0.2
A_2	—	2.8	—
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	9.8	10.0	10.2
E	9.8	10.0	10.2
e	—	0.8	—
HD	12.5	12.8	13.1
HE	12.5	12.8	13.1
L	0.4	0.6	0.8
L_1	—	1.4	—
y	—	—	0.1
θ	0°	—	10°
b_2	—	0.5	—
l_2	1.3	—	—
MD	—	10.6	—
ME	—	10.6	—

3.6 Machine instructions

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
ADC (Note 1) (Note 5)	When T = 0 A ← A + M + C When T = 1 M(X) ← M(X) + M + C	Adds the carry, accumulator and memory contents. The results are entered into the accumulator. Adds the contents of the memory in the address indicated by index register X, the contents of the memory specified by the addressing mode and the carry. The results are entered into the memory at the address indicated by index register X.				69	2	2							65	3	2			
AND (Note 1)	When T = 0 A ← A ∧ M When T = 1 M(X) ← M(X) ∧ M	"AND's" the accumulator and memory contents. The results are entered into the accumulator. "AND's" the contents of the memory of the address indicated by index register X and the contents of the memory specified by the addressing mode. The results are entered into the memory at the address indicated by index register X.				29	2	2							25	3	2			
ASL	7 0 C←□←0	Shifts the contents of accumulator or contents of memory one bit to the left. The low order bit of the accumulator or memory is cleared and the high order bit is shifted into the carry flag.							0A	2	1				06	5	2			
BBC (Note 4)	Ab or Mb = 0?	Branches when the contents of the bit specified in the accumulator or memory is "0".										13+20i	4	2				17+20i	5	3
BBS (Note 4)	Ab or Mb = 1?	Branches when the contents of the bit specified in the accumulator or memory is "1".										03+20i	4	2				07+20i	5	3
BCC (Note 4)	C = 0?	Branches when the contents of carry flag is "0".																		
BCS (Note 4)	C = 1?	Branches when the contents of carry flag is "1".																		
BEQ (Note 4)	Z = 1?	Branches when the contents of zero flag is "1".																		
BIT	A ∧ M	"AND's" the contents of accumulator and memory. The results are not entered anywhere.													24	3	2			
BMI (Note 4)	N = 1?	Branches when the contents of negative flag is "1".																		
BNE (Note 4)	Z = 0?	Branches when the contents of zero flag is "0".																		
BPL (Note 4)	N = 0?	Branches when the contents of negative flag is "0".																		
BRA	PC ← PC ± offset	Jumps to address specified by adding offset to the program counter.																		
BRK	B ← 1 M(S) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 M(S) ← PS S ← S – 1 PC_L ← AD_L PC_H ← AD_H	Executes a software interrupt.	00	7	1															

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2							N	V	•	•	•	•	Z	C
35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2							N	•	•	•	•	•	Z	•
16	6	2				0E	6	3	1E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																											90	2	2				•	•	•	•	•	•	•	•
																											B0	2	2				•	•	•	•	•	•	•	•
																											F0	2	2				•	•	•	•	•	•	•	•
						2C	4	3																									M7	M6	•	•	•	•	Z	•
																											30	2	2				•	•	•	•	•	•	•	•
																											D0	2	2				•	•	•	•	•	•	•	•
																											10	2	2				•	•	•	•	•	•	•	•
																											80	4	2				•	•	•	•	•	•	•	•
																																	•	•	•	1	•	1	•	•

3.6 Machine Instructions

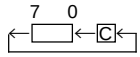
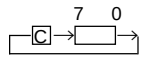
Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
BVC (Note 4)	V = 0?	Branches when the contents of overflow flag is "0".																		
BVS (Note 4)	V = 1?	Branches when the contents of overflow flag is "1".																		
CLB	Ab or Mb ← 0	Clears the contents of the bit specified in the accumulator or memory to "0".										1B+20i	2	1				1F+20i	5	2
CLC	C ← 0	Clears the contents of the carry flag to "0".	18	2	1															
CLD	D ← 0	Clears the contents of decimal mode flag to "0".	D8	2	1															
CLI	I ← 0	Clears the contents of interrupt disable flag to "0".	58	2	1															
CLT	T ← 0	Clears the contents of index X mode flag to "0".	12	2	1															
CLV	V ← 0	Clears the contents of overflow flag to "0".	B8	2	1															
CMP (Note 3)	When T = 0 A – M When T = 1 M(X) – M	Compares the contents of accumulator and memory. Compares the contents of the memory specified by the addressing mode with the contents of the address indicated by index register X.				C9	2	2							C5	3	2			
COM	M ← $\overline{M}$	Forms a one's complement of the contents of memory, and stores it into memory.													44	5	2			
CPX	X – M	Compares the contents of index register X and memory.				E0	2	2							E4	3	2			
CPY	Y – M	Compares the contents of index register Y and memory.				C0	2	2							C4	3	2			
DEC	A ← A – 1 or M ← M – 1	Decrements the contents of the accumulator or memory by 1.							1A	2	1				C6	5	2			
DEX	X ← X – 1	Decrements the contents of index register X by 1.	CA	2	1															
DEY	Y ← Y – 1	Decrements the contents of index register Y by 1.	88	2	1															
DIV	A ← (M(zz + X + 1), M(zz + X)) / A M(S) ← 1's complememt of Remainder S ← S – 1	Divides the 16-bit data that is the contents of M (zz + x + 1) for high byte and the contents of M (zz + x) for low byte by the accumulator. Stores the quotient in the accumulator and the 1's complement of the remainder on the stack.																		
EOR (Note 1)	When T = 0 A ← A ∀ M When T = 1 M(X) ← M(X) ∀ M	"Exclusive-ORs" the contents of accumulator and memory. The results are stored in the accumulator. "Exclusive-ORs" the contents of the memory specified by the addressing mode and the contents of the memory at the address indicated by index register X. The results are stored into the memory at the address indicated by index register X.				49	2	2							45	3	2			
INC	A ← A + 1 or M ← M + 1	Increments the contents of accumulator or memory by 1.							3A	2	1				E6	5	2			
INX	X ← X + 1	Increments the contents of index register X by 1.	E8	2	1															
INY	Y ← Y + 1	Increments the contents of index register Y by 1.	C8	2	1															

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																											50	2	2				•	•	•	•	•	•	•	•
																											70	2	2				•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	0
																																	•	•	•	•	0	•	•	•
																																	•	•	•	•	•	0	•	•
																																	•	•	0	•	•	•	•	•
																																	•	0	•	•	•	•	•	•
D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
																																	N	•	•	•	•	•	Z	•
						EC	4	3																									N	•	•	•	•	•	Z	C
						CC	4	3																									N	•	•	•	•	•	Z	C
D6	6	2				CE	6	3	DE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
E2	16	2																															•	•	•	•	•	•	•	•
55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•

3.6 Machine Instructions

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
JMP	If addressing mode is ABS $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is IND $PC_L \leftarrow M(AD_H, AD_L)$ $PC_H \leftarrow M(AD_H, AD_L + 1)$ If addressing mode is ZP, IND $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	Jumps to the specified address.																		
JSR	$M(S) \leftarrow PC_H$ $S \leftarrow S - 1$ $M(S) \leftarrow PC_L$ $S \leftarrow S - 1$ After executing the above, if addressing mode is ABS, $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ if addressing mode is SP, $PC_L \leftarrow AD_L$ $PC_H \leftarrow FF$ If addressing mode is ZP, IND, $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	After storing contents of program counter in stack, and jumps to the specified address.																		
LDA (Note 2)	When T = 0 $A \leftarrow M$ When T = 1 $M(X) \leftarrow M$	Load accumulator with contents of memory. Load memory indicated by index register X with contents of memory specified by the addressing mode.				A9	2	2							A5	3	2			
LDM	$M \leftarrow nn$	Load memory with immediate value.													3C	4	3			
LDX	$X \leftarrow M$	Load index register X with contents of memory.				A2	2	2							A6	3	2			
LDY	$Y \leftarrow M$	Load index register Y with contents of memory.				A0	2	2							A4	3	2			
LSR	7 0 0→□→C	Shift the contents of accumulator or memory to the right by one bit. The low order bit of accumulator or memory is stored in carry, 7th bit is cleared.							4A	2	1				46	5	2			
MUL	$M(S) \cdot A \leftarrow A \times M(zz + X)$ $S \leftarrow S - 1$	Multiplies the accumulator with the contents of memory specified by the zero page X addressing mode and stores the high byte of the result on the stack and the low byte in the accumulator.																		
NOP	$PC \leftarrow PC + 1$	No operation.	EA	2	1															
ORA (Note 1)	When T = 0 $A \leftarrow A \vee M$ When T = 1 $M(X) \leftarrow M(X) \vee M$	"Logical OR's" the contents of memory and accumulator. The result is stored in the accumulator. "Logical OR's" the contents of memory indicated by index register X and contents of memory specified by the addressing mode. The result is stored in the memory specified by index register X.				09	2	2							05	3	2			

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
						4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
						20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
			B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•
56	6	2				4E	6	3	5E	7	3																						0	•	•	•	•	•	Z	C
62	15	2																															•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
PHA	M(S) ← A S ← S – 1	Saves the contents of the accumulator in memory at the address indicated by the stack pointer and decrements the contents of stack pointer by 1.	48	3	1															
PHP	M(S) ← PS S ← S – 1	Saves the contents of the processor status register in memory at the address indicated by the stack pointer and decrements the contents of the stack pointer by 1.	08	3	1															
PLA	S ← S + 1 A ← M(S)	Increments the contents of the stack pointer by 1 and restores the accumulator from the memory at the address indicated by the stack pointer.	68	4	1															
PLP	S ← S + 1 PS ← M(S)	Increments the contents of stack pointer by 1 and restores the processor status register from the memory at the address indicated by the stack pointer.	28	4	1															
ROL		Shifts the contents of the memory or accumulator to the left by one bit. The high order bit is shifted into the carry flag and the carry flag is shifted into the low order bit.							2A	2	1				26	5	2			
ROR		Shifts the contents of the memory or accumulator to the right by one bit. The low order bit is shifted into the carry flag and the carry flag is shifted into the high order bit.							6A	2	1				66	5	2			
RRF		Rotates the contents of memory to the right by 4 bits.													82	8	2			
RTI	S ← S + 1 PS ← M(S) S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	Returns from an interrupt routine to the main routine.	40	6	1															
RTS	S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	Returns from a subroutine to the main routine.	60	6	1															
SBC (Note 1) (Note 5)	When T = 0 A ← A – M – $\overline{C}$ When T = 1 M(X) ← M(X) – M – $\overline{C}$	Subtracts the contents of memory and complement of carry flag from the contents of accumulator. The results are stored into the accumulator. Subtracts contents of complement of carry flag and contents of the memory indicated by the addressing mode from the memory at the address indicated by index register X. The results are stored into the memory of the address indicated by index register X.				E9	2	2							E5	3	2			
SEB	Ab or Mb ← 1	Sets the specified bit in the accumulator or memory to “1”.										0B + 20i	2	1				0F + 20i	5	2
SEC	C ← 1	Sets the contents of the carry flag to “1”.	38	2	1															
SED	D ← 1	Sets the contents of the decimal mode flag to “1”.	F8	2	1															
SEI	I ← 1	Sets the contents of the interrupt disable flag to “1”.	78	2	1															
SET	T ← 1	Sets the contents of the index X mode flag to “1”.	32	2	1															

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																															•	•	•	•	•	•	•	•
																															•	•	•	•	•	•	•	•
																															N	•	•	•	•	•	Z	•
																															(Value saved in stack)							
36	6	2				2E	6	3	3E	7	3																				N	•	•	•	•	•	Z	C
76	6	2				6E	6	3	7E	7	3																				N	•	•	•	•	•	Z	C
																															•	•	•	•	•	•	•	•
																															(Value saved in stack)							
																															•	•	•	•	•	•	•	•
F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
																															•	•	•	•	•	•	•	•
																															•	•	•	•	•	•	•	1
																															•	•	•	•	1	•	•	•
																															•	•	•	•	•	1	•	•
																															•	•	1	•	•	•	•	•

3.6 Machine Instructions

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
STA	M ← A	Stores the contents of accumulator in memory.													85	4	2			
STP		Stops the oscillator.	42	2	1															
STX	M ← X	Stores the contents of index register X in memory.													86	4	2			
STY	M ← Y	Stores the contents of index register Y in memory.													84	4	2			
TAX	X ← A	Transfers the contents of the accumulator to index register X.	AA	2	1															
TAY	Y ← A	Transfers the contents of the accumulator to index register Y.	A8	2	1															
TST	M = 0?	Tests whether the contents of memory are “0” or not.													64	3	2			
TSX	X ← S	Transfers the contents of the stack pointer to index register X.	BA	2	1															
TXA	A ← X	Transfers the contents of index register X to the accumulator.	8A	2	1															
TXS	S ← X	Transfers the contents of index register X to the stack pointer.	9A	2	1															
TYA	A ← Y	Transfers the contents of index register Y to the accumulator.	98	2	1															
WIT		Stops the internal clock.	C2	2	1															

Notes 1 : The number of cycles “n” is increased by 3 when T is 1.
2 : The number of cycles “n” is increased by 2 when T is 1.
3 : The number of cycles “n” is increased by 1 when T is 1.
4 : The number of cycles “n” is increased by 2 when branching has occurred.
5 : N, V, and Z flags are invalid in decimal operation mode.

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
			96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	–	Subtraction
A	Accumulator or Accumulator addressing mode	∧	Logical OR
		∨	Logical AND
BIT, A	Accumulator bit relative addressing mode	∀	Logical exclusive OR
		—	Negation
ZP	Zero page addressing mode	←	Shows direction of data flow
BIT, ZP	Zero page bit relative addressing mode	X	Index register X
		Y	Index register Y
ZP, X	Zero page X addressing mode	S	Stack pointer
ZP, Y	Zero page Y addressing mode	PC	Program counter
ABS	Absolute addressing mode	PS	Processor status register
ABS, X	Absolute X addressing mode	PC_H	8 high-order bits of program counter
ABS, Y	Absolute Y addressing mode	PC_L	8 low-order bits of program counter
IND	Indirect absolute addressing mode	AD_H	8 high-order bits of address
		AD_L	8 low-order bits of address
ZP, IND	Zero page indirect absolute addressing mode	FF	FF in Hexadecimal notation
		nn	Immediate value
IND, X	Indirect X addressing mode	M	Memory specified by address designation of any addressing mode
IND, Y	Indirect Y addressing mode		
REL	Relative addressing mode	M(X)	Memory of address indicated by contents of index register X
SP	Special page addressing mode		
C	Carry flag	M(S)	Memory of address indicated by contents of stack pointer
Z	Zero flag		
I	Interrupt disable flag	$M(AD_H, AD_L)$	Contents of memory at address indicated by AD_H and AD_L, in AD_H is 8 high-order bits and AD_L is 8 low-order bits.
D	Decimal mode flag		
B	Break flag		
T	X-modified arithmetic mode flag	$M(00, AD_L)$	Contents of address indicated by zero page AD_L
V	Overflow flag	Ab	1 bit of accumulator
N	Negative flag	Mb	1 bit of memory
		OP	Opcode
		n	Number of cycles
		#	Number of bytes

3.7 List of Instruction Codes

D7 – D4 \ D3 – D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	MUL ZP, X	BBS 3, A	TST ZP	ADC ZP	ROR ZP	BBS 3, ZP	PLA	ADC IMM	ROR A	SEB 3, A	JMP IND	ADC ABS	ROR ABS	SEB 3, ZP
0111	7	BVS	ADC IND, Y	—	BBC 3, A	—	ADC ZP, X	ROR ZP, X	BBC 3, ZP	SEI	ADC ABS, Y	—	CLB 3, A	—	ADC ABS, X	ROR ABS, X	CLB 3, ZP
1000	8	BRA	STA IND, X	RRF ZP	BBS 4, A	STY ZP	STA ZP	STX ZP	BBS 4, ZP	DEY	—	TXA	SEB 4, A	STY ABS	STA ABS	STX ABS	SEB 4, ZP
1001	9	BCC	STA IND, Y	—	BBC 4, A	STY ZP, X	STA ZP, X	STX ZP, Y	BBC 4, ZP	TYA	STA ABS, Y	TXS	CLB 4, A	—	STA ABS, X	—	CLB 4, ZP
1010	A	LDY IMM	LDA IND, X	LDX IMM	BBS 5, A	LDY ZP	LDA ZP	LDX ZP	BBS 5, ZP	TAY	LDA IMM	TAX	SEB 5, A	LDY ABS	LDA ABS	LDX ABS	SEB 5, ZP
1011	B	BCS	LDA IND, Y	JMP ZP, IND	BBC 5, A	LDY ZP, X	LDA ZP, X	LDX ZP, Y	BBC 5, ZP	CLV	LDA ABS, Y	TSX	CLB 5, A	LDY ABS, X	LDA ABS, X	LDX ABS, Y	CLB 5, ZP
1100	C	CPY IMM	CMP IND, X	WIT	BBS 6, A	CPY ZP	CMP ZP	DEC ZP	BBS 6, ZP	INY	CMP IMM	DEX	SEB 6, A	CPY ABS	CMP ABS	DEC ABS	SEB 6, ZP
1101	D	BNE	CMP IND, Y	—	BBC 6, A	—	CMP ZP, X	DEC ZP, X	BBC 6, ZP	CLD	CMP ABS, Y	—	CLB 6, A	—	CMP ABS, X	DEC ABS, X	CLB 6, ZP
1110	E	CPX IMM	SBC IND, X	DIV ZP, X	BBS 7, A	CPX ZP	SBC ZP	INC ZP	BBS 7, ZP	INX	SBC IMM	NOP	SEB 7, A	CPX ABS	SBC ABS	INC ABS	SEB 7, ZP
1111	F	BEQ	SBC IND, Y	—	BBC 7, A	—	SBC ZP, X	INC ZP, X	BBC 7, ZP	SED	SBC ABS, Y	—	CLB 7, A	—	SBC ABS, X	INC ABS, X	CLB 7, ZP

3-byte instruction
2-byte instruction
1-byte instruction

3.8 SFR Memory Map

Figure 3.8.1 shows the SFR memory map.

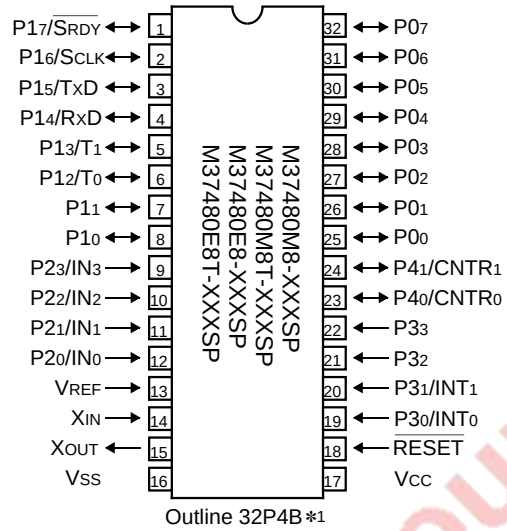
00C0₁₆	Port P0 register (P0)		00E0₁₆	Transmit/receive buffer register (TB/RB)
00C1₁₆	Port P0 direction register (P0D)		00E1₁₆	Serial I/O status register (SIOSTS)
00C2₁₆	Port P1 register (P1)		00E2₁₆	Serial I/O control register (SIOCON)
00C3₁₆	Port P1 direction register (P1D)		00E3₁₆	UART control register (UARTCON)
00C4₁₆	Port P2 register (P2)		00E4₁₆	Baud rate generator (BRG)
00C5₁₆			00E5₁₆	Bus collision detection control register (BUSARBCON)
00C6₁₆	Port P3 register (P3)		00E6₁₆	
00C7₁₆			00E7₁₆	
00C8₁₆	Port P4 register (P4)		00E8₁₆	
00C9₁₆	Port P4 direction register (P4D)		00E9₁₆	
00CA₁₆	Port P5 register (P5)	(**Note**)	00EA₁₆	
00CB₁₆	Port P5 direction register (P5D)	(**Note**)	00EB₁₆	
00CC₁₆			00EC₁₆	
00CD₁₆			00ED₁₆	
00CE₁₆			00EE₁₆	
00CF₁₆			00EF₁₆	Watchdog timer H (WDTH)
00D0₁₆	Port P0 pull-up control register (P0PCON)		00F0₁₆	Timer X low-order (TXL)
00D1₁₆	Port P1 pull-up control register (P1PCON)		00F1₁₆	Timer X high-order (TXH)
00D2₁₆	Port P4P5 input control register (P4P5CON)		00F2₁₆	Timer Y low-order (TYL)
00D3₁₆			00F3₁₆	Timer Y high-order (TYH)
00D4₁₆	Edge polarity selection register (EG)		00F4₁₆	Timer 1 (T1)
00D5₁₆			00F5₁₆	Timer 2 (T2)
00D6₁₆			00F6₁₆	Timer X mode register (TXM)
00D7₁₆			00F7₁₆	Timer Y mode register (TYM)
00D8₁₆			00F8₁₆	Timer XY control register (TXYCON)
00D9₁₆	A-D control register (ADCON)		00F9₁₆	Timer 1 mode register (T1M)
00DA₁₆	A-D conversion register (AD)		00FA₁₆	Timer 2 mode register (T2M)
00DB₁₆			00FB₁₆	CPU mode register (CPUM)
00DC₁₆			00FC₁₆	Interrupt request register 1 (IREQ1)
00DD₁₆			00FD₁₆	Interrupt request register 2 (IREQ2)
00DE₁₆	**STP** instruction operation control register (STPCON)		00FE₁₆	Interrupt control register 1 (ICON1)
00DF₁₆			00FF₁₆	Interrupt control register 2 (ICON2)

Note: These registers are not allocated in the 7480 Group.

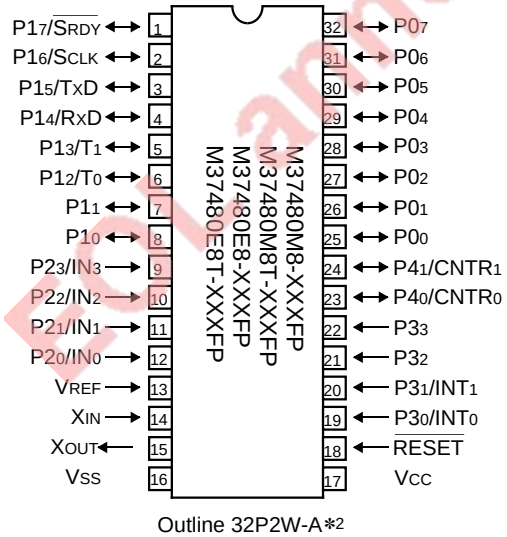
Figure 3.8.1 SFR Memory Map

3.9 Pinouts

Figures 3.9.1 and 3.9.2 show the pinouts of the 7480 Group and 7481 Group.



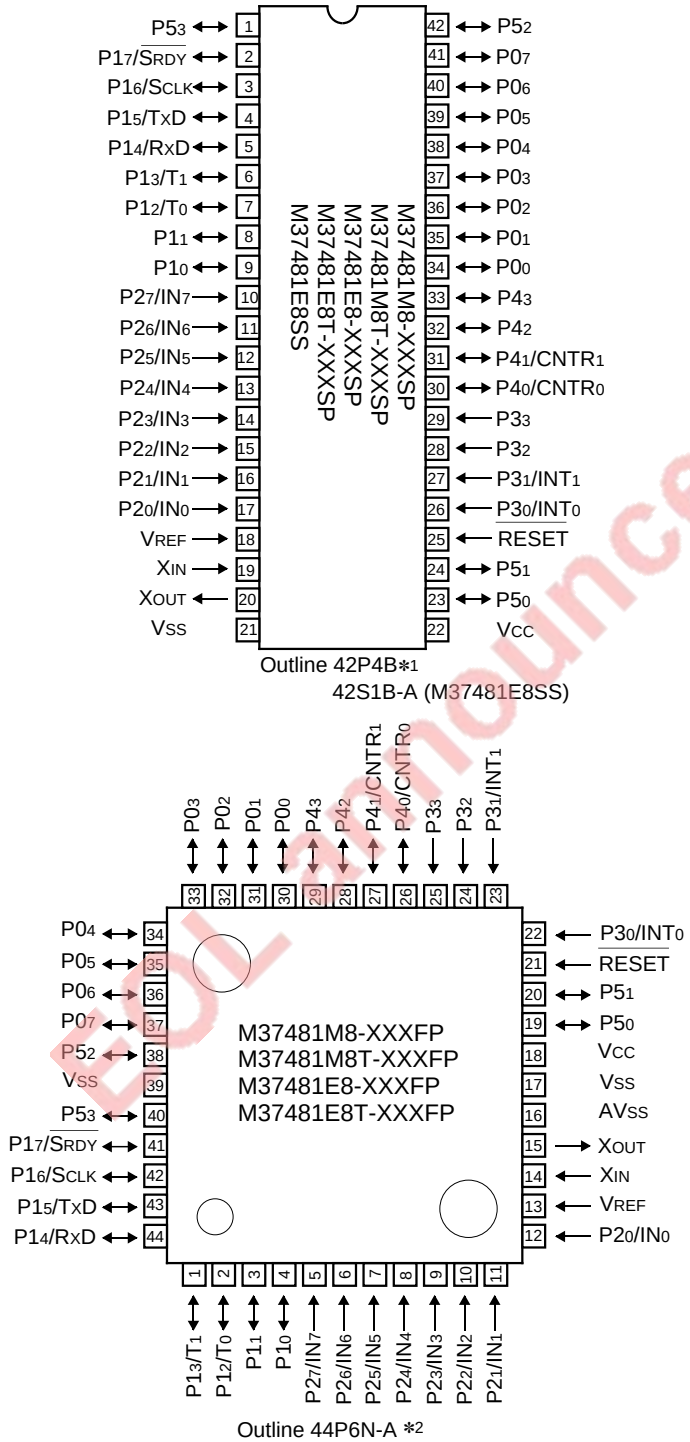
Outline 32P4B ∗1



Outline 32P2W-A ∗2

∗ **1:** The M37480M2T-XXXSP, M37480M4-XXXSP and M37480M4T-XXXSP are also included in the 32P4B packages, respectively. All of these products are pin-compatible.
2: The M37480M2T-XXXFP, M37480M4-XXXFP and M37480M4T-XXXFP are also included in the 32P2W-A packages, respectively. All of these products are pin-compatible.
Note: The only differences between the 32P4B package product and the 32P2W-A package product are package outline and absolute maximum ratings.

Figure 3.9.1 Pinout of 7480 Group (top view)



*** 1:** The M37481M2T-XXXSP, M37481M4-XXXSP and M37481M4T-XXXSP are also included in the 42P4B packages, respectively. All of these products are pin-compatible.

2: The M37481M2T-XXXFP, M37481M4-XXXFP and M37481M4T-XXXFP are also included in the 44P6N-A packages, respectively. All of these products are pin-compatible.

Note: The only differences between the 42P4B package product and the 44P6N-A package product are package outline, absolute maximum ratings and the fact that the 44P6N-A package product has the AVss pin.

Figure 3.9.2 Pinout of 7481 Group (top view)

This page left blank intentionally.

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
7480 Group
7481 Group

Nov. First Edition 1997

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

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

User's Manual
7480 Group
7481 Group

Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

Printed in Japan (ROD)

New publication, effective Nov. 1997.
Specifications subject to change without notice.

REVISION DESCRIPTION LIST	7480 GROUP AND 7481 GROUP USER'S MANUAL

Rev. No.	Revision Description	Rev. date
1.0	First Edition	971130

GRADE A

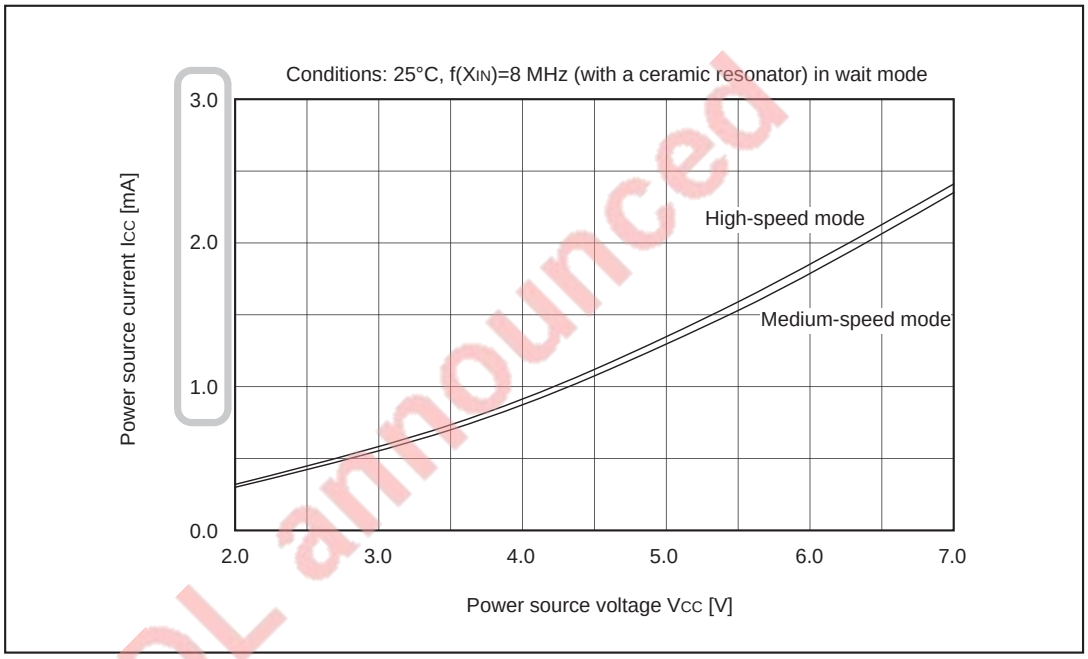
MESC TECHNICAL NEWS

No.M740-14-9712

Additional information of 7480/7481 Group (Rev.A)

The following errors exist in the 7480 Group and 7481 Group User's Manual.
Please refer to the corrected information as shown below.

Corrected information of 7480 Group and 7481 Group User's Manual (Rev.A)

Page	Error	Correct
1-192 Figure 1.21.5	 : Corrected points	
2-24 Section 2.3.3 SPECIFICATIONS	●LAN communication format: Simplified SAE J1850 (PWM system)	●LAN communication format: Simplified SAE* J1850 (PWM system) *SAE: Society of Automotive Engineers