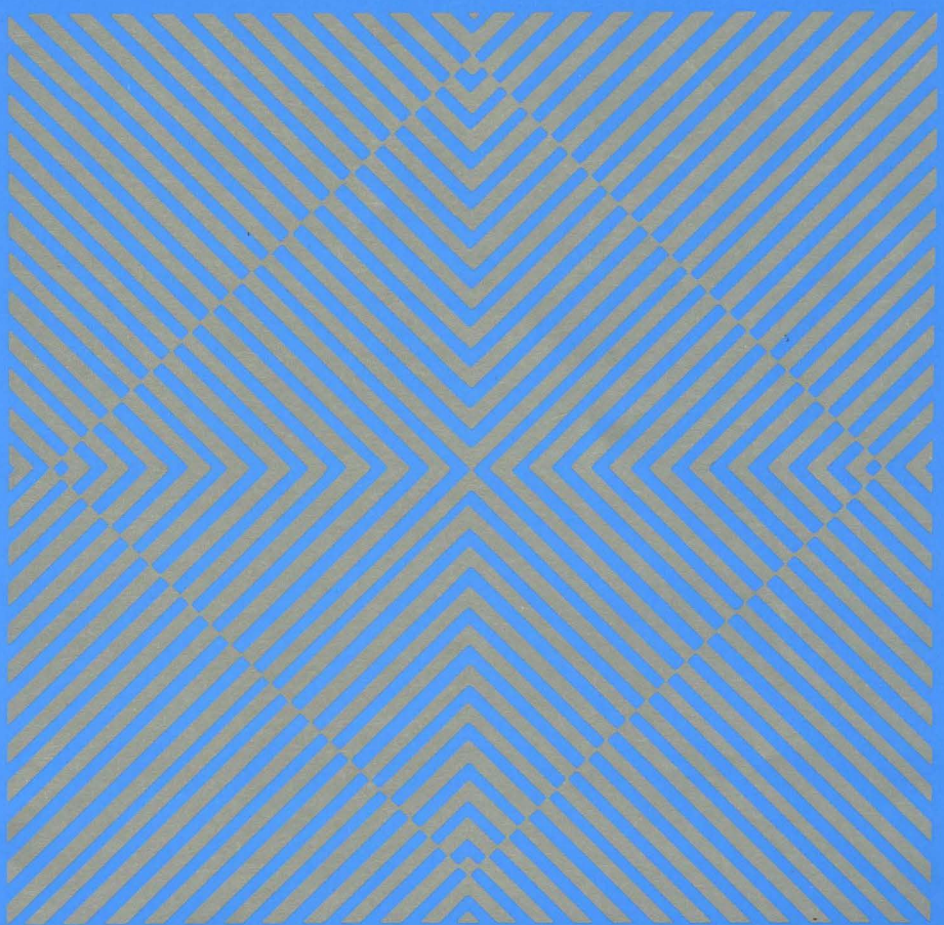
MITSUBISHI SEMICONDUCTORS

M37700M2-XXXFP
M37700M2AXXXFP

USER'S MANUAL



MITSUBISHI ELECTRIC

USER'S MANUAL
M37700M2-XXXFP
M37700M2AXXXFP

Difference between Original M37700 and Revised Version (Version –A)

Difference between Original M37700 and Revised Version (Version –A)

For the M37700/M37701 series, original version and –A version are available for the following 14 product types:

M37700SFP/M37700SAFP
M37700S4FP/M37700S4AFP
M37700E4FS/M37700E4AFS
M37700E2FP/M37700E2AFP
M37700E4FP/M37700E4AFP
M37701E2SP/M37701E2ASP
M37700E2FS/M37700E2AFS

The –A version are indicated by the suffix "–A" following the product type.

	Original product	–A version
Product type	▲M37700SFP XXXXXX	▲M37700SFP–A XXXXXX

The differences between the original version and the –A version are as follows:

(1)HOLD function

●Original product

The HOLD function does not work properly. Therefore, in memory expansion mode and microprocessor mode, the $P4_0/\overline{HOLD}$ pin must be held at "H" level to prevent the use of the HOLD function. In single-chip mode, this pin can be a normal port $P4_0$.

●–A version

The HOLD function can be used properly.

(2)$P4_2$/ø function

●Original product

When ø output is selected from the $P4_2$/ø pin in memory expansion mode and microprocessor mode, the ø output stops at "L" level when the $P4_1/\overline{RDY}$ pin is pulled to "L" level.

●–A version

When ø output is selected from the $P4_2$/ø pin in memory expansion mode and microprocessor mode, the ø output does not stop at "L" level when the $P4_1/\overline{RDY}$ pin is pulled to "L" level.

(3)PWM output function of the timer A

●Original product

Pulse width must be changed while the PWM output is at "L". If write is performed while the PWM output as "H", the PWM output frequency temporarily changes.

●–A version

Pulse width can be changed at any time. The PWM output frequency does not change when write is performed while the PWM output is "H". However, the "L" interval (width) of the PWM output must be at least two timer clock source cycles. In other words, the value that can be set in the timer is 00_{16} to FD_{16} for 8-bit PWM mode and 0000_{16} to $FFFD_{16}$ in 16-bit PWM mode.

(4)Reading from processor mode register ($5E_{16}$) and one-shot start flag (42_{16})

●Original product

The software reset bit in the processor mode register and the bits corresponding to timers A0 to A4 in the one-shot start flag are unpredictable when read. Therefore, read-modify-write type instructions cannot used for these addresses.

●–A version

The software reset bit in the processor mode register and the bits corresponding to timers A0 to A4 in the one-shot start flag returns a "0" when read. Therefore, read-modify-write type instructions can be used for the addresses.

Preface

This manual describes the hardware of the Mitsubishi CNOS 16-bit microcomputer M37700M2-XXXFP. After reading this manual, the user should be able to fully utilize the functions of the M37700 family of microcomputers.

For details concerning the softwares for the M37700 family, refer to the MELPS 7700 software manual. For details concerning the development support tools (assembler, option boards), refer to the respective operation manuals.

Table of Contents

CHAPTER 1. DESCRIPTION

1.1 Description 2
1.2 M37700 Family
1.2.1 M37700 family 3
1.2.2 Difference between types 4
1.2.3 Meaning of type name 5
1.3 Performance Overview 6
1.4 Pin Configuration 7
1.5 Pin Description 8
1.6 Block Diagram 10

CHAPTER 2. FUNCTIONAL DESCRIPTION

2.1 Central Processing Unit (CPU)
2.1.1 Accumulator (Acc) 12
2.1.2 Index register X 12
2.1.3 Index register Y 12
2.1.4 Stack pointer (S) 14
2.1.5 Program counter (PC) 14
2.1.6 Program bank register (PG) 14
2.1.7 Data bank register (DT) 14
2.1.8 Direct page register (DPR) 15
2.1.9 Processing status register (PS) 16
2.2 Internal Bus Interface
2.2.1 Internal bus interface overview 18
2.2.2 Bus interface unit functions 18
2.2.3 Bus interface unit operations 20
2.2.4 Data read/write operations 22
2.3 Addressable memory space
2.3.1 Banks 23
2.3.2 Direct page 23
2.4 Memory Allocation
2.4.1 Internal memory and peripheral device memory allocation 24
2.4.2 Processor modes 26
2.4.3 External memory area bus control 29
2.5 Input/Output Pins
2.5.1 Programmable I/O ports 30
2.5.2 Pin functions 32
2.5.3 Single-chip mode pin functions 36
2.5.4 Memory expansion and microprocessor mode pin functions 38
2.6 Interrupts
2.6.1 Interrupt functions 42
2.6.2 Source of interrupts 43
2.6.3 Interrupt control 45
2.6.4 Interrupt order 48
2.6.5 Interrupt priority detection circuit 49
2.6.6 Interrupt priority detection time 51
2.6.7 Interrupt processing sequence 52
2.6.8 Interrupt service routine 53
2.6.9 Returning from an interrupt service routine 53
2.6.10 Interrupt response time 54

2.7 Timer A
2.7.1 Timer A description 56
2.7.2 Block description 57
2.7.3 Timer mode 62
2.7.4 Event counter mode 66
2.7.5 One-shot pulse mode 70
2.7.6 PWM (Pulse Width Modulation) mode 74
2.8 Timer B
2.8.1 Timer B description 78
2.8.2 Block diagram 78
2.8.3 Timer mode 82
2.8.4 Event counter mode 84
2.8.5 Pulse cycle/pulse width measurement mode 86
2.9 Serial I/O
2.9.1 Serial I/O description 89
2.9.2 Block description 90
2.9.3 Serial I/O operation mode selection 99
2.9.4 Clock synchronous serial I/O 100
2.9.5 Clock asynchronous serial I/O (UART) 107
2.10 A-D Converter
2.10.1 A-D converter description 118
2.10.2 Block description 120
2.10.3 A-D conversion mode operation 124
2.11 Watchdog Timer
2.11.1 Watchdog timer description 126
2.11.2 Operation description 127
2.11.3 Watchdog timer usage 127

CHAPTER 3. RESET
3.1 Reset
3.1.1 Reset operation 130
3.1.2 Internal states at reset 131
3.2 Reset circuit 134
3.3 Software reset 135

CHAPTER 4. OSCILLATING CIRCUIT
4.1 Oscillating Circuit
4.1.1 Circuit using a ceramic oscillator or a crystal oscillator 138
4.1.2 External clock input circuit 138
4.2 Clock Generation Circuit 140

CHAPTER 5. ELECTRICAL CHARACTERISTICS
5.1 Absolute Maximum Ratings 142
5.2 Recommended Operating Conditions 143
5.3 M37700M2-XXXFP DC/AC Characteristics
5.3.1 Electrical characteristics 144
5.3.2 A-D converter characteristics 144
5.3.3 Timing requirements 145
5.3.4 Switching characteristics 148
5.4 M37700M2AXXXFP DC/AC Characteristics
5.4.1 Electrical characteristics 150
5.4.2 A-D converter characteristics 150
5.4.3 Timing requirements 151
5.4.4 Switching characteristics 154
5.5 Timing Diagrams 156

CHAPTER 6. STANDARD CHARACTERISTICS
6.1 Standard Port Characteristics 162
6.2 Icc–f(XIN) Standard Characteristics 163
6.3 A-D Converter Standard Characteristics 164

CHAPTER 7. USAGE PRECAUTIONS
7.1 Software
7.1.1 Stack pointer (S) 166
7.1.2 Program bank register (PG) and data bank register (DT) 166
7.1.3 Direct page register (DPR) 166
7.1.4 Processor status register (PS) 166
7.1.5 Register save and restore instructions 166
7.1.6 PSH instruction 169
7.1.7 BRK instruction 169
7.1.8 BRA instruction 169
7.1.9 Instruction execution time (instruction execution cycle) 169
7.2 Hardware
7.2.1 Memory related features 170
7.2.2 Input/output pin related features 170
7.2.3 Interrupts 171
7.2.4 Timers related features 171
7.2.5 Serial I/O 176
7.2.6 A-D conversion 176
7.2.7 Watchdog timer 177
7.2.8 Reset related features 177
7.2.9 Microprocessor status during stop, wait, one-shot wait, ready, and hold 177

CHAPTER 8.TREATMENT OF INTERNAL PROM TYPE
8.1 Description 180
8.2 Functional Description 180
8.3 EPROM Mode 182
8.4 Fast Programming Algorithm 184
8.5 Notes
8.5.1 Notes on an built-in PROM types 186
8.5.2 Notes on one-time PROM type 186
8.5.3 Notes on EPROM type 186

CHAPTER 9.M37701M2-XXXSP
9.1 Description 188
9.2 M37701 Family 189
9.3 M37701M2-XXXSP Characterictics 189
9.4 M37701M2-XXXSP Performance Overview 190
9.5 Diferences Between M37701M2-XXXSP and M37700M2-XXXFP 191
9.6 M37701 Functional Description
9.6.1 A-D converter 192
9.6.2 Timers 192
9.6.3 Serial I/O 192
9.6.4 Ports 192

TABLE OF CONTENTS

APPENDIX 1. OUTLINE DRAWING

1.80-pin plastic molded QFP (80P6) 194
2.80-pin ceramic LCC (80D0O) 194
3.Outline of EPROM Type IC Socket 195
4.Outline of EPROM Type IC Socket 196
5.Outline of One-time PROM Type IC Socket 198
6.Common Printed Circuit Foot Pattern for 80P6, 80P6N, LCC IC Sockets 198

APPENDIX 2. SETTING OF UNUSED PINS

1.Setting of unused pins in single-chip mode 200
2.Setting of unused pins in memory expansion and microprocessor mode 200

APPENDIX 3. M37700 FAMILY PERFORMANCE OVERVIEW

1.M37700 family common performance 202
2.Difference in external clock input frequency by type 202
3.Memory characteristics and size by type 203
4.External ROM types 203
5.Package 203

APPENDIX 4. M37700 FAMILY MEMORY MAP

1.Memory map in single-chip mode 206
2.Memory map in memory expansion mode 207
3.Memory map in microprocessor mode (and external ROM type) 208

APPENDIX 5. SFR AREA MEMORY MAP 210

APPENDIX 6. CONTROL REGISTERS

1.Port Pi data direction registers (i=0~8) 214
2.A-D control register 215
3.UARTi transmit/receive mode registers (i=0, 1) 215
4.UARTi transmit/receive control register 0 (i=0, 1) 216
5.UARTi transmit/receive control register 1 (i=0, 1) 216
6.Count start flag 217
7.One-shot start flag 217
8.Up-down flag 217
9.Timer Ai mode registers (i=0~4)
(1) Timer mode 219
(2) Event counter mode 220
(3) One-shot pulse mode 220
(4) PWM mode 221
10.Timer Bi mode registers (i=0~2)
(1) Timer mode 222
(2) Event counter mode 223
(3) Pulse cycle/pulse width measurement mode 223
11.Processor mode register 224
12.Interrupt control registers
A-D conversion 225
UART0/1 transmission 225
UART0/1 receive 225
timers A0~A4 225
timers B0~B2 225
13.INT0~INT2 interrupt control registers 225

APPENDIX 7.INSTRUCTION CODE TABLE 228

APPENDIX 8.MACHINE INSTRUCTIONS 232

APPENDIX 9.MASK ROM ORDERING METHOD 248

APPENDIX 10.Q & A 250

CHAPTER 1
DESCRIPTION

CHAPTER 1.DESCRIPTION

1.1 Description

The M37700M2-XXXFP is a 16-bit single-chip microcomputer designed with high-performance CMOS silicon gate technology. It is housed in an 80-pin plastic molded flat package.

This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controllers that require high-speed processing of large amounts of data.

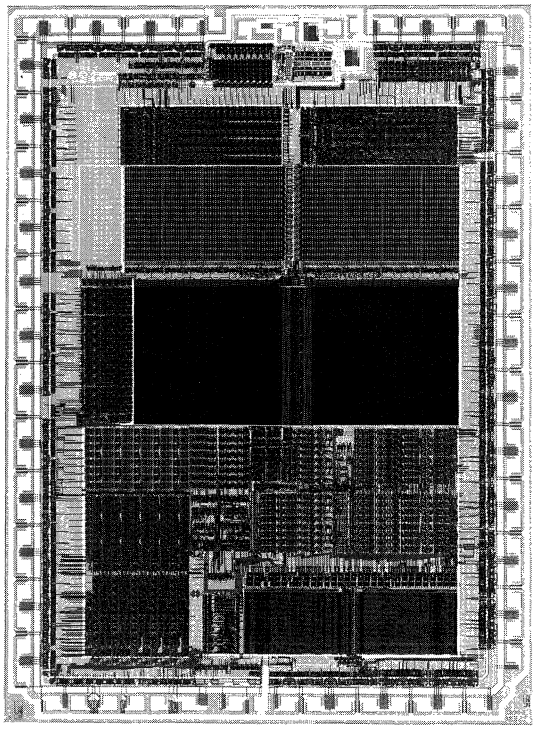


Photo of M37700M2-XXXFP Chip

1.2 M37700 Family

1.2.1 M37700 family

The M37700 family consists of chips shown in Table 1.2.1 with the M37700M2-XXXFP as the base chip. These chips are all pin compatible and provide a variety of memory characteristics, memory size, and operating clock frequencies to enable the user to select the chip best suited for his system. Hereafter, the M37700 family microcomputers will be referred to simply as the M37700 unless there is a specific difference by type.

Table 1.2.1 M37700 Family

Type name	ROM size (bytes)	RAM size (bytes)	Clock frequency (MHz)	Remarks
M37700M2-XXXFP	16K (Mask ROM)	512	8	
M37700M2AXXXFP	16K (Mask ROM)	512	16	High-speed type of M37700M2-XXXFP
M37700SFP	—	512	8	External ROM type of M37700M2-XXXFP
M37700SAFP	—	512	16	External ROM type of M37700M2AXXXFP
M37700E2-XXXFP	16K (One-time PROM)	512	8	One-time PROM type of M37700M2-XXXFP
M37700E2AXXXFP	16K (One-time PROM)	512	16	One-time PROM type of M37700M2AXXXFP
M37700E2FS	16K (EPROM)	512	8	EPROM type of M37700M2-XXXFP
M37700E2AFS	16K (EPROM)	512	16	EPROM type of M37700M2AXXXFP
M37700M4-XXXFP	32K (Mask ROM)	2048	8	Memory expansion type of M37700M2-XXXFP
M37700M4AXXXFP	32K (Mask ROM)	2048	16	Memory expansion type of M37700M2AXXXFP
M37700S4FP	—	2048	8	External ROM type of M37700M4-XXXFP
M37700S4AFP	—	2048	16	External ROM type of M37700M4AXXXFP
M37700E4-XXXFP	32K (One-time PROM)	2048	8	One-time PROM type of M37700M4-XXXFP
M37700E4AXXXFP	32K (One-time PROM)	2048	16	One-time PROM type of M37700M4AXXXFP
M37700E4FS	32K (EPROM)	2048	8	EPROM type of M37700M4-XXXFP
M37700E4AFS	32K (EPROM)	2048	16	EPROM type of M37700M4AXXXFP

1.2.2 Differences between types

① Mask ROM type microcomputer
This type of microcomputers has programs printed using a mask during the manufacturing process.

- Applicable types: M37700M2-XXXFP/M37700M2AXXXFP
  M37700M4-XXXFP/M37700M4AXXXFP

② External ROM type microcomputer
This type of microcomputers has no internal ROM. External ROMs must be provided. The function is identical with the mask ROM version microcomputer when it is operating in microprocessor mode.

- Applicable types: M37700SFP/M37700SAFP
  M37700S4FP/M37700S4AFP

③ One-time PROM type microcomputer
This type of microcomputer is equipped with one-time PROM (Programmable ROM). Programs can be written using commercially available PROM writers. However, programs can be written only once and the written program cannot be erased.

- Applicable types: M37700E2-XXXFP/M37700E2AXXXFP
  M37700E4-XXXFP/M37700E4AXXXFP

④ EPROM type microcomputer
This type of microcomputer is equipped with EPROM and programs can be written using commercially available PROM writers. Furthermore, programs can be re-written by exposing the window on top of the package to an ultraviolet rays.

- Applicable types: M37700E2FS/M37700E2AFS
  M37700E4FS/M37700E4AFS

1.2.3 Meaning of type name

The names of the Mitsubishi Single-chip microcomputers indicate the differences in functions and characteristics. The name of the M37700 family has the following meaning:

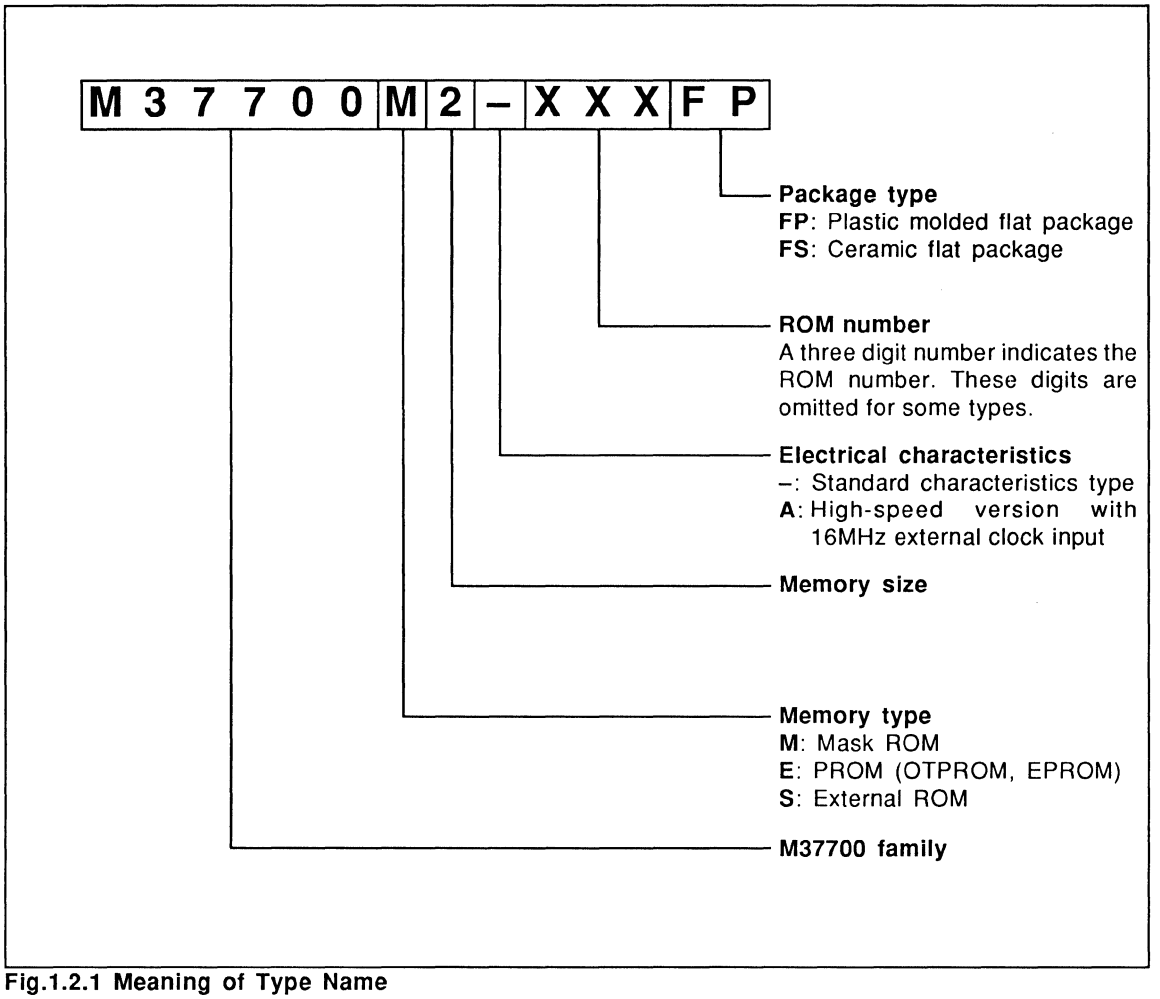


Fig.1.2.1 Meaning of Type Name

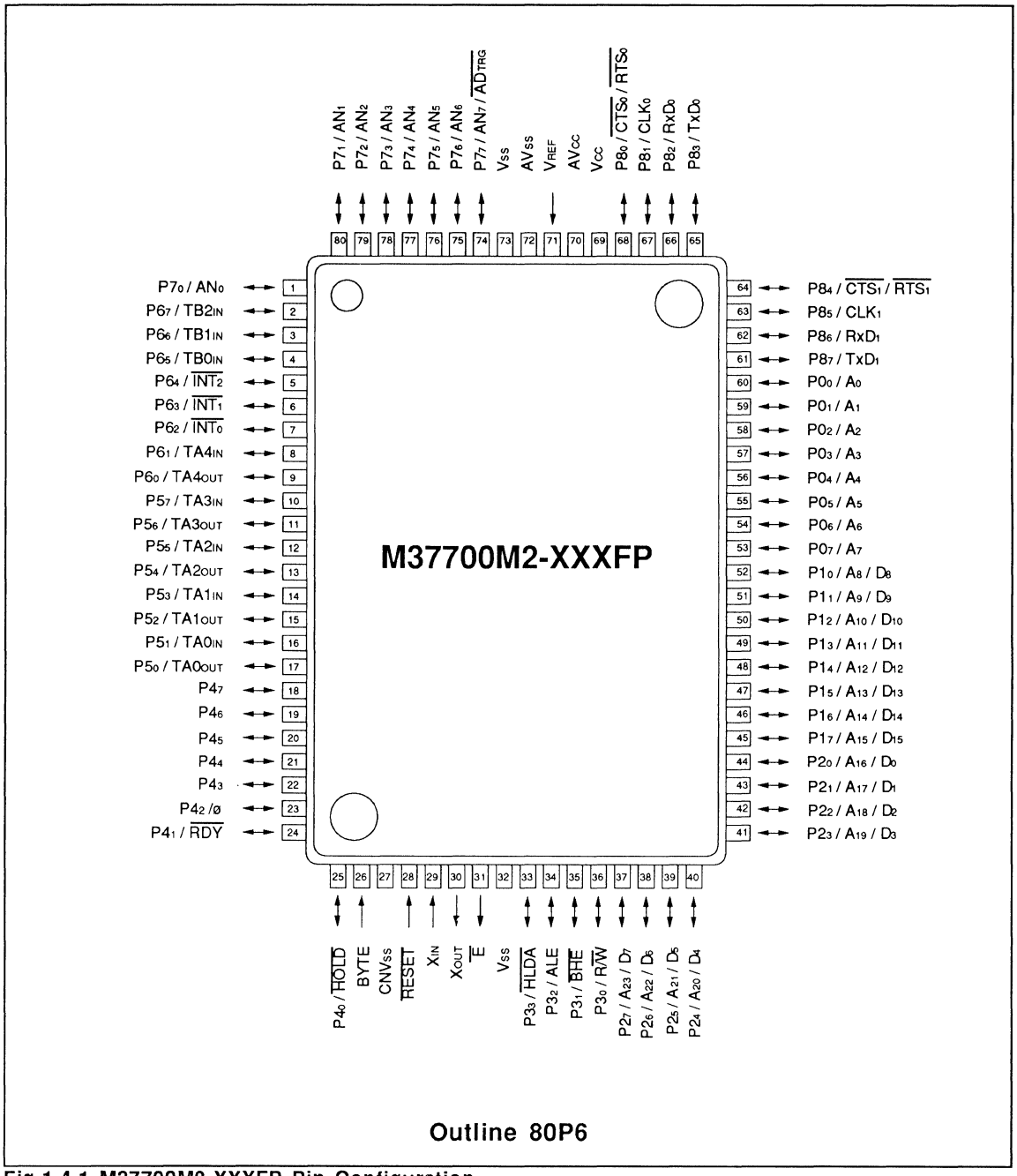
1.3 Performance Overview

Table 1.3.1 shows the performance overview of the M37700M2-XXXFP/ M37700M2AXXXFP. Refer to "Appendix 3. M37700 Family Performance Overview" for the performance of other types.

Table 1.3.1 M37700M2-XXXFP/M37700M2AXXXFP Performance Overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time	M37700M2-XXXFP	500ns (shortest instruction at 8MHz frequency)
	M37700M2AXXXFP	250ns (shortest instruction at 16MHz frequency)
Clock frequency	M37700M2-XXXFP	8MHz (maximum)
	M37700M2AXXXFP	16MHz (maximum)
Memory size	ROM	16384 bytes
	RAM	512 bytes
Input/Output ports	Ports P0~P2, P4~P8	8 bits x 8
	Port P3	4 bits x 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16 bits x 5
	TB0, TB1, TB2	16 bits x 3
Serial I/O		(UART or clock synchronous serial I/O) x 2
A-D converter		8 bits x 1 (8 channels)
Watchdog timer		12 bits x 1
Interrupts		3 external, 16 internal (priority levels 0 to 7 can be set for each interrupt with software)
Clock generating circuit		Built-in (externally connected to a ceramic resonator or quartz crystal oscillator)
Supply voltage		5V±10%
Power dissipation		30mW (at external 8MHz frequency)
Input/Output characteristics	Input/Output voltage	5V
	Output current	5mA
Memory expansion		Maximum 16M bytes
Operating temperature range		–10 to 70°C
Device structure		CMOS high-performance silicon gate process
Package		80-pin plastic molded QFP

1.4 Pin Configuration



Outline 80P6

Fig.1.4.1 M37700M2-XXXFP Pin Configuration

1.5 Pin Description

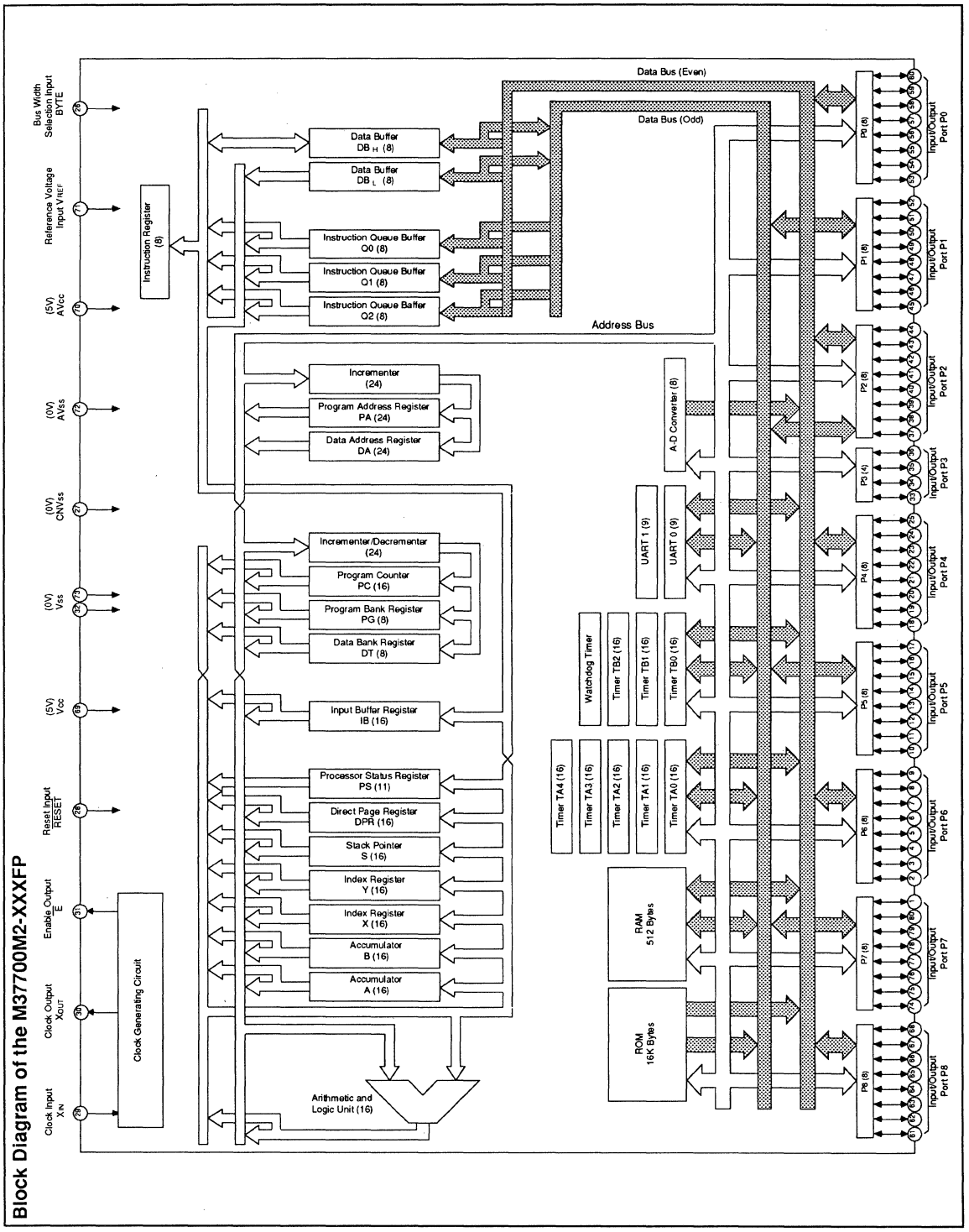
Table 1.5.1 Pin Description (a)

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5V±10% to V_{CC} and 0V to V_{SS}.
CNVss	CNVss input	Input	This pin controls the processor mode. Connect to Vss for single-chip mode. It must be connected to V_{CC} for external ROM types.
$\overline{RESET}$	Reset input	Input	The microcomputer is reset when this pin is set to "L" level.
X_{IN}	Clock input	Input	These are the I/O pins of the internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	When in memory expansion mode or microprocessor mode, this pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when the signal level is "L" and 8 bits when the signal level is "H".
AV_{CC}, AV_{SS}	Analog supply input		Power supply for the A-D converter. Externally connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS}.
V_{REF}	Reference voltage input	Input	This is a reference voltage input pin for the A-D converter.
$P0_0$~$P0_7$	I/O port P0	I/O	This port is a CMOS I/O port. An I/O direction register is available so that each pin can be programmed for input or output. Address (A_0~A_7) is output in memory expansion mode or microprocessor mode.
$P1_0$~$P1_7$	I/O port P1	I/O	This port is an 8-bit I/O port with the same function as P0. When the BYTE pin is set to "H" in memory expansion mode or microprocessor mode, address (A_8~A_{15}) is output. When the BYTE pin is set to "L", an address (A_8~A_{15}) is output when $\overline{E}$ pin level is "H" and high-order data (D_8~D_{15}) is input or output when $\overline{E}$ pin level is "L".
$P2_0$~$P2_7$	I/O port P2	I/O	This port is an 8-bit I/O port with the same function as P0. In memory expansion mode or microprocessor mode, an address (A_{16}~A_{23}) is output when $\overline{E}$ pin level is "H" and low-order data (D_0~D_7) is input or output when $\overline{E}$ pin output is "L".
$P3_0$~$P3_3$	I/O port P3	I/O	This port is a 4-bit I/O port with the same function as P0. In memory expansion mode or microprocessor mode, $P3_0$~$P3_3$ become R/$\overline{W}$, $\overline{BHE}$, ALE, and $\overline{HLDA}$ signals are output respectively.
$P4_0$~$P4_7$	I/O port P4	I/O	This port is an 8-bit I/O port with the same function as P0. $P4_2$ can be programmed as a ø output pin. In memory expansion mode or microprocessor mode, $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pin respectively.

Table 1.5.1 Pin Description (b)

Pin	Name	Input/Output	Functions
$P5_0$~$P5_7$	I/O port P5	I/O	This port is an 8-bit I/O port with the same function as P0. These pins can also be programmed as I/O pins for timers A0~A3.
$P6_0$~$P6_7$	I/O port P6	I/O	This port is an 8-bit I/O port with the same function as P0. These pins can be programmed as I/O pins for timer A4, external interrupt input pins for $\overline{INT_0}$~$\overline{INT_2}$, and input pins for timers B0~B2.
$P7_0$~$P7_7$	I/O port P7	I/O	This port is an 8-bit I/O port with the same function as P0. These pins can be programmed as analog input pins AN_0~AN_7. $P7_7$ also has an A-D conversion trigger input function.
$P8_0$~$P8_7$	I/O port P8	I/O	This port is an 8-bit I/O port with the same function as P0. These pins can be programmed as $\overline{CTS}/\overline{RTS}$, CLK, RxD, TxD pins for UART0 and UART1.

1.6 Block Diagram



CHAPTER 2
FUNCTIONAL DESCRIPTION

CHAPTER 2.FUNCTIONAL DESCRIPTION

2.1 Central Processing Unit (CPU)

The MELPS 7700 Series CPU has ten registers as shown in Figure 2.1.1. Each of these registers is described below.

2.1.1 Accumulator (Acc)

Accumulators A and B are available and each can be used as 8-bit or 16-bit register as necessary.

(1) Accumulator A (A)

Accumulator A is the main register of the microcomputer. Data operations such as calculations, data transfer, and input/output are executed mainly through accumulator A. It consists of 16 bits and the lower 8 bits can be used separately. The data length flag (m) determines whether the register is used as a 16-bit register or as an 8-bit register. It is used as a 16-bit register when flag m is "0" and as an 8-bit register when flag m is "1". Flag m is a part of the processor status register (PS) which is described later.

(2) Accumulator B (B)

Accumulator B has the same functions as accumulator A. The series MELPS 7700 instructions can use accumulator B instead of accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A. Accumulator B is also controlled by the data length flag m.

2.1.2 Index register X

Index register X consists of 16 bits and the lower 8 bits can be used separately. The index register length flag (x) determines whether the register is used as a 16-bit register or as an 8-bit register. It is used as a 16-bit register when flag x is "0" and as an 8-bit register when flag x is "1". Flag x is a part of the processor status register (PS) which is described later.

In index addressing mode, register X is used as the index register and the contents of this address is added to obtain the real address.

Also, when executing a block transfer instruction **MVP** or **MVN**, the contents of the index register X indicates the low-order 16 bits of the source data address. The third byte of the **MVP** and **MVN** is the high-order 8 bits of the source data address.

2.1.3 Index register Y

Index register Y is a 16-bit register with the same function as index register X. As with index register X, the index register length flag (x) determines whether this register is used as a 16-bit register or as an 8-bit register. Also, when executing a block transfer instruction **MVP** or **MVN**, the content of index register Y indicates the low order 16 bits of the destination data address. The second byte of the **MVP** and **MVN** is the high-order 8 bits of the destination data address.

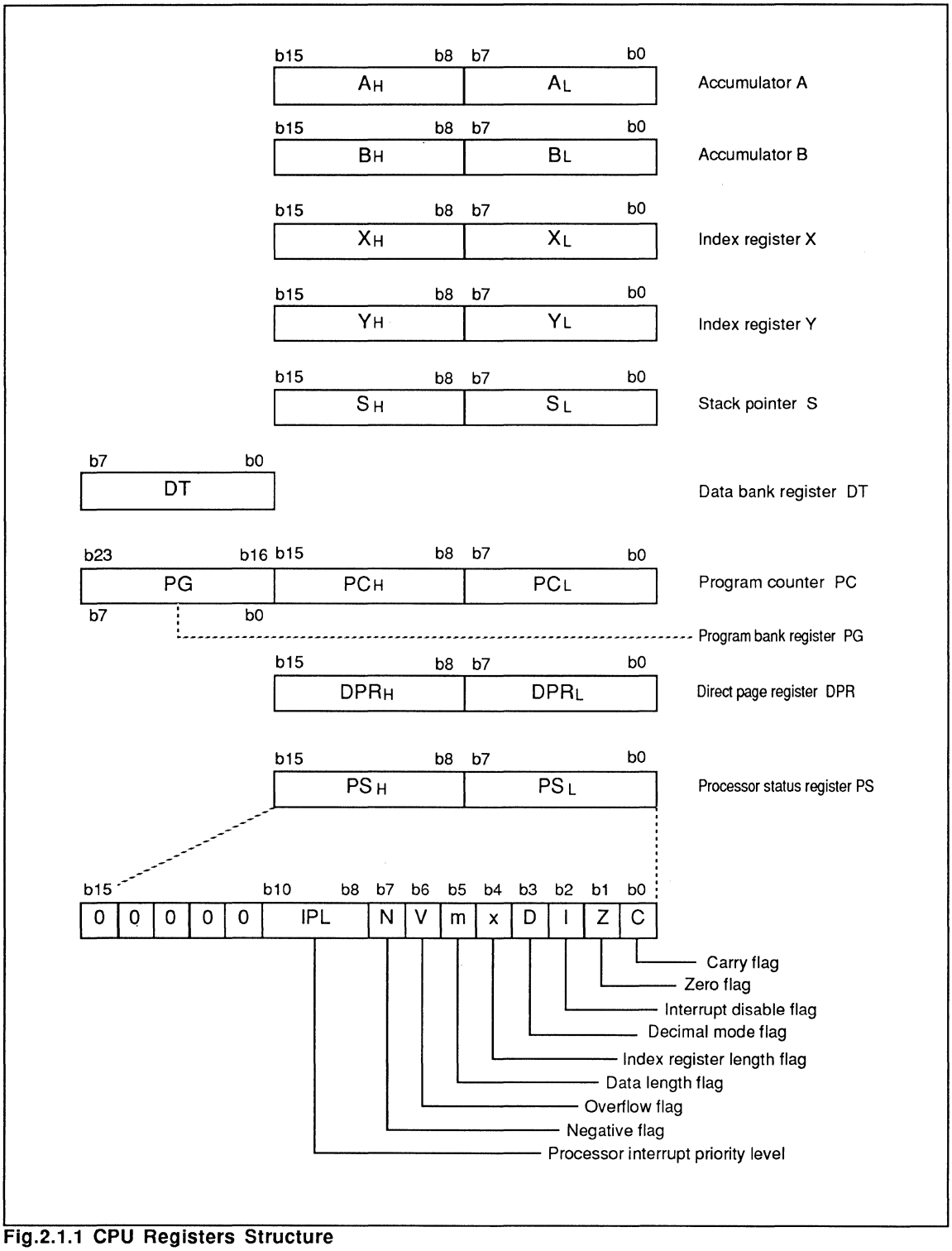


Fig.2.1.1 CPU Registers Structure

2.1.4 Stack pointer (S)

Stack pointer S is a 16-bit register. It is used during a subroutine call or interrupt. It is also used during addressing modes using the stack. The contents of the stack pointer S indicates the address (stack area) for saving registers during subroutine calls and interrupts. Normally, the stack area is reserved in internal RAM.

When an interrupt occurs, the contents of the program bank register PG is saved at the address indicated by the content of the stack pointer S and the content of the stack pointer is decremented by 1. Then the contents of the program counter PC and the processor status register PS are saved with the high-order bytes followed by the low-order bytes (PC_H, PC_L, PS_H, PS_L). The contents of the stack pointer S after an interrupt is equal to the content before the interrupt –5. When returning to the original routine after processing the interrupt, the registers saved in the stack area are restored to the original registers in the reverse sequence and the content of the stack pointer is returned to the status before the interrupt. The same operation is performed during a subroutine call, but the content of the processor status register PS is not saved (the content of the program bank register PG may not be saved either depending on the addressing mode).

The user is responsible for saving registers other than those described above during interrupts or subroutine calls. In addition, the stack pointer S must be initialized at the beginning of the program because its content is unpredictable after a reset. Normally, the stack pointer is initialized with the highest address of the internal RAM. The contents of the stack area changes when subroutines are nested or when multiple interrupts occur. Therefore, make sure necessary data in the internal RAM are not destroyed when nesting subroutines.

2.1.5 Program counter (PC)

Program counter PC is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed.

2.1.6 Program bank register (PG)

Program bank register PG is an 8-bit register that indicates the high-order 8 bits (bank) of the next program memory address to be executed. When a carry occurs after incrementing the content of the program counter, the content of the program bank register PG is incremented by 1. Also, when a carry or borrow occurs after adding or subtracting the content of the program counter PC, the content of the program bank register PG is incremented or decremented by 1 so that programs can be written without considering bank boundaries.

In single-chip mode, do not store values other than "00_{16}" because only address between 0000_{16} and $FFFF_{16}$ can be accessed.

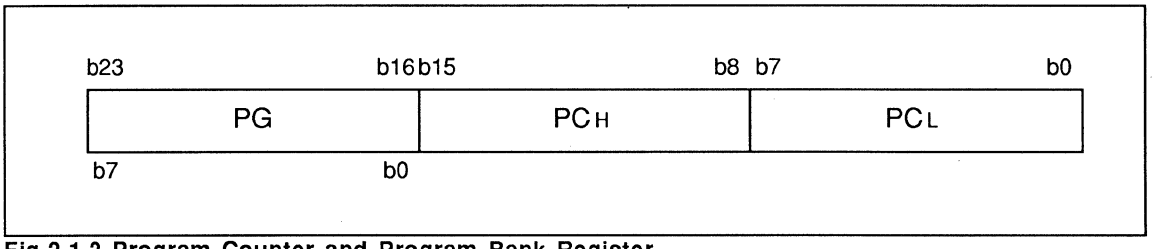


Fig.2.1.2 Program Counter and Program Bank Register

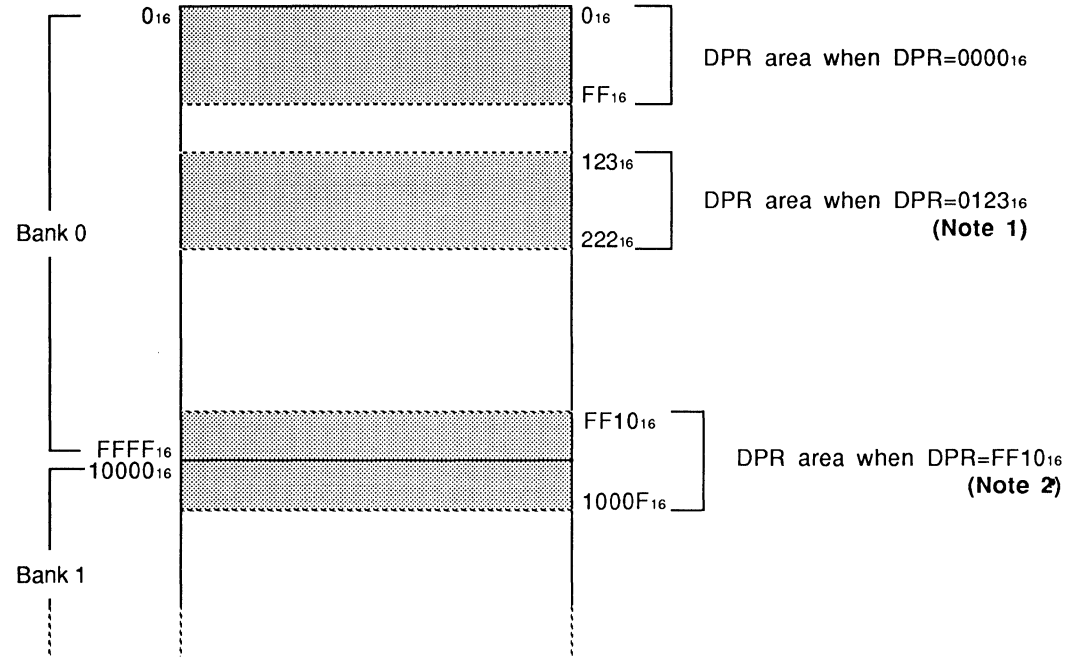
2.1.7 Data bank register (DT)

Data bank register DT is an 8-bit register. With some addressing modes, the content of this register is used as the high-order 8 bits of a 24-bit address. In single-chip mode, do not store values other than "00_{16}" because only address between 0000_{16} and $FFFF_{16}$ can be accessed.

2.1.8 Direct page register (DPR)

Direct page register DPR is a 16-bit register. The content of this register indicates whether the direct page area is allocated in bank 0 or spans across bank 0 and 1. This area can be accessed with two bytes by using the direct page addressing mode.

The content of the DPR is the base address (lowermost address) of the direct page area which extends 256 bytes above this address. The DPR can contain a value from 0000_{16} to $FFFF_{16}$. If it contains a value equal to or greater than "$FF01_{16}$", the direct page area spans across banks 0 and 1. If the low-order 8 bits of the DPR is "00_{16}", the number of cycles required to generate an address is minimized. Therefore, the low-order 8-bits of the DPR should normally be set to "00_{16}".



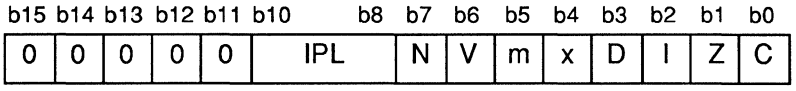
Note 1 : The number of execution cycles is incremented by 1 when the low-order 8 bits of the DPR are not "00_{16}".

Note 2 : The direct page spans across banks 0 and 1 when the DPR is "$FF01_{16}$" or greater.

Fig.2.1.3 Setting Direct Page with Direct Page Register (DPR)

2.1.9 Processor status register (PS)

Processor status register is an 11-bit register. It consists of flags to indicate the result of operation and CPU interrupt levels. The flags C, Z, V, and N are tested by branch instructions.
The details of the processor status register bits are described below.



Note : Bits 11 to 15 will always be "0" when the contents of the processor status register are read.

Fig.1.2.4 Processor Status Register

(1)Carry flag (C)

The carry flag is assigned to bit 0 of the processor status register. It contains the carry or borrow bit from the arithmetic and logic unit (ALU) after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set with the **SEC** or **SEP** instruction and cleared with the **CLC** or **CLP** instruction.

(2)Zero flag (Z)

The zero flag is assigned to bit 1 of the processor status register. It is set if the result of an arithmetic operation or data transfer is zero and cleared if otherwise. This flag can be set and cleared directly with the **SEP** and **CLP** instructions.
Note : The content of this flag has no meaning during decimal mode addition (**ADC** instruction).

(3)Interrupt disable flag (I)

The interrupt disable flag is assigned to bit 2 of the processor status register. It disables all maskable interrupts (interrupts other than watchdog timer, **BRK** instruction, and zero divide). Interrupts are disabled when this flag is "1". When there is an interrupt, it is set automatically to prevent multiple interrupts. This flag can be set with the **SEI** or **SEP** instruction and cleared with the **CLI** or **CLP** instruction. This flag is set during reset.

(4)Decimal mode flag (D)

The decimal mode flag is assigned to bit 3 of the processor status register. It determines whether addition and subtraction are performed in binary or decimal. Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as two or four digit decimal (determined by the data length flag m). Decimal adjust is performed automatically (Decimal operation is possible only with the **ADC** and **SBC** instructions.) This flag can be set and cleared with the **SEP** and **CLP** instructions. This flag is cleared during reset.

(5) Index register length flag (x)

The index register length flag is assigned to bit 4 of the processor status register. It determines whether the index register X or index register Y is used as a 16-bit register or an 8-bit register. The register is used as a 16-bit register when flag x is "0" and as an 8-bit register when it is "1". This flag can be set and cleared with the **SEP** and **CLP** instructions. This flag is cleared during reset.

(6) Data length flag (m)

The data length flag is assigned to bit 5 of the program status register. It determines whether to treat data as 16-bit or as 8-bit. A data is treated as 16-bit when flag m is "0" and as 8-bit when it is "1". This flag can be set with the **SEM** or **SEP** instruction and cleared with the **CLM** or **CLP** instruction. This flag is cleared during reset.

(7)Overflow flag (V)

The overflow flag is assigned to bit 6 of the processor status register. It is used when adding or subtracting a word as signed binary. When the data length flag m is "0", the overflow flag is set when the result of addition or subtraction is outside the range between –32768 and +32767. When the data length flag m is "1", the overflow flag is set when the result of addition or subtraction is outside the range between –128 and +127. It is cleared in all other cases. The overflow flag can also be set and cleared directly with the **SEP**, **CLV**, and **CLP** instructions.

Note : This flag no meaning in decimal mode.

(8)Negative flag (N)

The negative flag is assigned to bit 7 of the processor status register. It is set when the result of arithmetic operation or data transfer is negative (Data bit 15 is 1 when data length flag m is "0" or data bit 7 is 1, when data length flag m is "1".). It is cleared in all other cases. It can also be set and cleared with the **SEP** and **CLP** instructions.

Note : This flag has no meaning in decimal mode.

(9)Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) is assigned to bits 8, 9, and 10 of the processor status register. These three bits determine the priority level of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority level of the requested interrupt (set with the interrupt control register) is higher than the processor interrupt priority. When an interrupt occurs, the IPL is saved in the stack and the processor interrupt priority is replaced by the interrupt priority of the accepted interrupt. This simplifies control of multiple interrupts.

There are no instructions to directly set or clear the IPL. It can be changed by placing the new IPL on the stack and updating the processor status register with a **PUL** or **PLP** instruction.

2.2 Internal Bus Interface

2.2.1 Internal bus interface overview

A bus interface unit (BIU) is provided between the CPU and the internal bus. Transfer of data between the CPU and memory or I/O device is always performed through the BIU. When the CPU reads data from memory or an I/O device, it sends the address to be read to the BIU. The BIU reads the data from the specified address and the CPU receives the data from the BIU. Similarly, the CPU sends the address to be written to the BIU when writing data. Thus the BIU controls the transfer of data between the CPU and bus.
Figure 2.2.1 shows a block diagram of the bus interface unit.

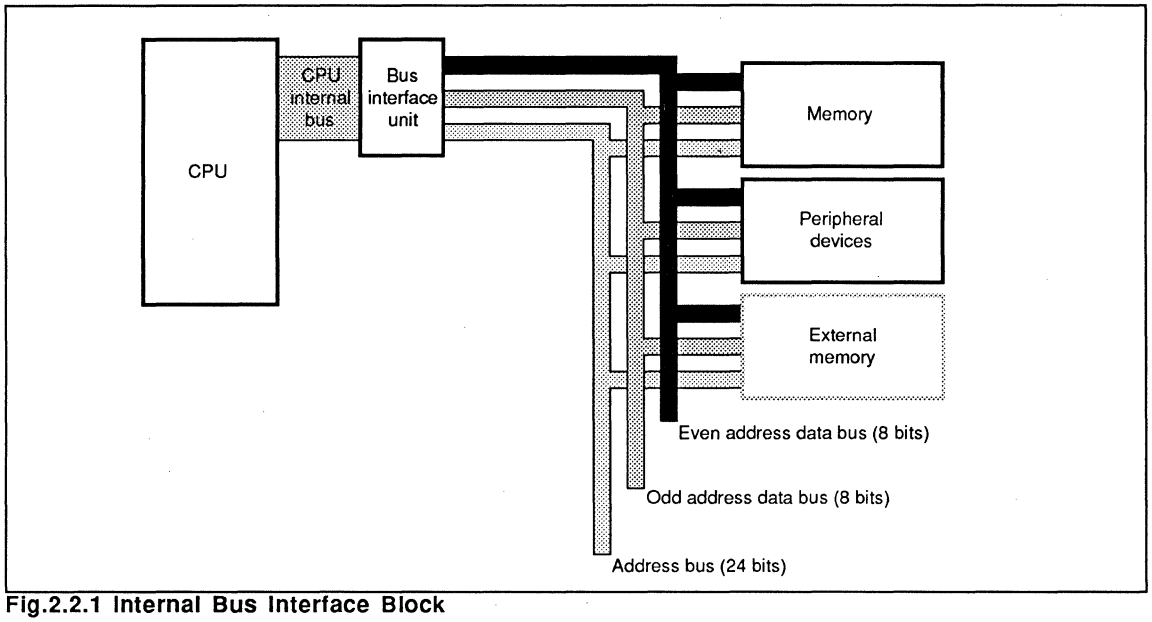


Fig.2.2.1 Internal Bus Interface Block

2.2.2 Bus Interface unit functions

The M37700 uses the clock ø (=f(XIN)/2) as the clock. The CPU also uses clock ø as the clock. However, since the CPU clock may be extended due to CPU wait under certain conditions, it is referred to as øCPU to distinguish it from clock ø.
The M37700 internal bus (address bus and data bus) operate at timing $\overline{E}$ which is slower than clock ø The operating clock of the CPU is different from the bus cycle because timing $\overline{E}$ is normally f(XIN)/4. Therefore, a BIU is provided between the CPU and bus to synchronize the transfer of data to and from memory and I/O device. The BIU enables the CPU to transfer data to and from memory through the bus without decreasing the instruction execution speed.
The BIU consists of four registers as shown in Figure 2.2.2. Table 2.2.1 summarizes the functions of each register and buffer.

Table 2.2.1 Functions of BIU Registers and Buffers

Name	Function
Program address register	Indicates the address of the program.
Instruction queue buffer	A three bytes buffer for temporarily holding instruction prefetched from memory.
Data address register	Indicates the address to be read from or to be written to memory or I/O.
Data buffer	A two bytes buffer for temporarily holding data read from memory or I/O device by the BIU or data written to memory or I/O device by the CPU.

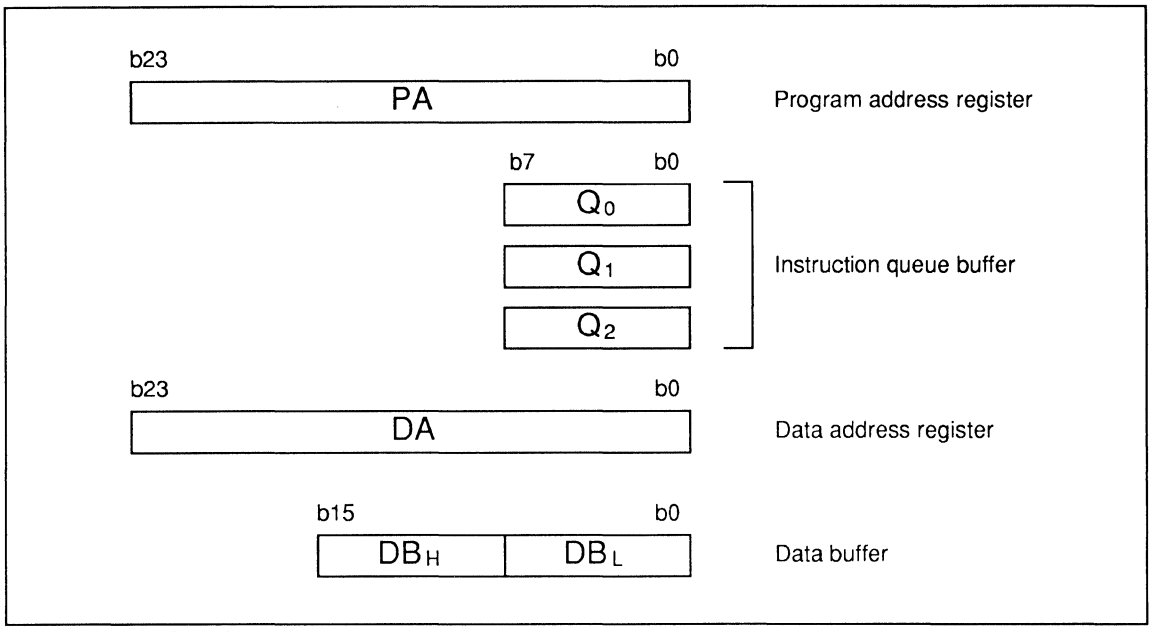


Fig.2.2.2 Bus Interface Unit Registers

The BIU performs the following operations.

1. **Prefetches an instruction code from the program memory (area where the program is stored) and stores it in the instruction queue buffer.**
   Normally, a program is executed sequentially in ascending order of addresses. Therefore, if the next instruction code is prefetched in the instruction queue buffer, the CPU can execute instructions simply by obtaining the instruction code from the instruction queue buffer. This will eliminate the time needed by the CPU to access the memory.
   When the CPU is not using the bus (for example when performing register to register operation), the BIU reads an instruction code from the program memory (area where the program is stored) and stores it in the instruction queue buffer. Data up to three bytes can be prefetched because the instruction queue buffer is three bytes long. Refer to "Section 2.2.4" for more information concerning instruction code prefetch.

2. **Reads data at the specified address into the BIU when the CPU requests data in memory and transfers it to the CPU.**
   When executing instructions that processes data in memory or I/O device, the CPU must access the address assigned to the memory or I/O device and read the data. Because the operating clock of the CPU and bus are different, the CPU reads the data through the data buffer of the BIU.

3. **Writes the data obtained from the CPU to the specified address in memory.**
   When writing data to a specific address, the CPU sends the address and data to the BIU. And after that, the CPU continue to execute the next instruction extracting from the instruction queue buffer, because actual writing to memory or I/O device is performed by the BIU.

4. **Controls read of word data from odd number address and outputs the control signals required to access external memory in byte unit.**
   The transfer of data between the CPU and BIU is always performed through a 24-bit address bus and 16-bit data bus. This is also true between the BIU and internal memory or I/O device. The wait bit and BYTE pin (external bus width selection input pin) determine the data width only when an external memory is accessed.

2.2.3 Bus Interface unit operations

Figure 2.2.3 shows the operating waveforms of the bus interface unit in memory expansion mode or microprocessor mode. The M37700 BIU always operates at one of the waveforms shown in Figure 2.2.3. The meaning of signals ALE and $\overline{E}$ in Figure 2.2.3 are as follows:

- ALE (Address Latch Enable)
  Signal used to latch only address signals from multiplexed signals containing data and address.
- $\overline{E}$
  Signal set to "L" level when the bus interface unit reads instruction code or data from memory or when it writes data to memory. Table 2.2.2 shows the bus status according to $\overline{E}$ and R/$\overline{W}$ signals.

Table 2.2.2 Bus Status According to $\overline{E}$ and R/$\overline{W}$

$\overline{E}$	R/$\overline{W}$	Bus Status
H	H	Not used
H	L	Not used
L	H	Read
L	L	Write

(1)Basic operation

Waveform (a) is the bus interface operating waveform under the following conditions:

- When a one byte internal/external memory is accessed.
- When two bytes in internal memory are accessed together (starting on an even address).
- When two bytes in external memory are accessed together (starting on an even address when the BYTE pin is at "L" level).

Waveform (b) is the bus interface operating waveform when accessing in byte unit under the following conditions:

- When two bytes in internal/external memory are accessed together (starting on an odd address).
- When two bytes in external memory are accessed together with the BYTE pin at "H" level.

As obtaining the instruction code from memory into the instruction queue buffer, waveform (a) is only used.

Waveforms (a) and (b) are the basic operating waveforms of the BIU. Waveform (a) or (b) is always used when accessing the internal memory. However, signals other than $\overline{E}$ cannot be observed in single-chip mode because the port P3 is used as a programmable I/O port.

(2)Effect of the wait bit

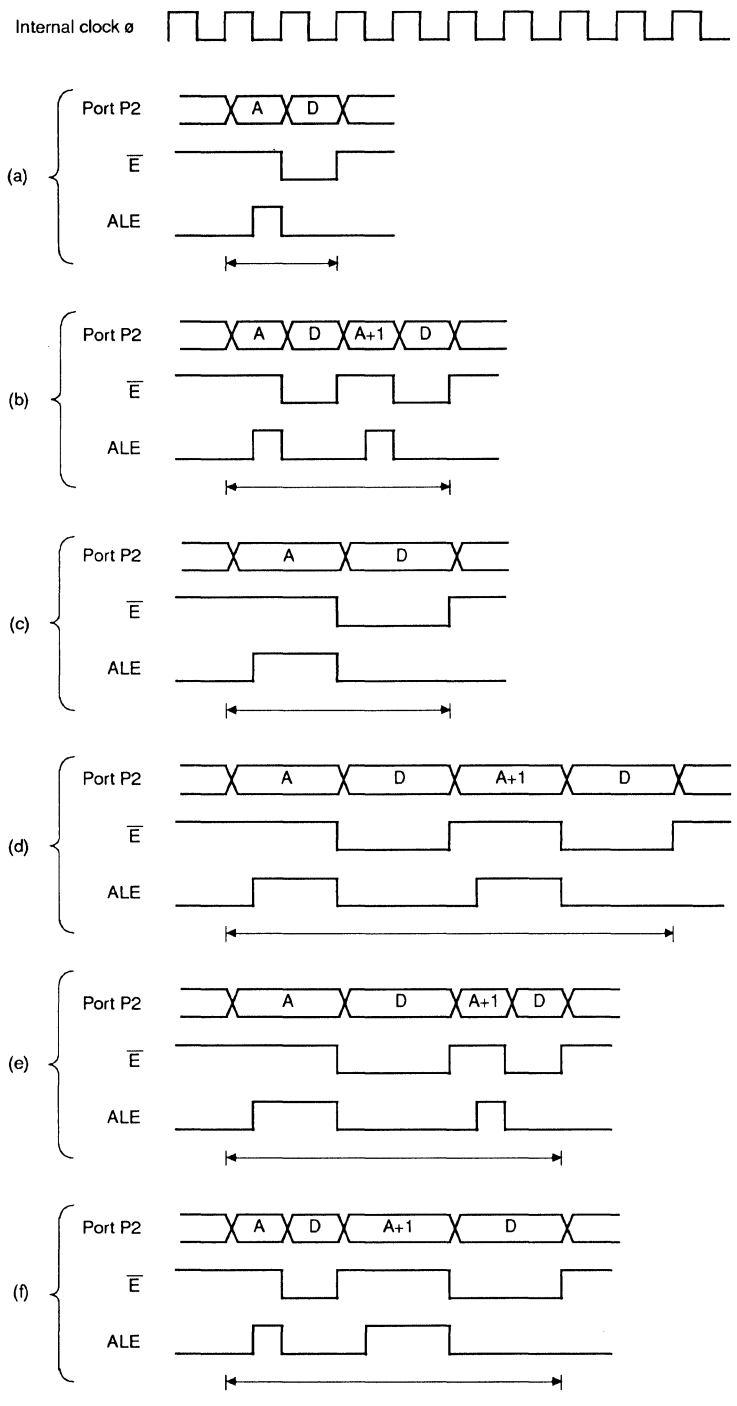
When accessing the external memory area, the BIU operating waveform changes according to the wait bit.

With the M37700, the external memory access time can be doubled (signal $\overline{E}$ is doubled) by clearing the wait bit (bit 2) in the processor mode register ($005E_{16}$). This enables external expansion of slow memories and peripheral LSIs.

Note : Internal memory access is not affected by the wait bit.

Figure 2.2.3 (c) to (f) show the effect of the wait bit on waveforms (a) and (b). Waveform (c) is the waveform when an external memory area is accessed under the conditions for waveform (a) with the wait bit cleared.

Waveforms (d) to (f) are the waveforms when an external memory area is accessed under the conditions for waveform (b) with the wait bit cleared. The entire waveform is affected by the wait bit for waveform (d) and the first half or the last half is affected respectively for waveforms (e) and (f).



- A: Address
- D: Data
- The waveform is for memory expansion mode or microprocessor mode.

Fig.2.2.3 Bus Interface Device Operating Waveform

2.2.4 Data read/write operations

(1)Instruction code read

The CPU reads instructions codes from the instruction queue buffer of the BIU and executes them. The CPU notifies the BIU that an instruction code is needed during the instruction code fetch cycle. At this point, the operation depends on whether the instruction queue buffer contains an instruction code or not. If there is an instruction code in the instruction queue buffer, it is passed to the CPU. If there is no instruction code in the instruction queue buffer, or if the amount of data in the instruction queue buffer is less than the necessary instruction code, the BIU halts the CPU until a sufficient amount of instruction codes is stored in the instruction queue buffer.

Even when there is no request for instruction code from the CPU, if the instruction queue buffer is empty or if there is only one instruction code and the bus is available at the next cycle (the CPU does not use the bus at the next cycle), the BIU reads instruction codes from memory and stores them in the instruction queue buffer (instruction prefetch). During instruction prefetch, if the first address accessed when reading an instruction code from memory is even, then the data at the next odd number address is also read and stored in the instruction queue buffer. If the first accessed address is odd, only one byte is read and stored in the instruction queue buffer. However, if the instruction code is read from external memory with the BYTE pin at "H" (external bus width 8-bits) in memory expansion or microprocessor mode, only one byte is read regardless of the accessed address.

Instruction code read is performed with operation (a) or (c) shown in Figure 2.2.3. When a branch or a jump or subroutine call instruction or an interrupt is executed, the content of the instruction queue buffer is cleared and a new instruction code is read from the new address.

(2)Data read/write

The CPU reads and writes data from/to the BIU data buffer. The CPU issues a request to BIU when it attempts to read or write data. At this point, if the BIU is using the bus or if there is a higher priority request, the CPU is made to wait until the BIU becomes ready. When the bus is available for data read or write, the BIU operates at one of the waveforms (a) to (f) shown in Figure 2.2.3.

- Data Read

  When the CPU requests data from the BIU, it waits until the data is became complete data in the data buffer. The BIU sends the address received from the CPU on the address bus, reads the content of memory when $\overline{E}$ is "L", and stores it in the data buffer.

- Data Write

  The CPU sends address data (address at which the data is written) and data to BIU.

  The address data is written in the BIU data address register and the data is written in the data buffer. The actual writing in memory is performed by BIU and the CPU can proceed to the next step without waiting for the BIU to complete writing data in memory. The BIU sends the address data it received from the CPU to the address bus, sends the contents of the data buffer to the data bus, and writes it to memory when $\overline{E}$ is "L".

2.3 Addressable memory space

The M37700 allocates all ROM, RAM, I/O, and various control registers in the same memory space. Therefore, data transfer and operation can be performed with the same instruction without distinguishing memory and I/O area.

The M37700 program counter (PC) consists of 16 bits. It is used together with an 8-bit program bank register (PG) to directly address a 16M-byte address space from 0_{16} to $FFFFFF_{16}$.

2.3.1 Banks

The M37700 address space is divided into 64K byte blocks called banks. The Series MELPS 7700 can access 256 banks from bank 0 to bank 255 (FF_{16}) in memory expansion or microprocessor mode.

The high order 8 bits of the 24-bit address indicate the bank and the content of the program bank register (PG) or the data bank register (DT) indicates the bank to be used.

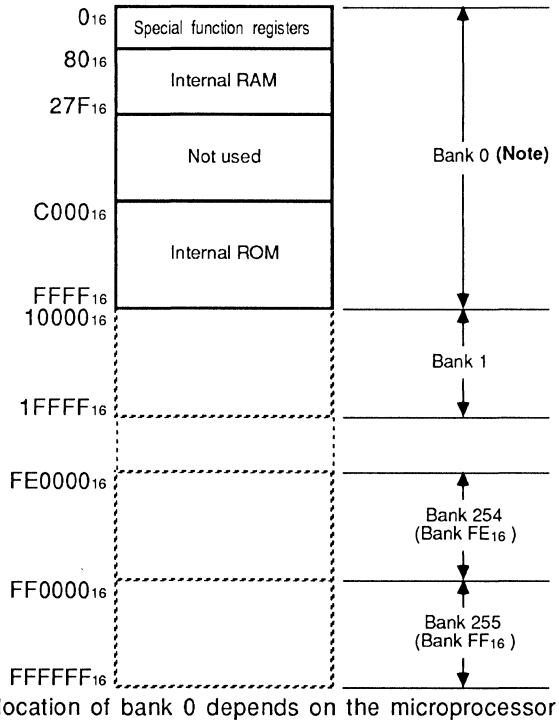
If the program counter overflows at a bank boundary, the content of the program bank register is incremented by 1. If a borrow occurs in the program counter register, the content of the program bank register is decremented by 1. Therefore, programs can be written without considering the bank boundaries. The banks can be accessed efficiently by using an addressing mode that uses the data bank register.

Bank 0 (addresses 0_{16} to $FFFF_{16}$) contains the internal ROM, internal RAM, and internal I/O control registers.

Note : In single-chip mode, only bank 0 can be accessed.

2.3.2 Direct Page

By using the direct page register (DPR), bank 0 or a 256-byte space spanning across bank 0 and bank 1 can be accessed with fewer instruction cycles in direct page addressing mode. This area is referred to as the direct page and is normally used for frequently accessed information (see "Section 2.1.8 Direct Page Register").



Note : The memory allocation of bank 0 depends on the microprocessor type. This diagram shows the allocation for M37700M2-XXXFP and M37700M2AXXXFP. Refer to the M37700 family memory map in Appendix 5 for other types. Also note that only bank 0 can be accessed in single-chip mode.

Fig.2.3.1 Addressable Memory Space

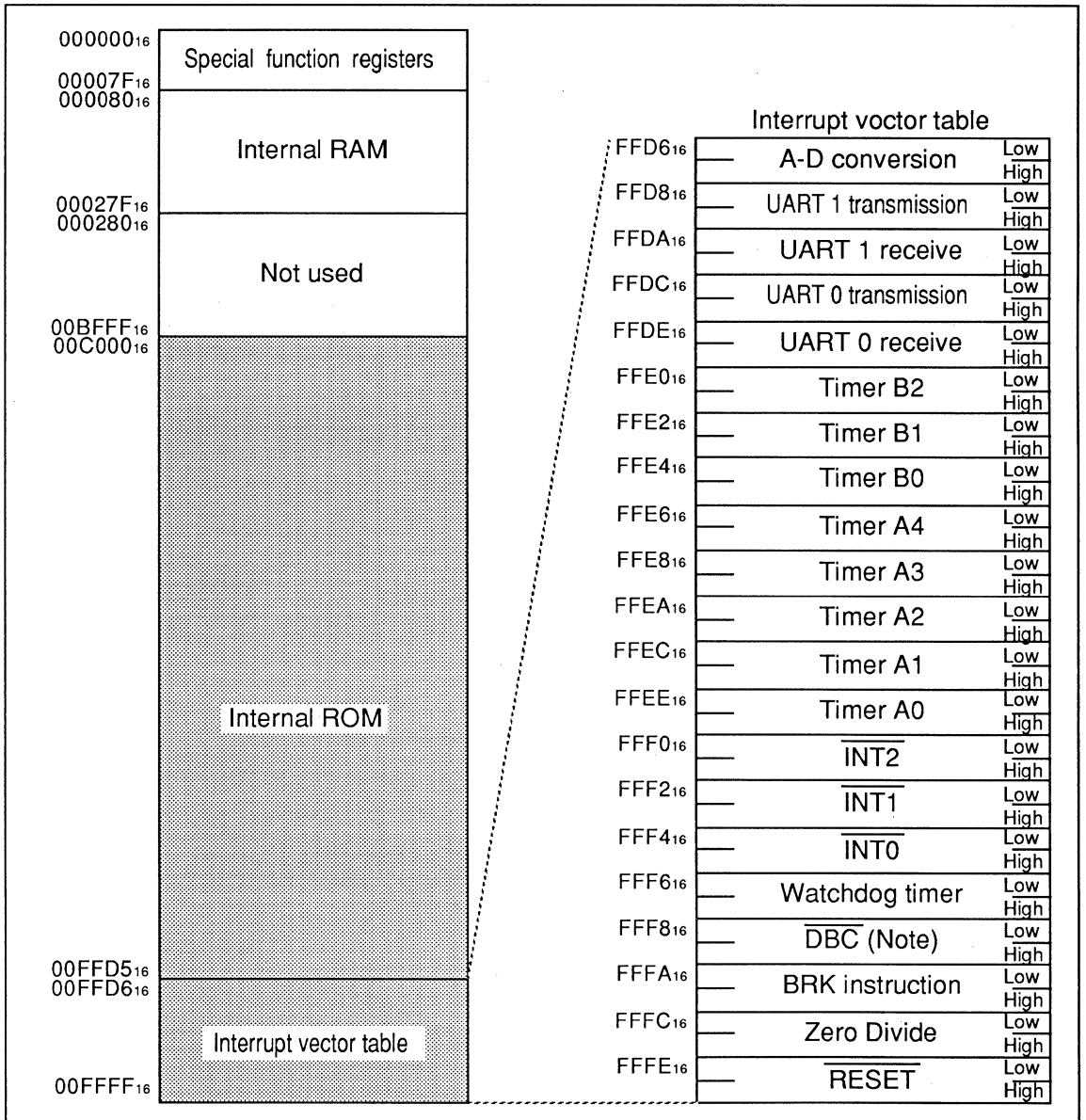
2.4 Memory Allocation

Figure 2.4.1 shows the memory map in single-chip mode. The allocated memory and I/O are described below.

2.4.1 Internal memory and peripheral device memory allocation

(1)SFR area

Addresses 0000_{16} to $007F_{16}$ of bank 0 are the SFR (Special Function Register) area. This area contains the control registers of internal peripheral devices, I/O ports, timers, and so on. Internal peripheral devices can be accessed through these registers. Figure 2.4.2 shows the memory map of the SFR area.



Note : $\overline{\text{DBC}}$ is an interrupt for the exclusive use of the debugger and is not available to use.

Fig.2.4.1 Memory Map (Single-chip mode)

Address (Hexadecimal notation)	
000000	
000001	
000002	Port P0 register
000003	Port P1 register
000004	Port P0 data direction register
000005	Port P1 data direction register
000006	Port P2 register
000007	Port P3 register
000008	Port P2 data direction register
000009	Port P3 data direction register
00000A	Port P4 register
00000B	Port P5 register
00000C	Port P4 data direction register
00000D	Port P5 data direction register
00000E	Port P6 register
00000F	Port P7 register
000010	Port P6 data direction register
000011	Port P7 data direction register
000012	Port P8 register
000013	
000014	Port P8 data direction register
000015	
000016	
000017	
000018	
000019	
00001A	
00001B	
00001C	
00001D	
00001E	A-D control register
00001F	
000020	A-D register 0
000021	
000022	A-D register 1
000023	
000024	A-D register 2
000025	
000026	A-D register 3
000027	
000028	A-D register 4
000029	
00002A	A-D register 5
00002B	
00002C	A-D register 6
00002D	
00002E	A-D register 7
00002F	
000030	UART 0 transmit/receive mode register
000031	UART 0 baud rate generator
000032 000033	UART 0 transmission buffer register
000034	UART 0 transmit/receive control register 0
000035	UART 0 transmit/receive control register 1
000036 000037	UART 0 receive buffer register
000038	UART 1 transmit/receive mode register
000039	UART 1 baud rate generator
00003A 00003B	UART 1 transmission buffer register
00003C	UART 1 transmit/receive control register 0
00003D	UART 1 transmit/receive control register 1
00003E 00003F	UART 1 receive buffer register

Address (Hexadecimal notation)	
000040	Count start flag
000041	
000042	One-shot start flag
000043	
000044	Up-down flag
000045	
000046 000047	Timer A0 register
000048 000049	Timer A1 register
00004A 00004B	Timer A2 register
00004C 00004D	Timer A3 register
00004E 00004F	Timer A4 register
000050 000051	Timer B0 register
000052 000053	Timer B1 register
000054 000055	Timer B2 register
000056	Timer A0 mode register
000057	Timer A1 mode register
000018	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	Timer B1 mode register
00005D	Timer B2 mode register
00005E	Processor mode register
00005F	
000060	Watchdog timer
000061	Watchdog timer frequency selection flag
000062	
000063	
000064	
000065	
000066	
000067	
000068	
000069	
00006A	
00006B	
00006C	
00006D	
00006E	
00006F	
000070	A-D conversion interrupt control register
000071	UART 0 transmission interrupt control register
000072	UART 0 receive interrupt control register
000073	UART 1 transmission interrupt control register
000074	UART 1 receive interrupt control register
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	Timer B1 interrupt control register
00007C	Timer B2 interrupt control register
00007D	$\overline{\text{INT}}$ 0 interrupt control register
00007E	$\overline{\text{INT}}$ 1 interrupt control register
00007F	$\overline{\text{INT}}$ 2 interrupt control register

Fig.2.4.2 SFR Area Memory Map

Each bit in the register can be either read only, write only, or read/write bit. An attempt to write to a read only bit is ignored and the result is unpredictable when a write only bit is read.
Some registers in the SFR area prohibits the use of instructions such as **CLB** or **SEB** that performs read-modify-write. See "Chapter 7. Usage Precautions" for more details.

(2)RAM

The M37700M2-XXXFP and M37700M2AXXXFP have a 512-byte static RAM at addresses 0080_{16} to $027F_{16}$ (Note) of bank 0. In addition to storing data, the internal RAM area is used as stack area during subroutine calls and interrupts. Therefore, be careful of subroutine nesting levels and multiple interrupt levels so that important data is not destroyed.
Note : See "Appendix 4. M37700 Family Memory Map" for other types.

(3)ROM

The M37700M2-XXXFP and M37700M2AXXXFP have a 16K-byte mask ROM at addresses $C000_{16}$ to $FFFF_{16}$ (Note) of bank 0. Addresses $FFD6_{16}$ to $FFFF_{16}$ are allocated to the interrupt vector table containing branch destinations (address of interrupt handling routines) when a reset or interrupt occurs. This area must be allocated to ROM in microprocessor mode and external ROM version (S version) which prohibit internal ROM.
Note : See "Appendix 4. M37700 Family Memory Map" for other types.

2.4.2 Processor modes

The M37700 can operate in single-chip mode, memory expansion mode, and microprocessor mode. The functions of some pins, memory organization, and address space depend on the processor mode. The processor mode can be selected internally or externally as described below.

- Externally changing the processor mode
  The processor mode after a reset start can be selected with the input level to the CNV_{SS} pin during reset start. Table 2.4.1 shows the relationship between the processor mode and the input level to the CNV_{SS} pin.

Table 2.4.1 Relationship between the Processor Mode and CNV_{SS} Pin Input Level

CNV_{SS} Pin	Processor Mode
V_{SS} level (0V)	Starts in single-chip mode after a reset. One of the three modes can be selected by changing the processor mode bit.
V_{CC} level (5V)	Starts in microprocessor mode after a reset. The other mode must not be selected by changing the processor mode bit.

- Internally changing the processor mode
  After a reset start with the CNVss pin set to Vss level, the processor mode can be changed internally from a program by changing the processor mode bits (bits 1 and 0 at address $5E_{16}$) in the processor mode register. Figure 2.4.4 shows the bit configuration of the processor mode register. When changing the processor mode internally, the actual function of each pin changes when the bus cycle $\overline{E}$ used to write to the processor mode register returns to "H" level.

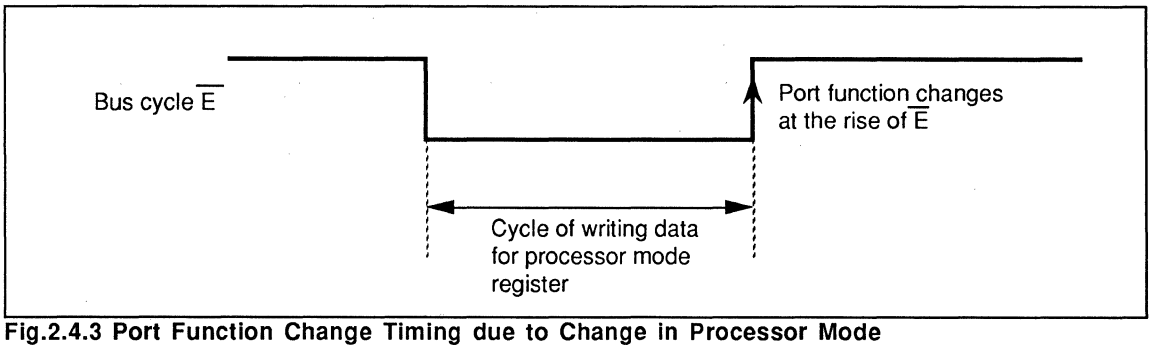
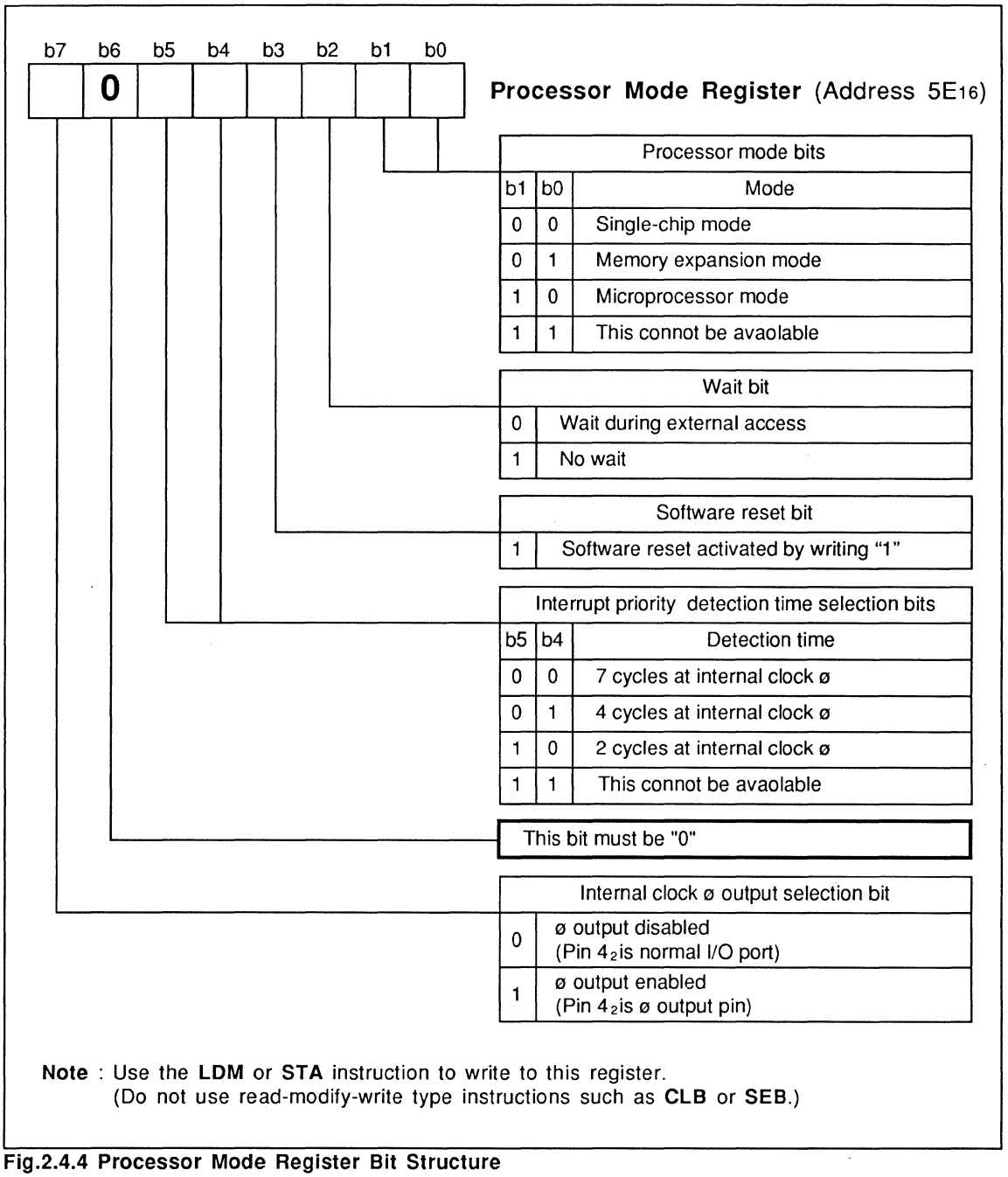


Fig.2.4.3 Port Function Change Timing due to Change in Processor Mode

(1)Single-chip mode

This mode is entered when starting after a reset with pin CNVss set to Vss level. In this mode, the address bus and data bus are not output externally and all ports function as programmable I/O pins (internal peripheral device I/O pins when internal peripheral devices are used). Also note that in single-chip mode, a non-zero value must not be stored in the data bank register and program bank register because only bank 0 is accessible. Furthermore, in this mode, the wait bit, which is described later, is ignored and internal memory and I/O are always accessed at no wait.



Note : Use the **LDM** or **STA** instruction to write to this register.
(Do not use read-modify-write type instructions such as **CLB** or **SEB**.)

Fig.2.4.4 Processor Mode Register Bit Structure

(2)Memory expansion mode

This mode is used when just using internal memory and I/O is not sufficient. In this mode, the memory and peripherals can be expanded to any area within a 16M-byte addressable memory space.

When the memory expansion mode is selected, ports P0 to P2 become the address bus and data bus and port P3 and part of P4 become the control signal I/O pins. In this case, the port register area associated with ports P0~P3 and part of P4 become unusable and lose their normal I/O pin functions, but other memory and peripherals can be used. See "Section 2.5 Input/Output Pins" for more details concerning the functions of ports P0~P4 when the memory expansion mode is selected.

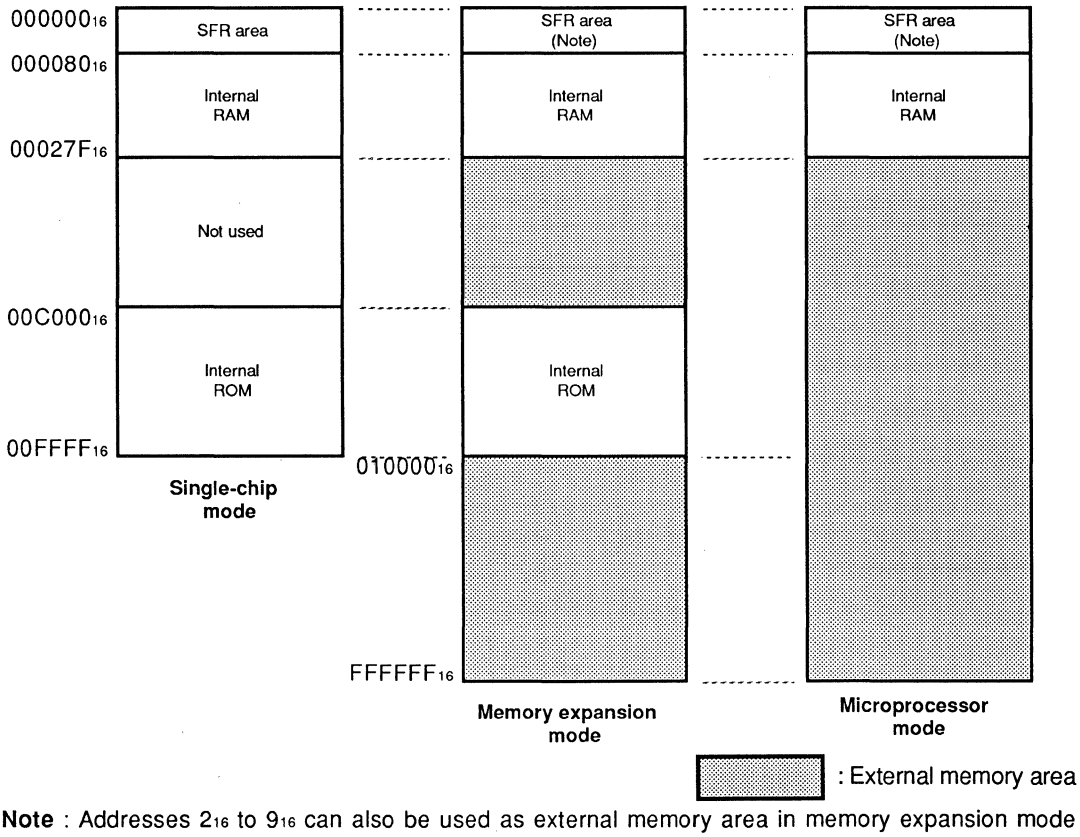
If an area overlapping the internal memory area is read when the external memory is extended, only the data in the internal memory is read into the CPU and the data in the external memory is not read into the CPU. However, if data is written in this area, it is written both in the internal memory and external memory.

Furthermore, the accessing of external memory in this mode is affected by the level of the BYTE pin and wait bit described in the next section.

(3)Microprocessor mode

The function of this mode is the same as the memory expansion mode except that access to internal ROM is disabled. This mode is suitable for small volume production or prototype models before full scale production because external ROM can be installed easily.

Figure 2.4.5 shows the memory allocation in each processor mode. See "Section 2.5 Input/Output Pins" for the change in port functions.



Note : Addresses 2_{16} to 9_{16} can also be used as external memory area in memory expansion mode and microprocessor mode.

Fig.2.4.5 Memory Map in Each Processor Mode (M37700M2-XXXFP)

2.4.3 External memory area bus control

The BYTE pin and the wait bit are provided to simplify access to external memory area in memory expansion mode and microprocessor mode. The BYTE pin and the wait bit are valid only when accessing external memory area and have no effect when accessing internal memory or internal peripherals. Therefore, the BYTE pin and the wait bit are ignored in single-chip mode.

(1)BYTE pin (external bus width selection pin)

When accessing the external memory in memory expansion mode or microprocessor mode, the input level to the BYTE pin is used to select between 8-bit data bus and 16-bit data bus. (See (2) Data bus in section 2.5.4)

The external bus width becomes 8-bits when the BYTE pin is at "H" level. In this case, data read/write to the external area is always performed in 8-bit (1-byte) unit and the port P2 pins become the data (D_0~D_7) I/O pins. The use of 8 bit peripheral ICs is simplified by setting the bus width for external area to 8-bits.

The external bus width becomes 16-bits when the BYTE pin is at "L" level. In this case, data read/write to the external area is always performed in 16-bit (1 word) unit and the port P2 pins become the data I/O pins for the low order byte (even address data: D_0~D_7) of a 16-bit data and the port P1 pins become the data I/O pins for the high order byte (odd address data: D_8~D_{15}) of a 16-bit data.

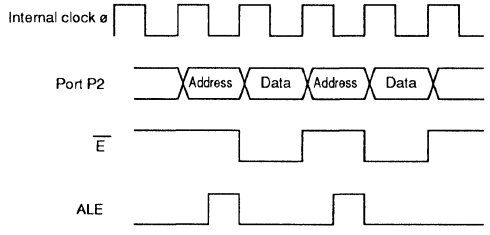
The data width is always 16-bits when accessing the internal memory area regardless of the BYTE pin level.

(2)Wait bit

The wait bit (processor mode register bit 2) is provided to attach slower memory when expanding external memory or I/O in memory expansion mode or microprocessor mode. When the wait bit is "0", a wait for external area access is enabled (one-shot wait mode) and bus operation is performed at 1/2 the bus cycle ($f(X_{IN})/4$) during no wait. When the wait bit is "1", bus operation becomes no wait mode and bus cycle is $f(X_{IN})/4$.

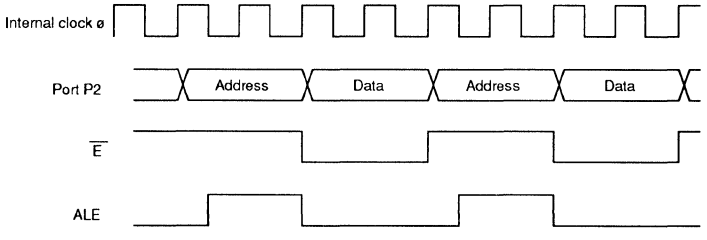
The wait bit is cleared during reset and the system starts in one-shot wait mode. Internal memory access is always performed at no wait because this bit is ignored.

(a) Waveform when the external memory area is accessed with the wait bit set



This waveform is always used when accessing the internal memory.

(b) Waveform when the external memory area is accessed with the wait bit cleared



For internal memory area access, waveform (a) is used even when the wait bit is "0".

Fig.2.4.6 Effect of Wait Bit for External Access

2.5 Input/Output Pins

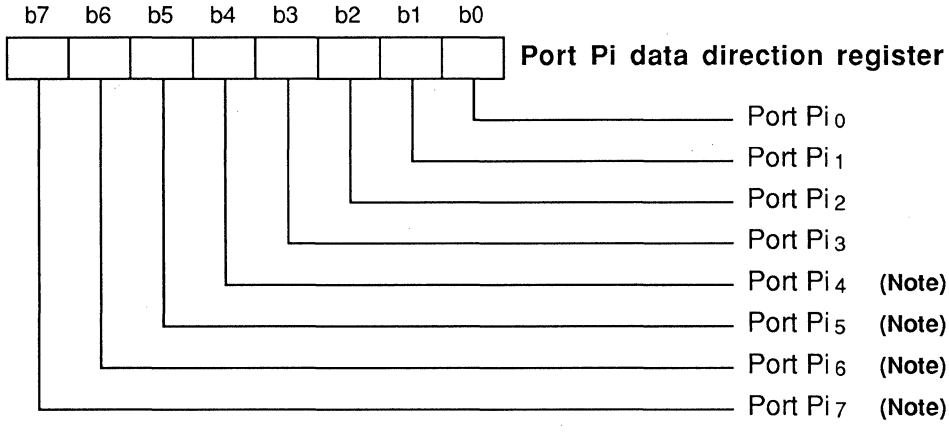
2.5.1 Programmable I/O ports

Each of the programmable I/O ports (P0~P8) has a data direction register which is used to select the input/output direction one bit unit. A port is used as an output pin when the corresponding bit in the port data direction register is "1" and as an input pin when the corresponding bit is "0". The port data direction register is allocated in the SFR area of bank 0. The input level/data can be read/write from a pin set to input/output by performing read/write to the port register, respectively.

(1)Data direction register

A data direction register corresponding to each port is allocated in the SFR area of bank 0. Each bit of the data direction register corresponds to a pin. Figure 2.5.1 shows the relationship between the direction register bits and pins. The I/O direction of the port is selected using the data direction register bits. The port is set to input pin when the corresponding bit is "0" and to output pin when the corresponding bit is "1".

At reset, the data direction registers are initialized to "00_{16}". Therefore, I/O ports are set to input.



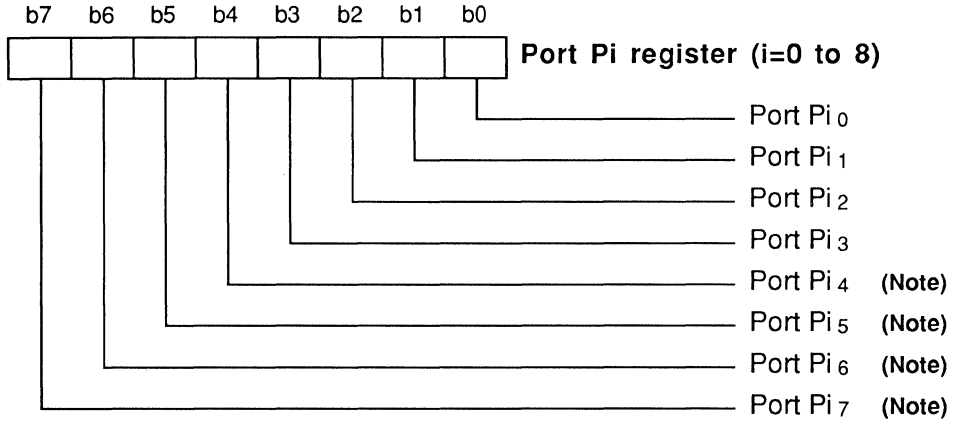
The I/O mode of each pin is determined by the value of the corresponding bit.
0: Input mode (the corresponding pin is an input pin)
1: Output mode (the corresponding pin is an output pin)

Note : The high-order 4 bits of port P3 data direction register are unused.

Fig.2.5.1 Relationship between the Port Data Direction register and Pins

(2)Port register

The port register is used to transfer data with external devices through the I/O ports. To output data from a port set to output, the data must be written to the corresponding bits of the port register. This data is written in the port latch and is output from the port that is set to output. If a port programmed for output is read, the content of the port latch is read. Therefore, the previously output value can be read correctly even when the output "H" voltage drops or "L" voltage rises due to external load. A pin programmed for input is floated and the value input to the pin can be read by reading the corresponding bit of the port register. If a value is written to a pin programmed for input, it is written in the port latch and the pin remains floating.



Data I/O is done by reading/writing the bit corresponding to the pin.

Note : Reading the high-order 4 bits of port P3 register always results in "0".

Fig.2.5.2 Relationship between the Port Register and Pins

2.5.2 Pin functions

Figure 2.5.3 shows the port peripheral circuits. The functions of some pins depend on the processor mode while others are not affected. This section describes those pins that are not affected by the processor mode. The next section describes the pin functions according to the processor mode.

(1)Effect of processor mode on pin functions

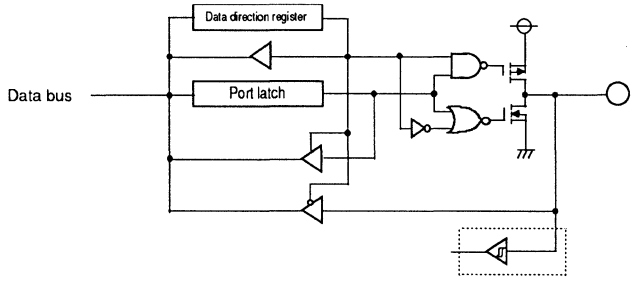
The function of some pins depends on the processor mode. Tables 2.5.1 and 2.5.2 show the pin functions according to processor mode. The function of port P1 also depends on the input level of the BYTE pin (external bus selection input pin). The details of the following pins are described in the next section.

Table 2.5.1 Pin Functions According to Processor Mode

Mode / Pin	Single-chip mode	Memory expansion mode and microprocessor mode	
		External 16-bit bus (BYTE="L")	External 8-bit bus (BYTE="H")
Port P0	Programmable I/O port	Address bus (A_0~A_7)	
Port P1	Programmable I/O port	Address bus (A_8~A_{15}) /Data bus (D_8~D_{15})	Address bus (A_8~A_{15})
Port P2	Programmable I/O port	Address bus (A_{16}~A_{23})/Data bus (D_0~D_7)	
Port P3	Programmable I/O port	$P3_0$....R/$\overline{W}$ output pin $P3_1$....$\overline{BHE}$ output pin $P3_2$....ALE output pin $P3_3$....$\overline{HLDA}$ output pin	
Port P4	Programmable I/O port **Note** : $P4_2$ pin can be programmed as ø output pin.	$P4_0$....$\overline{HOLD}$ input pin $P4_1$....$\overline{RDY}$ input pin $P4_2$~$P4_7$....Same as single-chip mode **Note** : $P4_2$ pin can be programmed as ø output pin.	
BYTE	Ignored	External bus selection input pin	

Ports $P0_7$~$P0_0$, $P1_7$~$P1_0$, $P2_7$~$P2_0$, $P3_3$~$P3_0$, $P4_6$~$P4_2$ (dotted area not included)

Ports $P4_0$, $P4_1$, $P4_7$, $P5_1$, $P5_3$, $P5_5$, $P5_7$, $P6_7$~$P6_1$, $P8_2$, $P8_6$ (dotted area included, however no hysteresis for $P8_2$ and $P8_6$)



Ports $P7_6$~$P7_0$ (dotted area not included)

Port $P7_7$ (dotted area included)

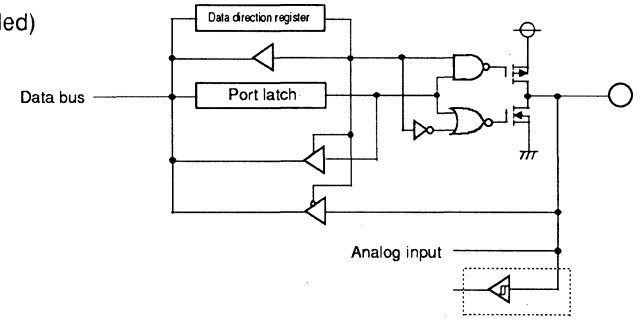
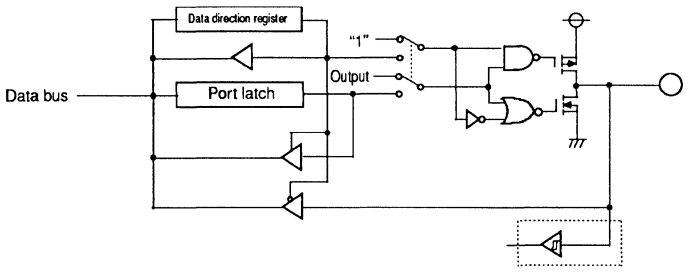


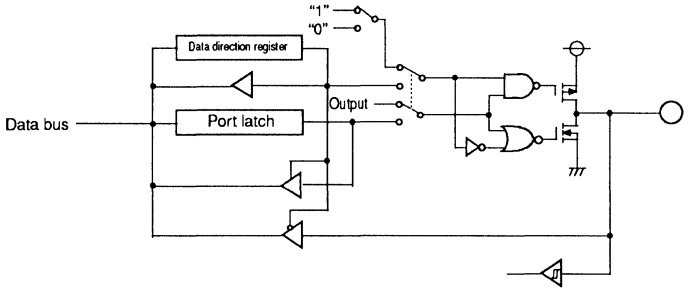
Fig 2.5.3 Port Peripheral Circuit (a)

Ports $P8_3$, $P8_7$
(dotted area not included)

Ports $P5_0$, $P5_2$, $P5_4$, $P5_6$, $P6_0$
(dotted area included)



Ports $P8_0$, $P8_1$, $P8_4$, $P8_5$



$\overline{E}$ output pin

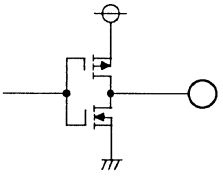


Fig 2.5.3 Port Peripheral Circuit (b)

Table 2.5.2 Functions of Ports P0~P4 by Processor Mode

Port \ Mode	CM_1	CM_0	Single-chip mode	Memory expansion mode	Microprocessor mode
			0	0	1
			0	1	0
Port P0			$\overline{E}$, $P0_7$~$P0_0$: I/O port	$\overline{E}$, $P0_7$~$P0_0$: A_7~A_0	
Port P1	BYTE="L"		$\overline{E}$, $P1_7$~$P1_0$: I/O port	$\overline{E}$, $P1_7$~$P1_0$: A_{15}~A_8 Data (odd)	
	BYTE="H"			$\overline{E}$, $P1_7$~$P1_0$: A_{15}~A_8	
Port P2	BYTE="L"		$\overline{E}$, $P2_7$~$P2_0$: I/O port	$\overline{E}$, $P2_7$~$P2_0$: A_{23}~A_{16} Data (even)	
	BYTE="H"			$\overline{E}$, $P2_7$~$P2_0$: A_{23}~A_{16} Data(odd,even)	
Port P3			$\overline{E}$, $P3_3$~$P3_0$: I/O port	$\overline{E}$, $P3_3$: $\overline{HLDA}$, $P3_2$: ALE, $P3_1$: $\overline{BHE}$, $P3_0$: R/$\overline{W}$	
Port P4	Processor mode register bit 7="0"		$\overline{E}$, $P4_7$~$P4_0$: I/O port	$\overline{E}$, $P4_7$~$P4_2$: I/O port, $P4_1$: $\overline{RDY}$, $P4_0$: $\overline{HOLD}$	
	Processor mode register bit 7="1"		$P4_2$: ø **Note** : Same as above except for $P4_2$.	$P4_2$: ø **Note** : Same as above except for $P4_2$.	

(2)Functions of pins unaffected by processor mode

Table 2.5.3 shows the functions of pins not affected by processor mode. The functions of these pins are the same in all modes.

Table 2.5.3 Functions of Pins Unaffected by Processor Mode

Pin	Function
Port P5	8-bit programmable I/O pin. (Also used as timer I/O pin.)
Port P6	8-bit programmable I/O pin. (Also used as timer I/O and external interrupt input pin.)
Port P7	8-bit programmable I/O pin. (Also used as analog input pin.)
Port P8	8-bit programmable I/O pin. (Also used as serial I/O pin.)
V_{CC}, V_{SS}	Supply voltage pins. 5V±10% is applied to V_{CC} and V_{SS} is connected to GND.
CNV_{SS}	This pin controls the processor mode. The processor mode is selected by changing the input voltage level to this pin (except change after reset start). See "Section 2.4.2 Processor Modes" for detail information concerning the processor mode. In single-chip mode, this pin must be set to the same level as V_{SS}.
AV_{CC}, AV_{SS}	A-D conversion circuit supply voltage pins. Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS}.
V_{REF}	Reference voltage input pin for the A-D converter. Analog input voltage from Vss level to the level of this pin can be converted. Apply any voltage up to V_{CC} level to this pin.
X_{IN}, X_{OUT}	Clock I/O pin for the internal oscillator circuit. The M37700 is equipped with an internal clock generator and the oscillating frequency is set by connecting a ceramic resonator or quartz crystal oscillator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open. The maximum clock input frequency is 8MHz for M37700M2-XXXFP and 16MHz for M37700M2AXXXFP.
$\overline{RESET}$	Reset input pin. Set this pin to "L" level to enter the reset state. Then when this pin is returned to "H" level, the reset state is deactivated and program loading starts from the address set in the reset vector. See "Chapter 3. Reset" for the contents of registers immediately after returning from reset.
$\overline{E}$	Internal bus cycle $\overline{E}$ is output.

Ports P5 to P8 have the programmable I/O port function as well as special functions such as I/O pins for external interrupt, timer, A-D convertor, and serial I/O. When these multiple function ports are used as special function output pins, they are automatically set to output mode, but when they are used as special function input pins, the port direction register must be set to input mode. In this case, the pin input level can be read from the port register. The methods for selecting special functions are described under each function.

As for port $P4_2$, an internal clock ø is output from this pin when the processor mode register bit 7 is set.

All ports function as programmable I/O port immediately after returning from reset.

2.5.3 Single-chip mode pin functions

In single-chip mode, 68 ports can be used as programmable I/O pins (using multiple function pins as I/O ports).

Figure 2.5.4 shows the I/O pins during single-chip mode.

Table 2.5.4 shows the functions of processor mode dependent pins (ports P0~P4, BYTE) during single-chip mode. See section "2.5.1 Programmable I/O Ports" for the programmable I/O port functions.

See table 2.5.3 for the functions of other pins.

Table 2.5.4 Functions of Ports P0~P4 and BYTE Pin in Single-Chip Mode

Pin	Functions	Pin	Functions
Port P0	8-bit programmable I/O port	Port P3	4-bit programmable I/O port
Port P1	8-bit programmable I/O port	Port P4	8-bit programmable I/O port
Port P2	8-bit programmable I/O port	BYTE	Ignored in single-chip mode

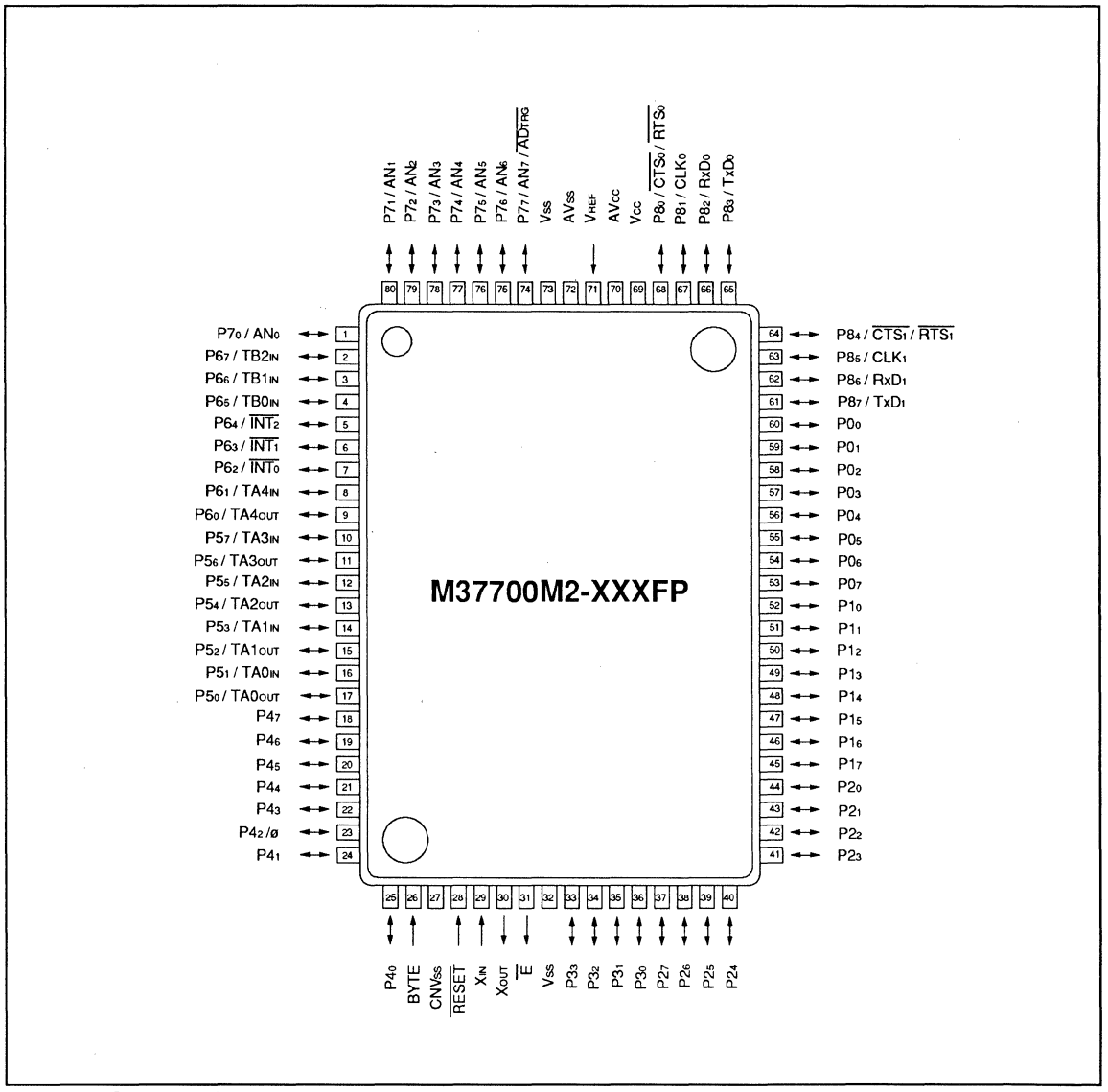


Fig.2.5.4 Single-chip Mode Pin Connection Diagram

Port $P4_2$ can be programmed to output ø by setting the processor mode register (PMR). When the ø output selection bit in the processor mode register is set to enable, ø output starts at the rising edge of bus cycle $\overline{E}$ that was pulled "L" to write "1" in the ø output selection bit.

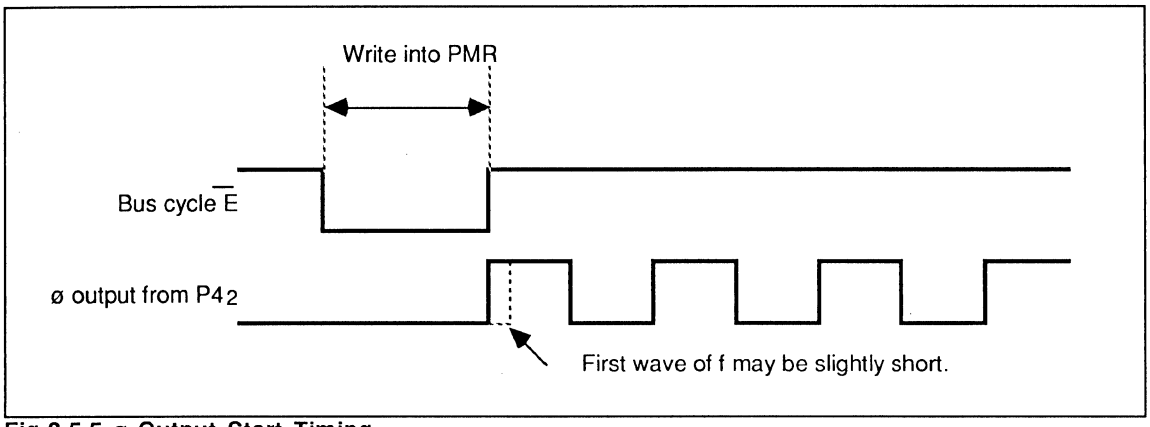


Fig.2.5.5 ø Output Start Timing

2.5.4 Memory expansion and microprocessor mode pin functions

The only difference between the memory expansion mode and microprocessor mode is whether access to internal ROM is disabled or not. (See section "2.4.2 Processor Modes".) The function of each pin is identical in memory expansion mode and microprocessor mode.

In memory expansion mode and microprocessor mode, there are 38 I/O ports (ports $P4_2$~$P4_7$ and P5~P8) as shown in Figure 2.5.6. The internal address bus and data bus can be used externally.

Table 2.5.5 shows the functions of processor mode dependent pins (ports P0~P4 and BYTE) in memory expansion mode and microprocessor mode.

See Table 2.5.3 for the functions of other pins.

Table 2.5.5 Pin Functions in Memory Expansion and Microprocessor Mode

Pin	Functions	Pin	Functions
Port P0	Address bus	Port P3	External memory control signal output
Port P1	Address bus/data bus (Note1)	$P4_1$ and $P4_0$	External control signal input (Note 2)
Port P2	Address bus/data bus	BYTE	External bus width selection signal input

Note 1 : This may be address bus only depending on the input level of the BYTE pin.

Note 2 : In memory expansion mode and microprocessor mode, the data direction registers of ports $P4_0$ and $P4_1$ must be set to input mode.
If port $P4_0$ or $P4_1$ is read in these mode, the level of $P4_0$ or $P4_1$ is obtained.

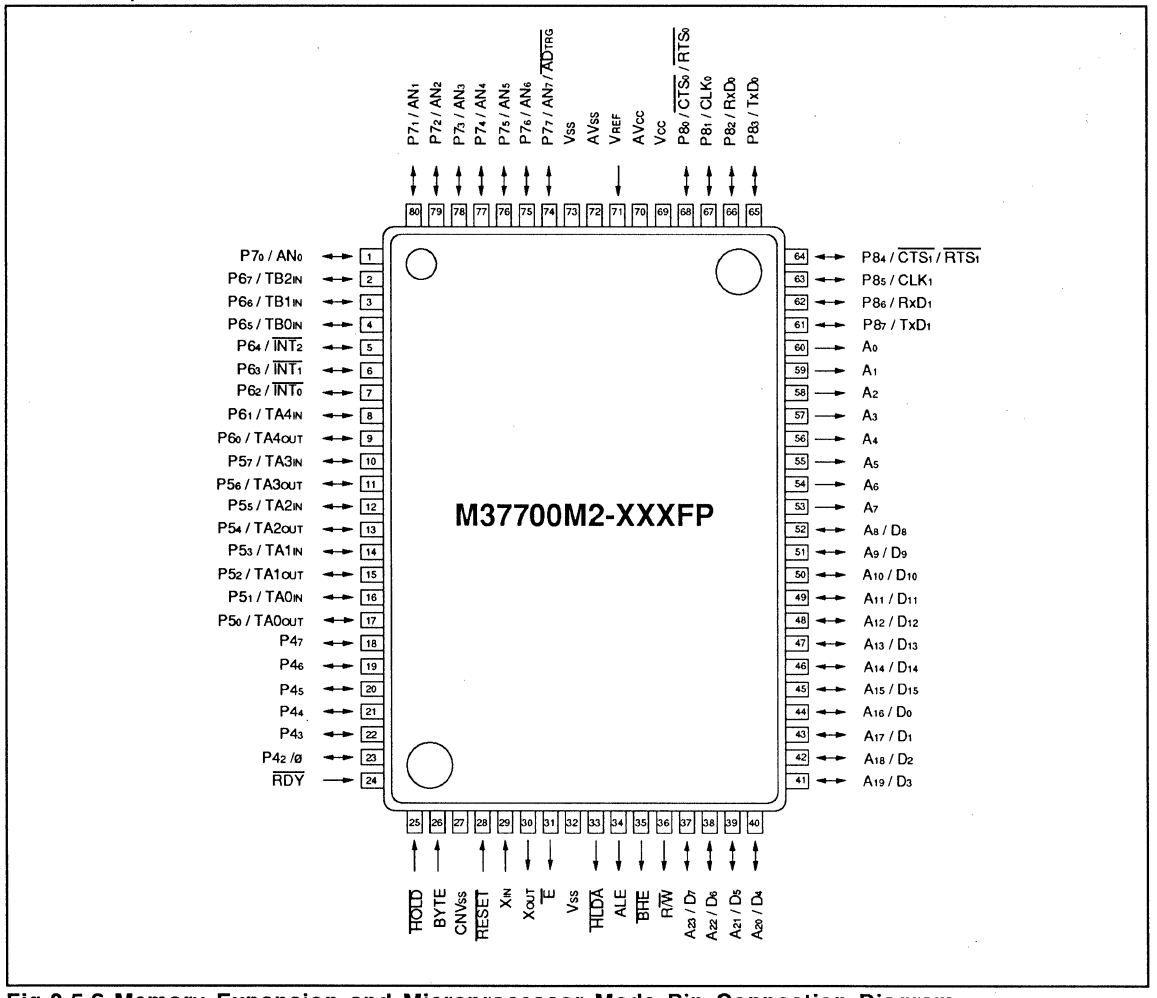


Fig.2.5.6 Memory Expansion and Microprocessor Mode Pin Connection Diagram

The functions of each pin in memory expansion mode and microprocessor mode are described below.

(1)Address bus Ports P0, P1, and P2

Ports P0, P1, and P2 become address signal output pins and lose their programmable I/O port functions.

The M37700 allows direct access to 16M-byte memory space from address 000000_{16} to $FFFFFF_{16}$. Therefore, 24 address signals are output externally in memory expansion mode and microprocessor mode which allow memory and I/O to be expanded externally.

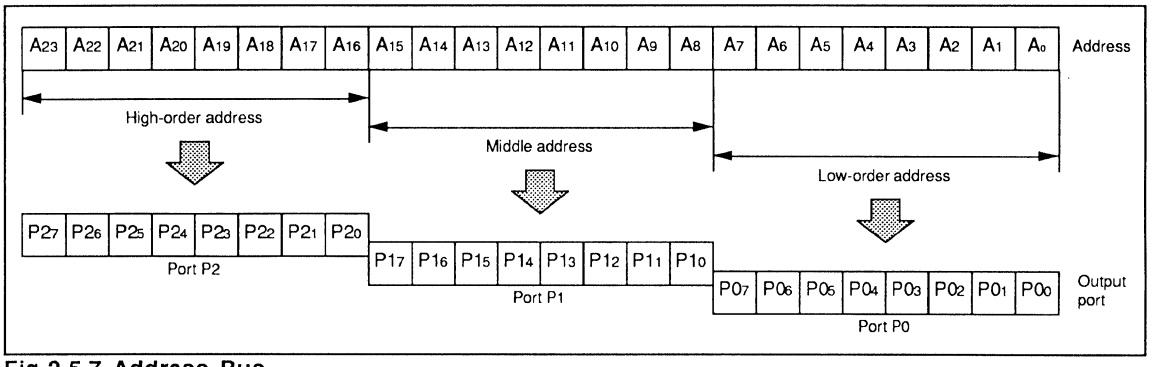


Fig.2.5.7 Address Bus

(2)Data bus

In addition to address signal (high-order and middle address bus) output function, ports P1 and P2 also function as data I/O pins. The level of the BYTE pin can be used to select between 8-bit or 16-bit data bus width.

●When the BYTE pin is at "L" (16-bit external bus width)

When the BYTE pin is at "L", the external bus width is 16 bits and even address data and odd address data are output simultaneously.

Ports P1 and P2 are used as address bus and data bus and multiplexed (address signal and data signal) signals are output from these ports.

Port P1 performs time division multiplexing of address data (A_{15}~A_8) output and odd address data input/output. Middle address data is output while $\overline{E}$ is at "H", and odd address data input/output is performed while $\overline{E}$ is at "L".

Similarly, Port P2 performs time division multiplexing of address data (A_{23}~A_{16}) output and even address data input/output. High-order address data is output while $\overline{E}$ is at "H", and even address data input/output is performed while $\overline{E}$ is at "L".

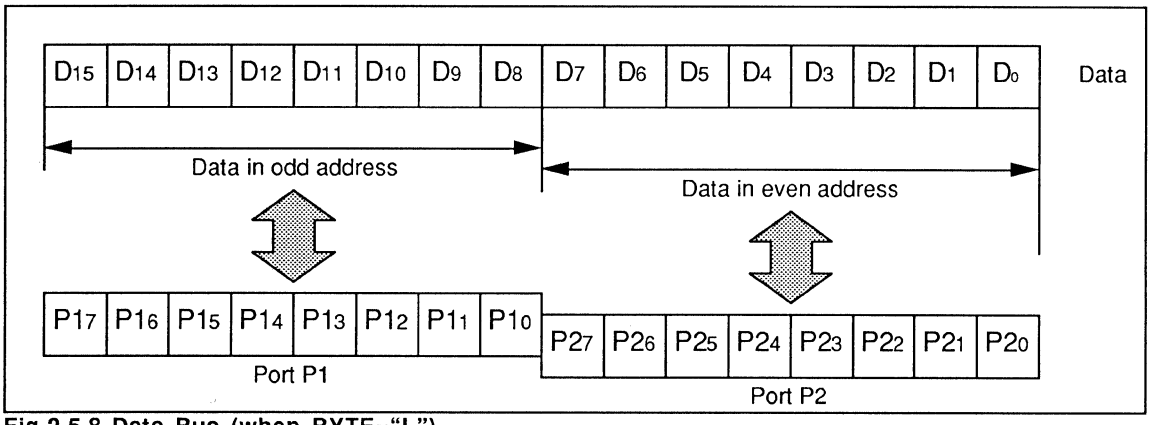


Fig.2.5.8 Data Bus (when BYTE="L")

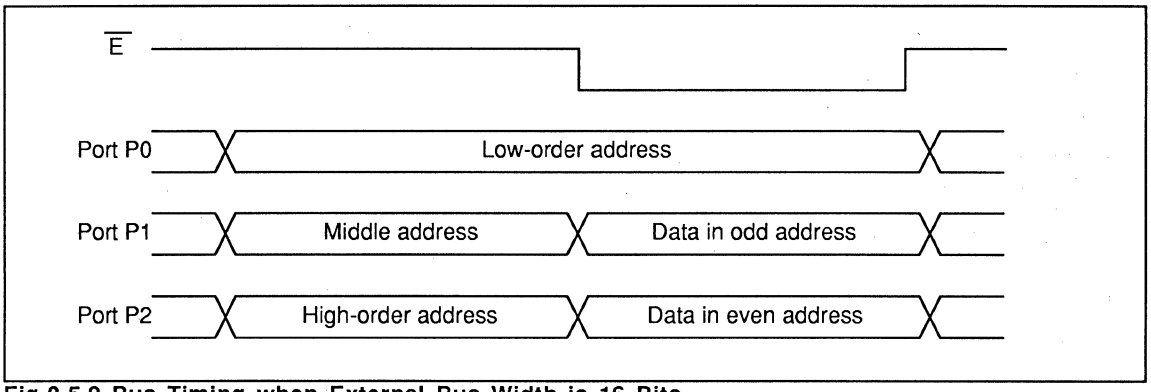


Fig.2.5.9 Bus Timing when External Bus Width is 16 Bits

●When the BYTE pin is at "H" (8-bit external bus width)

When the BYTE pin is at "H", the external bus width becomes 8 bits and the output of data and address data (A_{23}~A_{16}) are multiplexed.

Address data is output while $\overline{E}$ is at "H", and 8-bit data is input/output when $\overline{E}$ is at "L".

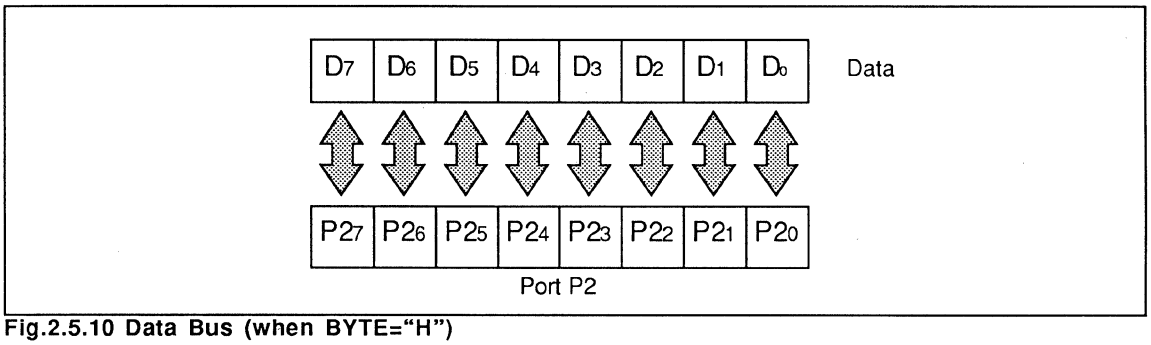


Fig.2.5.10 Data Bus (when BYTE="H")

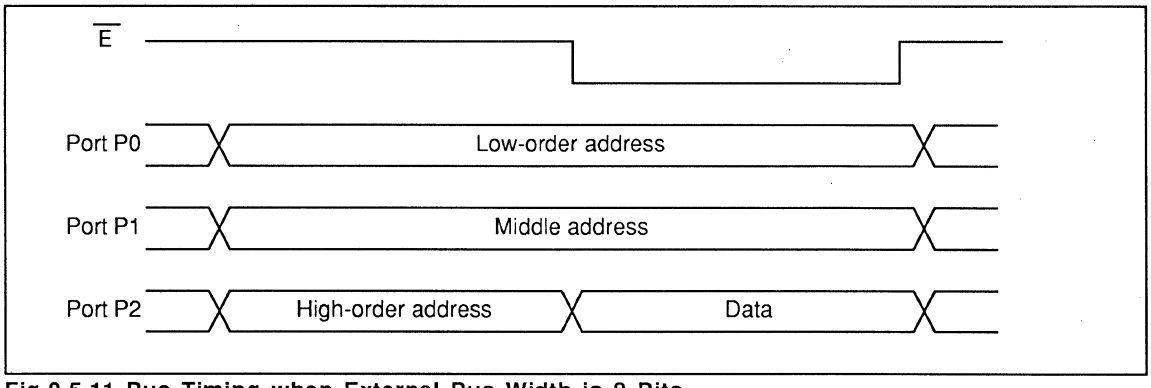


Fig.2.5.11 Bus Timing when External Bus Width is 8 Bits

(3)R/$\overline{W}$ output pin

A read/write signal indicating the data bus direction is output. The data bus is read when the level of this pin is at "H", and data is written to data bus when it is at "L". This signal is used for external memory input/output requests.

(4)$\overline{BHE}$ output pin

A byte high enable signal is output. This pin is at level "L" when an odd number address is accessed. This signal is used to expand the 8-bit memory and I/O when the external bus is used at 16-bit width.

(5)ALE signal output pin

This signal is used to obtain only address signal from the multiplexed signals of ports P1 and P2. A latch is opened externally when the ALE signal is at "H" to obtain the address data and the latched content is held while the ALE signal is at "L".

(6)$\overline{HOLD}$ input pin

This pin is used to input hold request signals. The microcomputer is held while this pin is at "L". When the hold request signal becomes "L", ϕ_{CPU} (CPU clock; $f(X_{IN})/2$) is stopped and the bus cycle $\overline{E}$ stops at "H". The hold status is cancelled and processing continues when the $\overline{HOLD}$ pin level becomes "H". Note that internal peripherals can continue to operate because only ϕ_{CPU} is stopped (internal clock ϕ is not stopped). However, the watchdog timer is stopped during a hold.
Table 2.5.6 shows the port status during a hold.

Table 2.5.6 Port Status During Hold

Port	Status during hold
P0~P2, $P3_0$, $P3_1$	Floating
$P3_2$, $P3_3$	Outputs "L" level
$P4_3$~$P4_7$, P5~P8	Holds the port status when "L" is applied to $\overline{HOLD}$ pin

(7)$\overline{HLDA}$ signal output pin

This pin is used to externally output a hold acknowledge signal. The hold acknowledge signal indicates that "L" is input to the $\overline{HOLD}$ pin and the microcomputer is in a hold state. An "L" level is output from this pin while the microcomputer is in a hold state.

(8)$\overline{RDY}$ signal input pin

This is a ready signal input pin. The bus cycle $\overline{E}$ can be stopped (ready state) when "L" is input to this pin. The port and bus status when "L" is input to the $\overline{RDY}$ pin is held while ready. The $\overline{RDY}$ signal is used when slow memory is externally connected.

(9)$\overline{E}$ output pin

This is the enable signal output pin. Data I/O is performed when the output of this pin is at "L". This signal controls the time division multiplexing of address information and data.

(10)BYTE pin

This is the byte enable signal input pin. The input level to this pin determines whether the external memory is used with 16-bit data width or 8-bit. When the BYTE pin input level is at "L", the data width is 16 bits and ports P1 and P2 become the data I/O pins (data bus). When the BYTE pin input level is at "H", the data width is 8 bits and port P2 becomes the data I/O pin (data bus). However, the data width is always 16 bits regardless of the BYTE pin level when accessing an internal memory.

(11)CNV_{SS} pin

This pin controls the microprocessor operating mode. Memory expansion and microprocessor modes are selected by resetting the microcomputer after setting this pin to the same level as the V_{SS} pin and then changing the processor mode bit in the processor mode register.
The microprocessor mode can also be selected by setting this pin to the same level as the V_{CC} pin and then resetting. This pin must be set to Vcc level for external ROM version microprocessors such as the M37700SFP (see "Section 2.4.2 Processor Modes").

2.6 Interrupts

2.6.1 Interrupt functions

The M37700 has 19 different sources of interrupts. When an interrupt occurs, a branch is made to the address (branch address) corresponding to the source. Therefore, a branch address corresponding to each interrupt must be stored at the address (interrupt vector address) corresponding to each interrupt at addresses $FFD6_{16}$ to $FFFF_{16}$ (interrupt vector table) in bank 0. These branch addresses are the start addresses of the interrupt handling routines (interrupt service routine).

When interrupt processing completes, the control must be returned to the original routine to resume processing. Therefore, the contents of the program counter (PC), program bank register (PG), and the processor status register (PS) just before an interrupt are automatically stored in the stack area (register saving). Then when interrupt processing is completed, the **RTI** instruction (return from interrupt service routine) can be used to restore the contents of the PC, PG, and PS registers to the respective registers and resume the original routine.

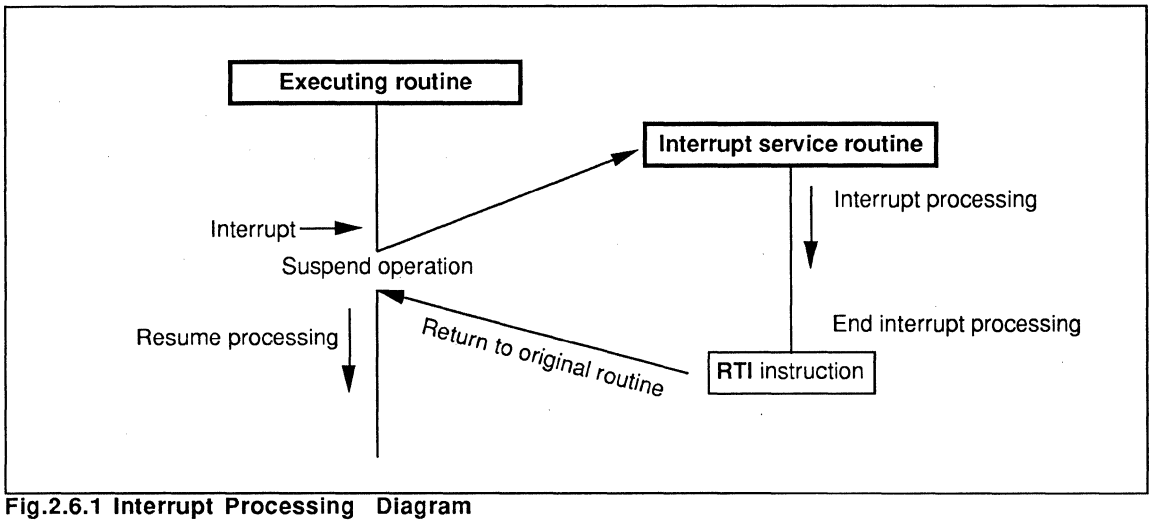


Fig.2.6.1 Interrupt Processing Diagram

2.6.2 Sources of interrupts

Table 2.6.1 shows the sources of interrupts and the corresponding vector address. Store the address of the interrupt service routine at the vector address shown in this table.

Table 2.6.1 Interrupt Sources and Vector Address

Interrupt source	Vector address		Remarks
	High-order address	Low-order address	
Reset (Note 1)	$00FFFF_{16}$	$00FFFE_{16}$	Non-maskable
Zero divide	$00FFFD_{16}$	$00FFFC_{16}$	Non-maskable software interrupt
BRK instruction	$00FFFB_{16}$	$00FFFA_{16}$	Non-maskable software interrupt
$\overline{DBC}$ (Note 2)	$00FFF9_{16}$	$00FFF8_{16}$	Not available to general user
Watchdog timer	$00FFF7_{16}$	$00FFF6_{16}$	Non-maskable interrupt
$\overline{INT0}$	$00FFF5_{16}$	$00FFF4_{16}$	External interrupt due to $\overline{INT_0}$ pin input signal
$\overline{INT1}$	$00FFF3_{16}$	$00FFF2_{16}$	External interrupt due to $\overline{INT_1}$ pin input signal
$\overline{INT2}$	$00FFF1_{16}$	$00FFF0_{16}$	External interrupt due to $\overline{INT_2}$ pin input signal
Timer A0	$00FFEF_{16}$	$00FFEE_{16}$	Timer A0 internal interrupt
Timer A1	$00FFED_{16}$	$00FFEC_{16}$	Timer A1 internal interrupt
Timer A2	$00FFEB_{16}$	$00FFEA_{16}$	Timer A2 internal interrupt
Timer A3	$00FFE9_{16}$	$00FFE8_{16}$	Timer A3 internal interrupt
Timer A4	$00FFE7_{16}$	$00FFE6_{16}$	Timer A4 internal interrupt
Timer B0	$00FFE5_{16}$	$00FFE4_{16}$	Timer B0 internal interrupt
Timer B1	$00FFE3_{16}$	$00FFE2_{16}$	Timer B1 internal interrupt
Timer B2	$00FFE1_{16}$	$00FFE0_{16}$	Timer B2 internal interrupt
UART0 receive	$00FFDF_{16}$	$00FFDE_{16}$	Valid only when the UART0 function is selected
UART0 transmit	$00FFDD_{16}$	$00FFDC_{16}$	
UART1 receive	$00FFDB_{16}$	$00FFDA_{16}$	Valid only when the UART1 function is selected
UART1 transmit	$00FFD9_{16}$	$00FFD8_{16}$	
A-D conversion	$00FFD7_{16}$	$00FFD6_{16}$	Internal interrupt that occurs when A-D conversion completes.

Note1: Reset is included because its operation is identical to an interrupt.
Note2: The $\overline{DBC}$ interrupt is a debug control interrupt and is not available to general users.

Each interrupt source is described below.

(1)Internal interrupt

Table 2.6.2 shows the internal interrupt sources.

Table 2.6.2 Internal Interrupt Sources

Interrupt	Interrupt source
Zero divide	Occurs when 0 is specified as the divisor for a **DIV** instruction. (See "MELPS 7700 Software Manual")
BRK instruction	Occurs when a **BRK** instruction is executed. (See "MELPS 7700 Software Manual")
Watchdog timer	Occurs when the topmost bit of the 12-bit watchdog timer becomes "0". (See section "2.12 Watchdog Timer")
Timer Ai	Occurs when timer Ai (i=0 to 4) overflows. (See section"2.7 Timer A")
Timer Bi	Occurs when timer Bi (i=0 to 2) overflows. (See section"2.8 Timer B")
UARTi receive	Occurs during UARTi (i=0,1) receive (See section "2.9 Serial I/O")
UARTi transmit	Occurs during UARTi (i=0,1) transmit (See section "2.9 Serial I/O")
A-D conversion	Occurs when A-D conversion completes (See section "2.11 A-D Converter")

(2)External interrupt ($\overline{INT_0}$~$\overline{INT_2}$)

These are interrupts that are caused by input level or input edge to pins $\overline{INT_0}$ to $\overline{INT_2}$. The interrupt sources can be selected using bits 4 and 5 of the $\overline{INTi}$ interrupt control register shown in Figure 2.6.2. Pins $\overline{INT_0}$~$\overline{INT_2}$ are shared with ports $P6_2$~$P6_4$. Therefore, the corresponding bit in the port P6 data direction register must be cleared to "0" in order to use these pins as external interrupt input pins. If the $\overline{INTi}$ interrupts are not used, the $\overline{INTi}$ interrupt priority should be set to "0" because the $\overline{INTi}$ interrupts always monitor the status of $P6_2$~$P6_4$ pins to raise interrupt requests.

The input signal to the $\overline{INTi}$ pins must have pulse width greater than 250ns at "H" or "L" regardless of the source oscillating frequency ($f(X_{IN})$).

Table 2.6.3 $\overline{INTi}$ Interrupt Sources

b5	b4	Interrupt Source
0	0	Falling edge of the signal input to the $\overline{INTi}$ pin
0	1	Rising edge of the signal input to the $\overline{INTi}$ pin
1	0	When the $\overline{INTi}$ pin status becomes "H"
1	1	When the $\overline{INTi}$ pin status becomes "L"

2.6.3 Interrupt control

The enabling and disabling of interrupts are controlled by the interrupt request bit, interrupt priority level, processor interrupt priority (IPL), and interrupt disable flag (I) (excluding some software interrupts). The interrupt disable flag and the processor interrupt priority level are assigned to the processor status register (PS). The interrupt request bit and the interrupt priority level are assigned to the interrupt control register of the respective interrupt. Figure 2.6.3 shows the structure of the interrupt control register. However, there is no interrupt control register for non-maskable interrupts such as zero divide interrupt, **BRK** instruction interrupt, and watchdog timer interrupt.

- Non-maskable interrupt: An interrupt that causes branch to the interrupt service routine regardless of the interrupt control flags.
- Maskable interrupt: An interrupt that can be disabled with the interrupt control flags.

The interrupt control flags are described below.

(1)Interrupt disable flag (I flag)

The interrupt disable flag (I flag) is assigned to bit 2 of the processor status register. This flag can be used to disable all maskable interrupts. All maskable interrupts are masked when the I flag is set and enabled when it is cleared. This flag is set during reset and must be cleared if interrupts are to be enabled.

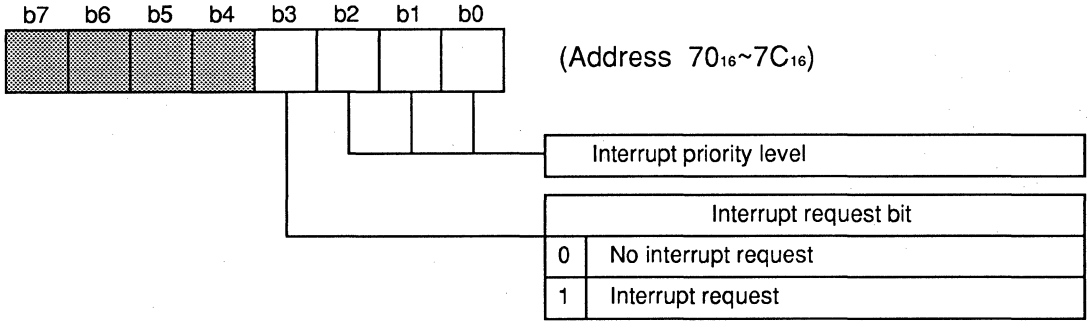
(2)Interrupt request bit

When an interrupt occurs, the interrupt request bit which is assigned to bit 3 of the corresponding interrupt control register is set. The interrupt request bit remains set until the interrupt is accepted and is cleared when the interrupt is accepted. This flag is used to indicate that an interrupt has occurred. This bit can be set and cleared from a program.

Address	
70_{16}	A-D conversion interrupt control register
71_{16}	UART0 transmit interrupt control register
72_{16}	UART0 receive interrupt control register
73_{16}	UART1 transmit interrupt control register
74_{16}	UART1 receive interrupt control register
75_{16}	Timer A0 interrupt control register
76_{16}	Timer A1 interrupt control register
77_{16}	Timer A2 interrupt control register
78_{16}	Timer A3 interrupt control register
79_{16}	Timer A4 interrupt control register
$7A_{16}$	Timer B0 interrupt control register
$7B_{16}$	Timer B1 interrupt control register
$7C_{16}$	Timer B2 interrupt control register
$7D_{16}$	$\overline{\text{INT0}}$ interrupt control register
$7E_{16}$	$\overline{\text{INT1}}$ interrupt control register
$7F_{16}$	$\overline{\text{INT2}}$ interrupt control register

Fig.2.6.2 Interrupt Control Register Memory Map

●A-D conversion, UART0, 1 transmit, UART0, 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers



●$\overline{INT0}$~$\overline{INT2}$ interrupt control register

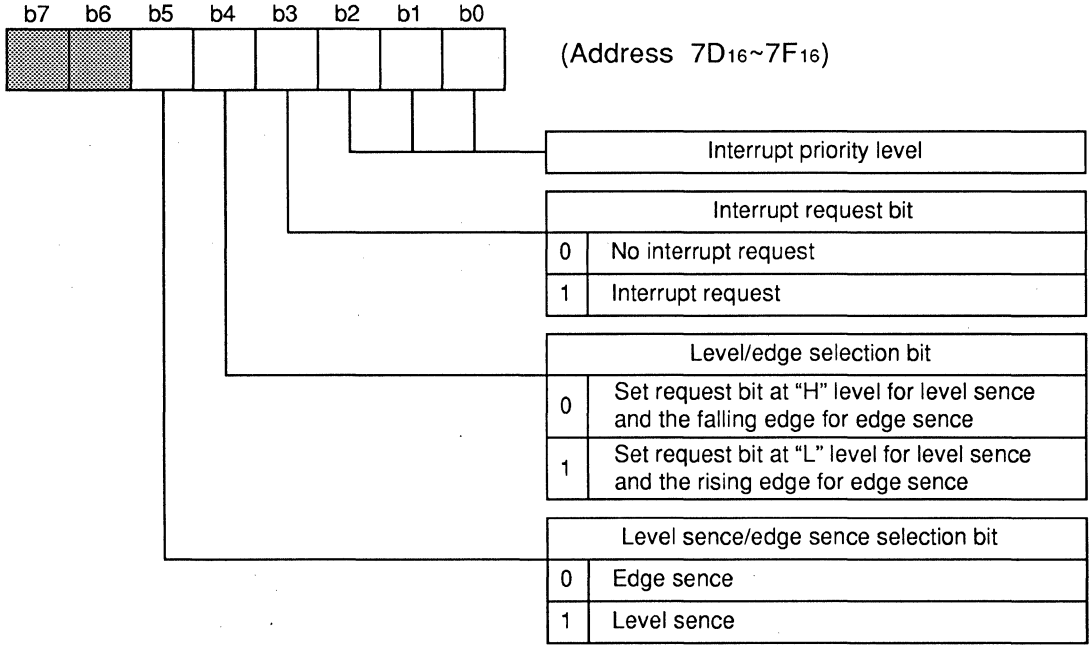


Fig.2.6.3 Interrupt Control Register Structure

(3)Interrupt priority level and processor interrupt priority level (IPL)

An interrupt priority level between 0 and 7 can be assigned to each interrupt using the interrupt priority level selection bits which are assigned to bits 0 to 2 of each interrupt control register. When an interrupt is raised, this priority level is compared with the processor IPL in the processor status register.

An interrupt is enabled when its interrupt priority level is greater than the IPL. Therefore, an interrupt can be disabled by setting its priority level to 0.

The interrupt disable flag, interrupt request bit, interrupt priority level, and IPL are independent of each other and do not affect other flags. An interrupt occurs only when the condition of these flags satisfy the interrupt occurrence condition. The combination of these flags can control the variety interrupt priority operation by programming.

Table 2.6.4 Interrupt Priority Level

Interrupt control register			Interrupt Priority Level	Priority
b2	b1	b0		
0	0	0	Level 0 (Interrupt disabled)	—
0	0	1	Level 1	Low
0	1	0	Level 2	
0	1	1	Level 3	
1	0	0	Level 4	
1	0	1	Level 5	
1	1	0	Level 6	
1	1	1	Level 7	High

Table 2.6.5 Interrupt Enable Level and Enabled Interrupts

IPL_2	IPL_1	IPL_0	Enabled interrupt priority level
0	0	0	Enable level 1 and above interrupts
0	0	1	Enable level 2 and above interrupts
0	1	0	Enable level 3 and above interrupts
0	1	1	Enable level 4 and above interrupts
1	0	0	Enable level 5 and above interrupts
1	0	1	Enable level 6 and above interrupts
1	1	0	Enable level 7 interrupts
1	1	1	Disable all maskable interrupts

IPL_0: Processor status register bit 8
IPL_1: Processor status register bit 9
IPL_2: Processor status register bit 10

2.6.4 Interrupt order

All interrupts are assigned a priority order. When all interrupts are enabled and more than one interrupt occurs during the same sampling interval (interval in which interrupt requests are checked), the one with the highest priority is accepted.

The priority order of all of the 19 sources except software interrupts (zero divide and **BRK** instruction interrupt) and watchdog timer interrupt can be set from a program using the interrupt priority level bits in the interrupt control register. Reset (the highest priority) and watchdog timer priorities are set by the hardware. Figure 2.6.4 shows the hardware interrupt priorities.

The M37700 is equipped with an interrupt priority order detection circuit to select the highest priority when more than one interrupt occurs within the same sampling interval.

Note : When a **BRK** instruction is executed or a zero divide is performed in an interrupt service routine, a **BRK** interrupt or a zero divide interrupt occurs and that interrupt is serviced. However, if multiple interrupts are enabled by setting the I flag to "0", interrupts with priority higher than IPL are accepted because the IPL is not changed. Furthermore, the watchdog timer interrupt is always enabled.

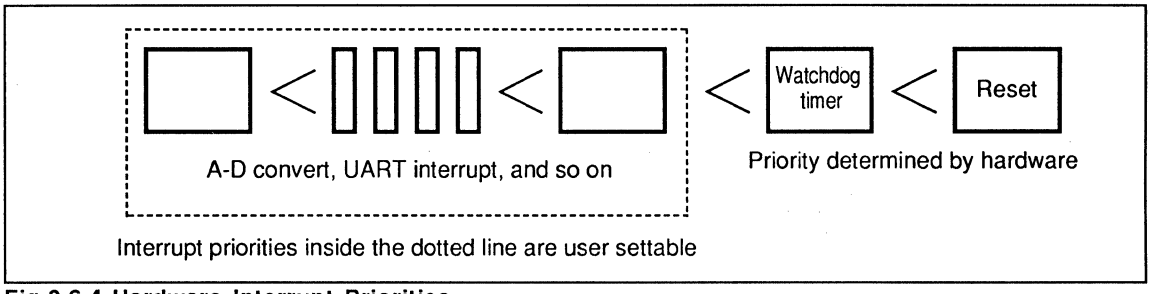


Fig.2.6.4 Hardware Interrupt Priorities

2.6.5 Interrupt priority detection circuit

Figure 2.6.5 shows the interrupt priority detection circuit.

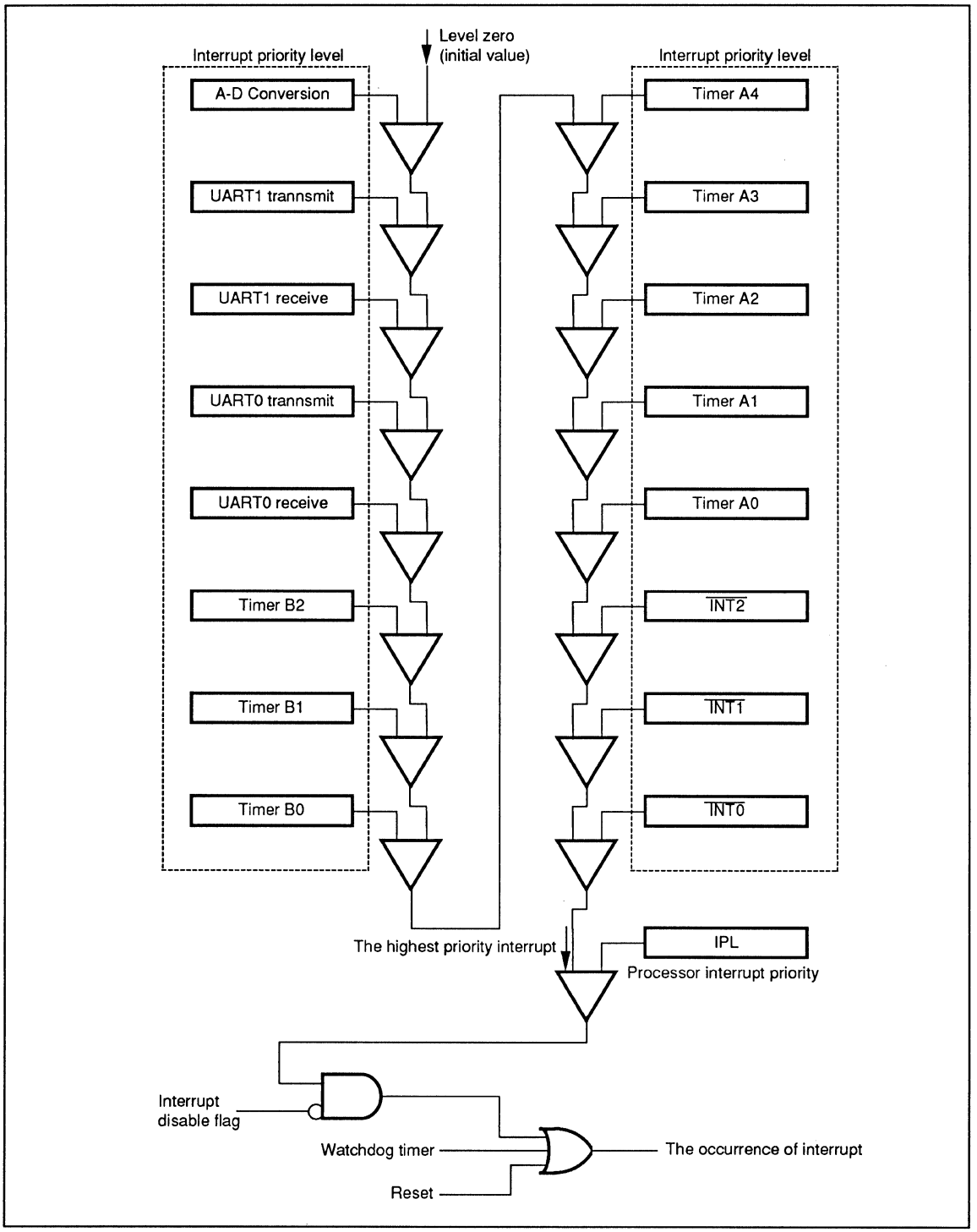


Fig.2.6.6 Interrupt Priority Detection Circuit

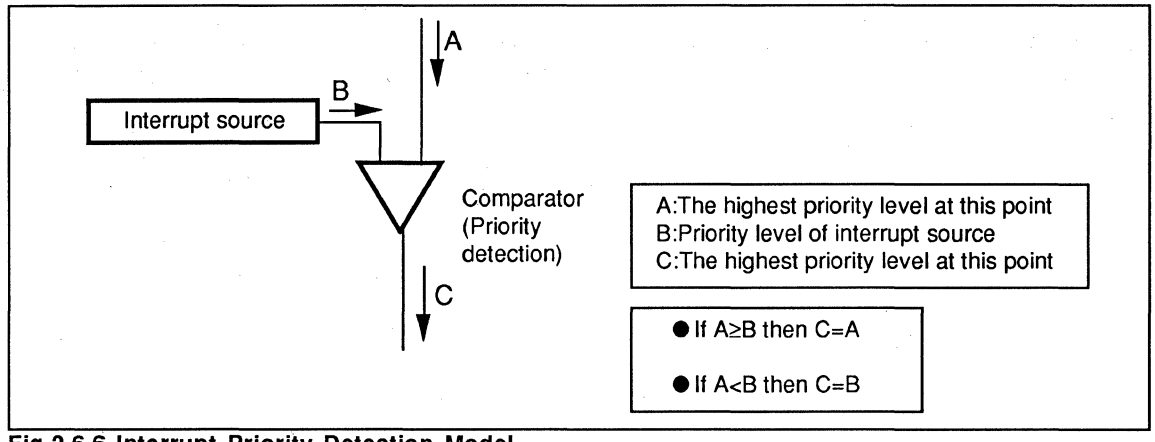


Fig.2.6.6 Interrupt Priority Detection Model

The interrupt priority level of the requested interrupt (B in Figure 2.6.6) is compared ,in the order shown in Figure 2.6.5, with the highest priority interrupt at this point (A in Figure 2.6.6) and the higher level interrupt is sent out as C to be compared with the next interrupt (A is initially 0). Unrequested interrupts are not compared and A is passed to C. If the priority levels of A and B are the same, A is selected. Therefore, the following relation exists if the software set priority levels are the same.

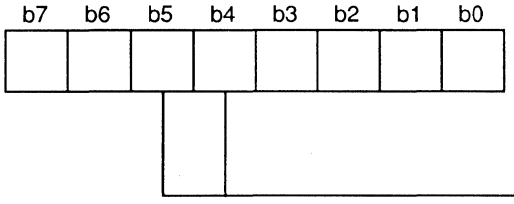
$\overline{INT0}$ < $\overline{INT1}$ < $\overline{INT2}$ < Timer A0 < Timer A1 < Timer A2 < Timer A3 < Timer A4 < Timer B0 < Timer B1 < Timer B2 < UART0 receive < UART0 transmit < UART1 receive < UART1 transmit < A-D conversion

As the result of this comparison, the interrupt with the highest priority is selected when there are multiple interrupts within the same sampling interval. Then that interrupt is enabled and its interrupt service routine is executed if its interrupt priority level is higher than the processor interrupt level (IPL) and the interrupt disable flag is "0".
The detection of interrupt priority level is synchronized with the sampling pulse generated during the operation code fetch cycle. While the interrupt level is being checked, the interrupt request bit and the interrupt priority level are latched so that they do not changed. They are sampled at the first half of the operation code fetch cycle and latched from the last half to the end of the level detection. Note that while the priority is being checked, no sampling pulse is generated even when it is the operation code fetch cycle (See Figure 2.6.8).

2.6.6 Interrupt priority detection time

With the M37700, the time it takes for the interrupt priority detection circuit to determine the level of an interrupt can be set by software. This is performed by setting the interrupt priority detection time selection bits in the processor mode register (PMR). Table 2.6.7 shows detection time corresponding to each combination of PMR bits 4 and 5. Figure 2.6.8 shows the relationship between the interrupt priority detection time and the sampling pulse.

After a reset, the interrupt priority level detection time selection bits are initialized to "00" and seven cycle mode is selected.

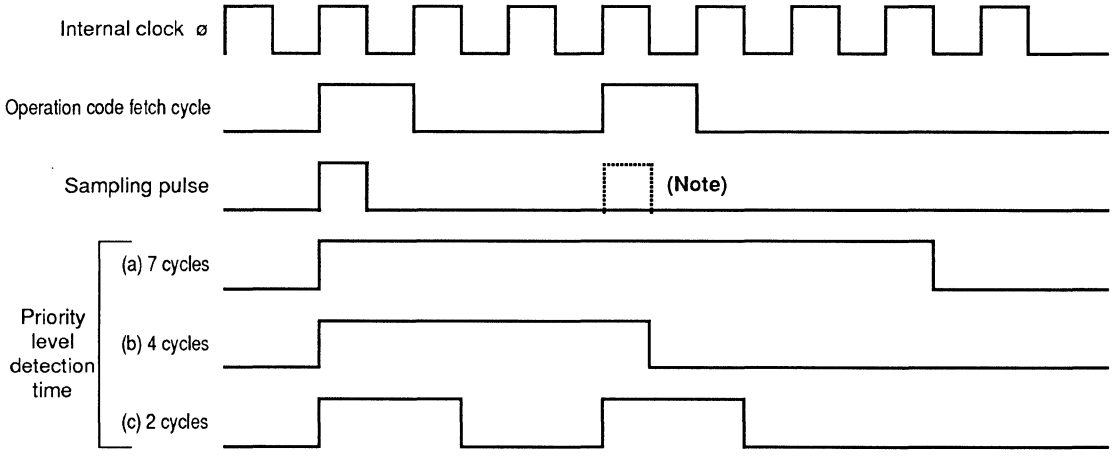


Processor mode register (Address $5E_{16}$)

Interrupt priority detection time selection bits		
b5	b4	Detection time
0	0	7 cycles at internal clock ø ((a) in Fig.2.6.8)
0	1	4 cycles at internal clock ø ((b) in Fig.2.6.8)
1	0	2 cycles at internal clock ø ((c) in Fig.2.6.8)
1	1	Not selection

Note : 1 cycle = ø = $f(X_{IN})/2$

Fig.2.6.7 Interrupt Priority Level Detection Time



Note: Pulse exists if 2 cycles is selected for priority level detection time.

Fig.2.6.8 Interrupt Priority Level Detection Time

2.6.7 Interrupt processing sequence

When an interrupt is accepted, interrupt processing starts from the next cycle of an instruction under execution at this point.

After execution of an instruction, under execution at accepting an interrupt, completes, an INTACK (Interrupt Acknowledge) sequence is executed and branch to the beginning of the interrupt service routine. The INTACK sequence operates as follows.

When an INTACK sequence starts, the contents of the program counter (PC) and the program bank register (PG) (indicates the address of the instruction code to be executed next) are saved in stack in the order of PG, PC_H (PC high-order byte), and PC_L (PC low-order byte). Then the contents of the processor status register (PS) are saved in stack in the order of PS_H (PS high-order byte) and PS_L (PS low-order byte), and the interrupt disable flag is set to "1". At the same time, the request bit of the accepted interrupt is cleared and the IPL in the processor status register is replaced by the interrupt priority level of the accepted interrupt. Then the vector address of the interrupt is stored in the program counter and PG becomes 00_{16}.

Note: IPL is set to the values shown in Table 2.6.6 when a reset, watchdog timer, or software interrupt occurs. This is useful when processing multiple interrupts. (See "2.6.8 Interrupt Service Routine".)

Table 2.6.6 Change in IPL when an Interrupt Occurs

Interrupt source	Change in processor interrupt leve
Reset	0 (000_2)
Watchdog timer	7 (111_2)
Zero divide	No change
BRK instruction	No change
Other interrupt	Priority level of the accepted interrupt

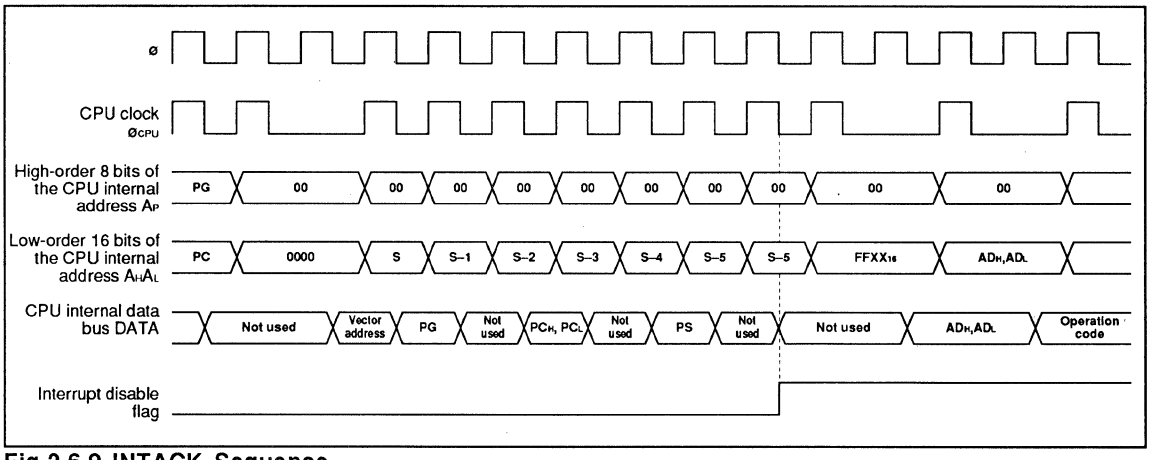


Fig.2.6.9 INTACK Sequence

The INTACK sequence is described below.

① Save the contents of PG and PC in stack before passing control to the interrupt service routine.
② Save the contents of the PS in stack just before passing control to the interrupt service routine
③ Set the interrupt disable flag to "1" to prohibit multiple interrupts.
④ Clear the request flag of the accepted interrupt.
⑤ Set the IPL to the priority level of the accepted interrupt.
(Useful when multiple interrupts are enabled.)
⑥ Change the contents of the PG and PC to branch to the interrupt service routine.
(Store 00_{16} in PG and the contents at the vector address for the interrupt in PC.)

2.6.8 Interrupt service routine

When control is passed to the interrupt service routine, the interrupt disable flag is set to "1" (interrupt is disabled). In addition, the interrupt request bit of the accepted interrupt is cleared. However, the request bit is retained if the interrupt was rejected by the interrupt priority level detection circuit. Furthermore, the IPL in the processor status register changes to the interrupt level of the accepted interrupt. This simplifies enabling of interrupts with higher interrupt level in the interrupt service routine (multiple interrupts). If multiple interrupts are allowed, the interrupt disable flag is cleared in the interrupt service routine. This enables accepting of higher priority interrupts as long as the IPL is not changed. Only the contents of the PC, PG, and PS are saved when control is passed to the interrupt service routine. Therefore, other necessary registers must be saved at the beginning of the interrupt service routine. The M37700 provides the **PSH** instruction to save all registers except the stack pointer with one instruction.

2.6.9 Returning from an interrupt service routine

A **RTI** instruction is used at the end of the interrupt service routine to return to the interrupted routine and continue processing. The **RTI** instruction restores the contents of the PG, PC, and PS saved before entering the interrupt service routine to their original registers. The other registers saved within the interrupt service routine must be restored with the **PUL** instruction before executing the **RTI** instruction. The request bit of other interrupts are retained after branching to the interrupt service routine. Therefore, if these interrupts are to be disabled after returning, these request bits must be cleared before executing the **RTI** instruction.

2.6.10 Interrupt response time

The time it takes an interrupt to be serviced after it has occurred is determined as follows:

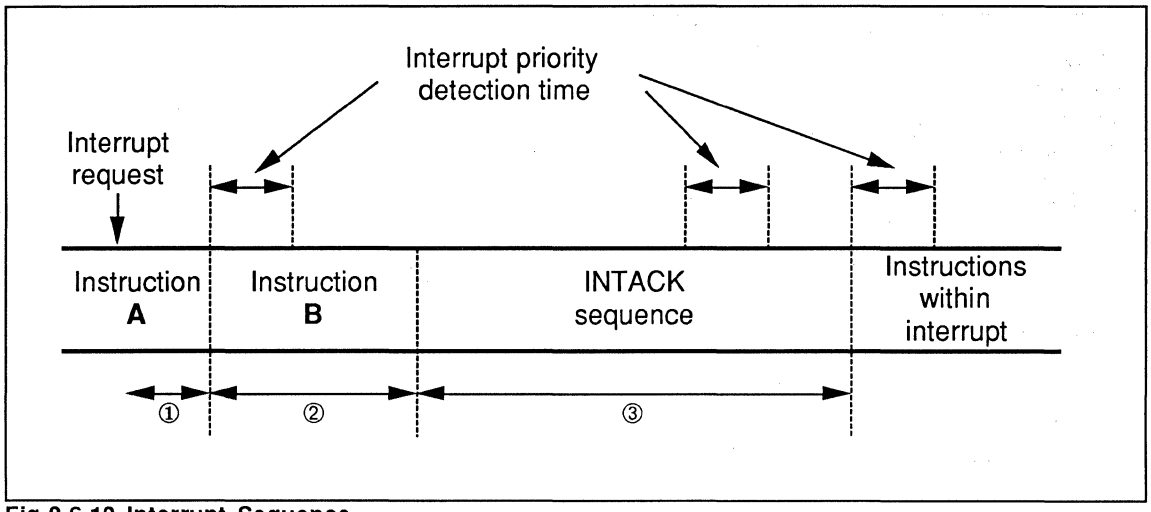


Fig.2.6.10 Interrupt Sequence

① Interval between interrupt occurrence and the end of instruction A that was interrupted.
② Interval between the start of the instruction B (interrupt priority detection start) and the end of instruction being executed when interrupt level detection has ended.
③ Time required for INTACK sequence such as saving registers and jumping to vector table address (13 cycles minimum).

Also note the following:
Interrupt priority detection is performed at the start of each instruction and during INTACK sequence. However, if the current instruction completes and the next instruction starts before detection completes, current detection is continued without starting detection for the next instruction.
The interrupt priority detection interval is selected with the processor mode register bits 4 and 5. The available intervals are two, four, or seven ø cycles.
Interrupts are not allowed while executing an instruction. Therefore, when using instructions that require some time to execute (such as **MVP**, **MVN**, and **RLA** instructions) at places where interrupts may occur, care must be taken the handling time.

Table 2.6.7 shows the interrupt response time for a certain instruction.

Table 2.6.7 Interrupt Response Time Example

		Minimum instruction	Maximum instruction		
			(A)	(B)	(C)
Time required for interrupt detection, priority detection, and single instruction execution.		2	917522	106	28
Time required to save the program counter, program bank register, and processor status register		13	15	15	15
General register save time	A	5	6	6	6
	A, B, X, Y	20	26	26	26
	A, B, X, Y, DPR, DT	23	31	31	31
Total (cycles)	A	19	917543	127	49
	A, B, X, Y	35	917563	147	69
	A, B, X, Y, DPR, DT	38	917568	152	74
Time (μs)	A	2.375	114692.875	15.875	6.125
	A, B, X, Y	4.375	114695.375	18.375	8.625
	A, B, X, Y, DPR, DT	4.750	114696.000	19.000	9.250

(A) **MVP** instruction (when 64K-byte data is transferred)
(B) **DIV** instruction (direct indirect long indexed Y)
(C) **AND** instruction (direct indirect long indexed Y)

2.7 Timer A

2.7.1 Timer A description

Timer A consists of five external output function timers TA0~TA4. These timers have identical functions (excluding two-phase pulse signal processing function) and are independent. There are four operating modes depending on the setting of the timer Ai (i=0~4) mode selection bit in the timer Ai mode register which is described later.

●Timer mode [00]*

This mode counts the selected internal clock and generates interrupt at an arbitrary frequency. Gate function (enable/disable count operation with the input level to the TAiIN pin) and polarity output function (output signal that changes phase each time the timer underflows from the TAiOUT pin) are available and can be selected by program.

●Event count mode [01]*

This mode counts the external clock input from the TAiIN pin. Whether to use it as an incremental counter or as a decremental counter can be selected internally or externally. An interrupt is generated at an arbitrary frequency. In addition, the pulse output function (output a signal that changes phase each time the counter underflows or overflows from the TAiOUT pin) can be selected by program. The two-phase pulse signal processing function can be selected for TA2, TA3, and TA4.

●One-shot pulse mode [10]*

In this mode, the timer is driven by an internal or external trigger and "H" level is output from the TAiOUT pin for an arbitrary interval.

●PWM (pulse width modulation) mode [11]*

In this mode, an arbitrary pulse width signal is output repeatedly from TAiOUT. PWM output is started by an internal or external trigger.

*The numbers in brackets are the contents of the timer Ai mode selection bit described later.

2.7.2 Block diagram

Figure 2.7.1 shows the block diagram of timer Ai. It is followed by the description of timer Ai related registers.

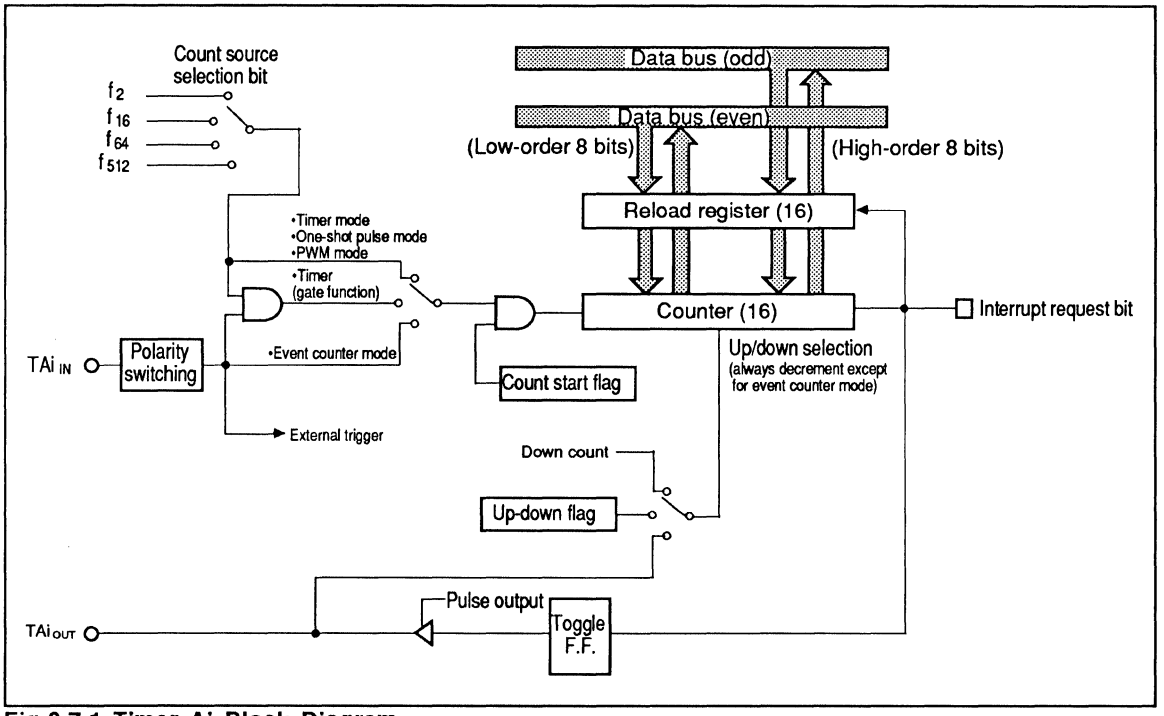


Fig.2.7.1 Timer Ai Block Diagram

(1)Counter and reload register

The counter and reload register consist of 16 bits. The counter counts the clock (count source) selected with the TAi mode register and its content is incremented (+1) or decremented (–1) each time a clock is input. The reload register is used to store the initial value of the counter.

Values are set in the counter with the timer Ai register (except PWM mode). The value written in the timer Ai register is also written in the counter and the reload register. Thereafter, the content of the counter changes each time a count clock (count source) is input, but not the content of the reload register.

(2)Count start flag

This register consists of flags used to start and stop each counter. The operation of each counter is controlled by the corresponding flag in this register. A count clock is input to the timer when this flag is set to "1" and disabled when it is set to "0". Each flag is automatically cleared and count is disabled when a value is set in the timer (a value is written in timer Ai register). (Except PWM mode.)

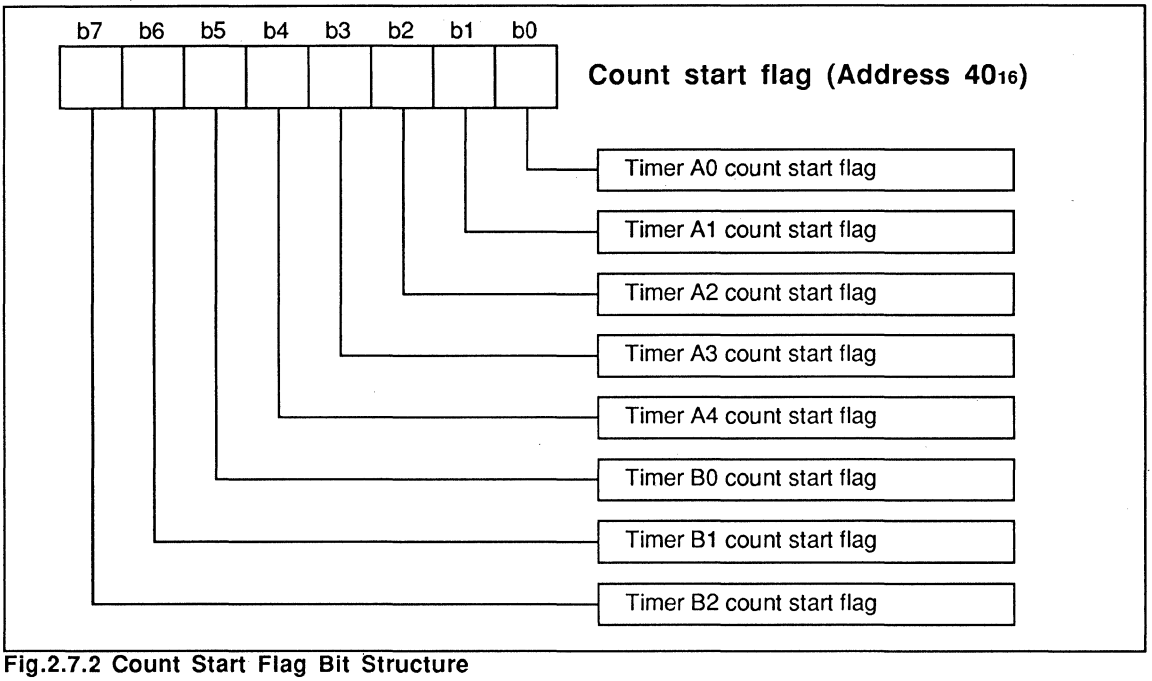


Fig.2.7.2 Count Start Flag Bit Structure

(3)One-shot start flag

This register consists of one-shot start flags that are used during one-shot pulse mode. A one-shot start internal trigger is generated by setting the bit corresponding to each timer to "1". This register consists of write only bits and the **LDM** or **STA** instructions must be used to write to it. (Do not use instructions such as **CLB** and **SEB** which perform read modify write.)

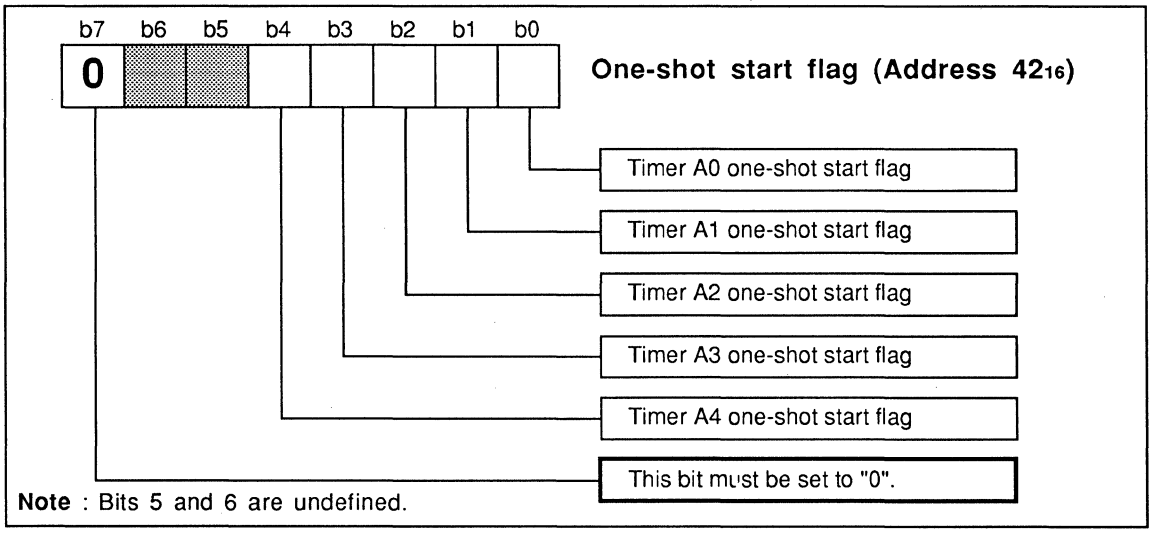


Fig.2.7.3 One-shot Start Flag Bit Structure

(4)Up-down flag

This register consists of up-down flags used during event count mode and two-phase pulse signal selection bits. Bits 7 to 5 are write only bits, but read modify write type instructions such as **CLB** and **SEB** can be used for this register.

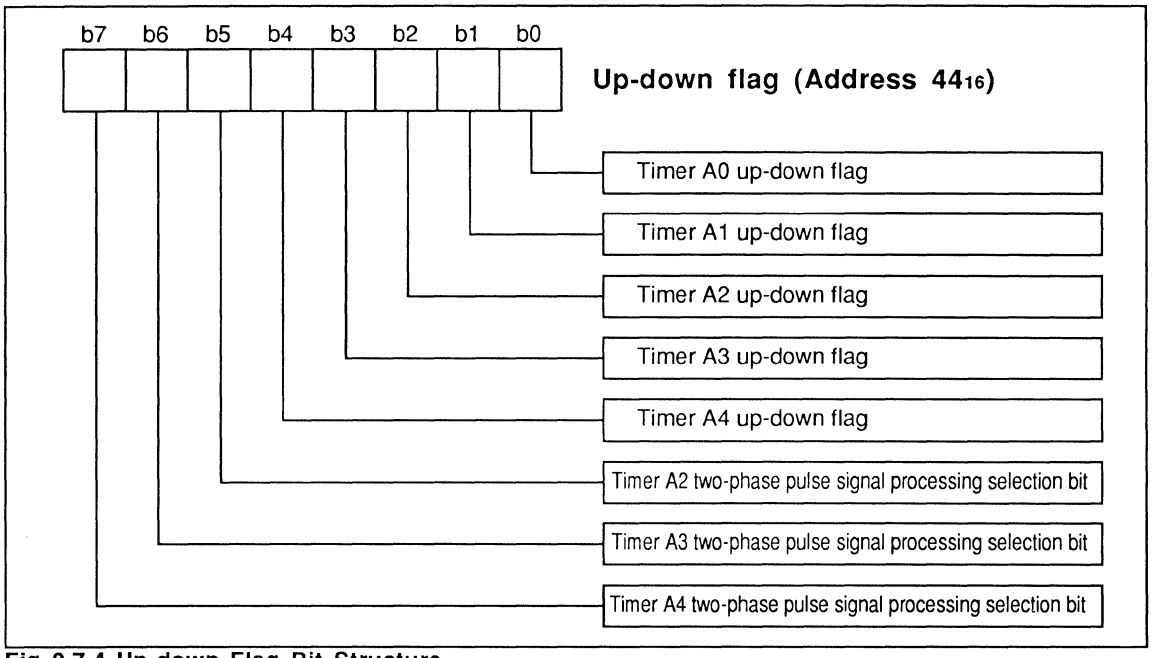


Fig 2.7.4 Up-down Flag Bit Structure

●Timer Ai up-down flags

These flags are valid during event count mode when the count up-down flag is selected as the increment/decrement trigger. A counter is decremented when this flag is "0" and incremented when it is "1".

●Two-phase signal processing selection bit

In event count mode, the counter can be controlled using two waveforms with their phases shifted by 90° (two-phase pulse signal processing function). This bit must be set to "0" when the two-phase pulse signal processing function is not used and in other modes.

(5)Timer Ai register

The data written in this register is stored in the counter and the reload register. Reading this register returns the content of the counter at that point.

The timer Ai register is divided in to high-order byte and low-order byte. Writing data and reading of halted timer can be performed in byte or word unit. However, the high-order and low-order bytes must be read simultaneously when the counter is operating.

Table 2.7.1 Timer Ai Register Address

Timer Ai register	High-order byte	Low-order byte
Timer A0 register	Address 47_{16}	Address 46_{16}
Timer A1 register	Address 49_{16}	Address 48_{16}
Timer A2 register	Address $4B_{16}$	Address $4A_{16}$
Timer A3 register	Address $4D_{16}$	Address $4C_{16}$
Timer A4 register	Address $4F_{16}$	Address $4E_{16}$

(6)Timer Ai mode register

The timer Ai mode registers control the timer operating modes and counter source and function selection. Bits 1 and 0 control the timer operating modes. Note that the meaning of each bit differs according to the timer operating mode. Refer to the description of the respective operating mode for the bit configuration in each operating mode.

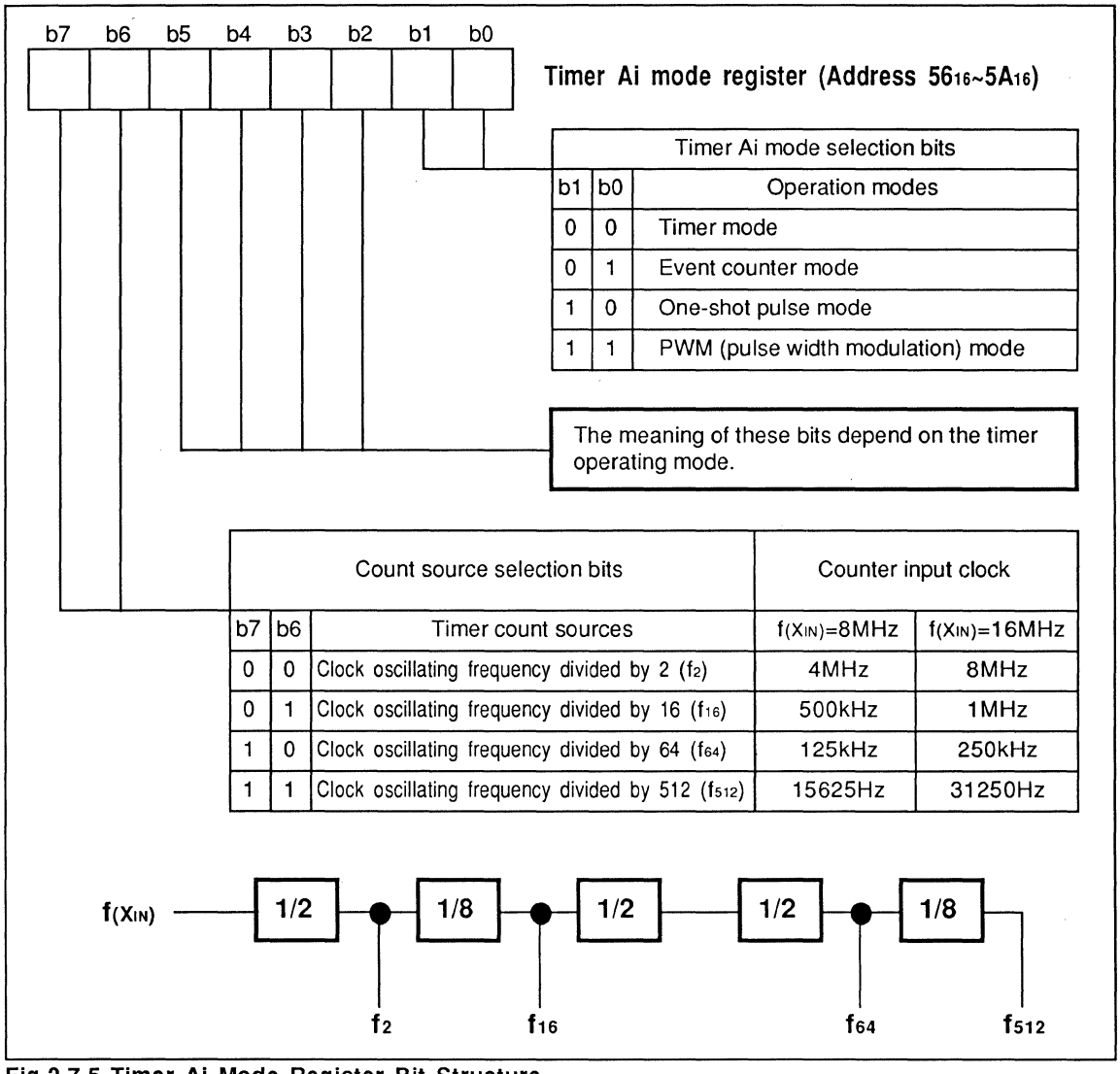


Fig.2.7.5 Timer Ai Mode Register Bit Structure

●Timer Ai mode selection bits

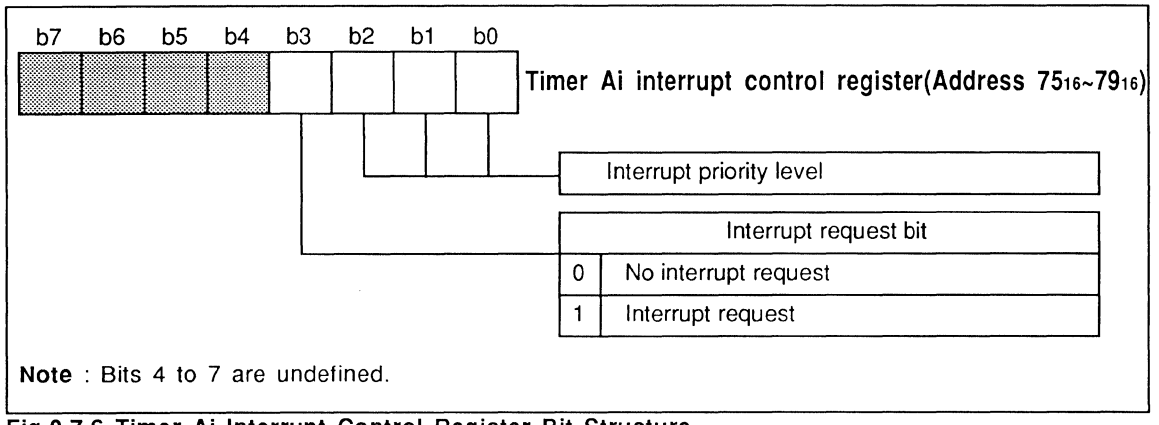
These bits are used to control the timer operating modes.

●Counter source selection bits

These bits are used to select the counter source (except in event counter mode).

(7)Timer Ai interrupt control register

The timer Ai interrupt control register consists of interrupt priority level selection bits and interrupt request bits.



Note : Bits 4 to 7 are undefined.

Fig.2.7.6 Timer Ai Interrupt Control Register Bit Structure

●Interrupt priority level selection bit

This bit is used to select the interrupt priority level. It should be set to a level between 1 and 7 when using a timer Ai interrupt. An interrupt is allowed only when this level is greater than the processor interrupt priority level (IPL) in the processor status register (PS). (When interrupt disable flag I is "0".) Set these bits to "000₂" to disable timer Ai interrupt.

Table 2.7.2 Interrupt Priority Level

Interrupt control register			Interrupt priority level	Priority
b2	b1	b0		
0	0	0	Level 0 (Interrupt disabled)	—
0	0	1	Level 1	Low
0	1	0	Level 2	↑
0	1	1	Level 3	↑
1	0	0	Level 4	↑
1	0	1	Level 5	↑
1	1	0	Level 6	↑
1	1	1	Level 7	High

●Interrupt request bit

This bit is set to "1" when a timer Ai interrupt request occurs. This bit can be set or cleared by program.

2.7.3 Timer mode [00]

A timer mode is selected by setting the timer Ai mode register bits 1 and 0 to "0". When this mode is selected, bit 5 of the timer Ai mode register must be set to "0".
Figure 2.7.7 shows the bit configuration of the timer Ai mode register in timer mode.

(1)Functions

In timer mode, the selected internal clock is decremented and an interrupt occurs when the counter underflows.

Timer dividing ratio ..... 1/(n+1)
n:Value set in timer Ai register
(a value between 0000_{16} and $FFFF_{16}$)

The following functions can be selected with the timer Ai mode register.

●Gate function

Controls count with input signal to timer Ai input pin A_{IIN}.

●Pulse output function

Outputs signal that changes polarity every time the content of the counter becomes "0000_{16}" from the timer Ai output pin TA_{iOUT}.

(2)Basic function

First the mode, count source, gate function, and pulse output function are selected with the timer Ai mode register bits. Then when a value n (between 0000_{16} and $FFFF_{16}$) is written in the timer Ai register, the count start flag is cleared (count disabled) and n is stored in the counter and the reload register.
When the count start flag is set to "1" (count enabled), the internal clock selected with the source selection bit is input to the counter. The content of the counter is decremented by 1 each time a clock is input. At the next clock input after the content of the counter reaches 0_{16}, the content of the reload register is loaded in the counter and the interrupt request bit is set to "1". Counting continues in this manner and the interrupt request bit is set to "1" each time the content of the counter changes from 0_{16} to n. Therefore, a timer Ai interrupt request occurs at every n+1 count of the clock input. The interrupt request bit remains set until the interrupt is accepted or it is cleared by program.
The content of the counter can be read at any time by reading the content of the timer Ai register, but the content of the reload register cannot be read.

Note : Interrupts must be enabled in order to use timer Ai interrupt. See "Section 2.6 Interrupts" for more information.

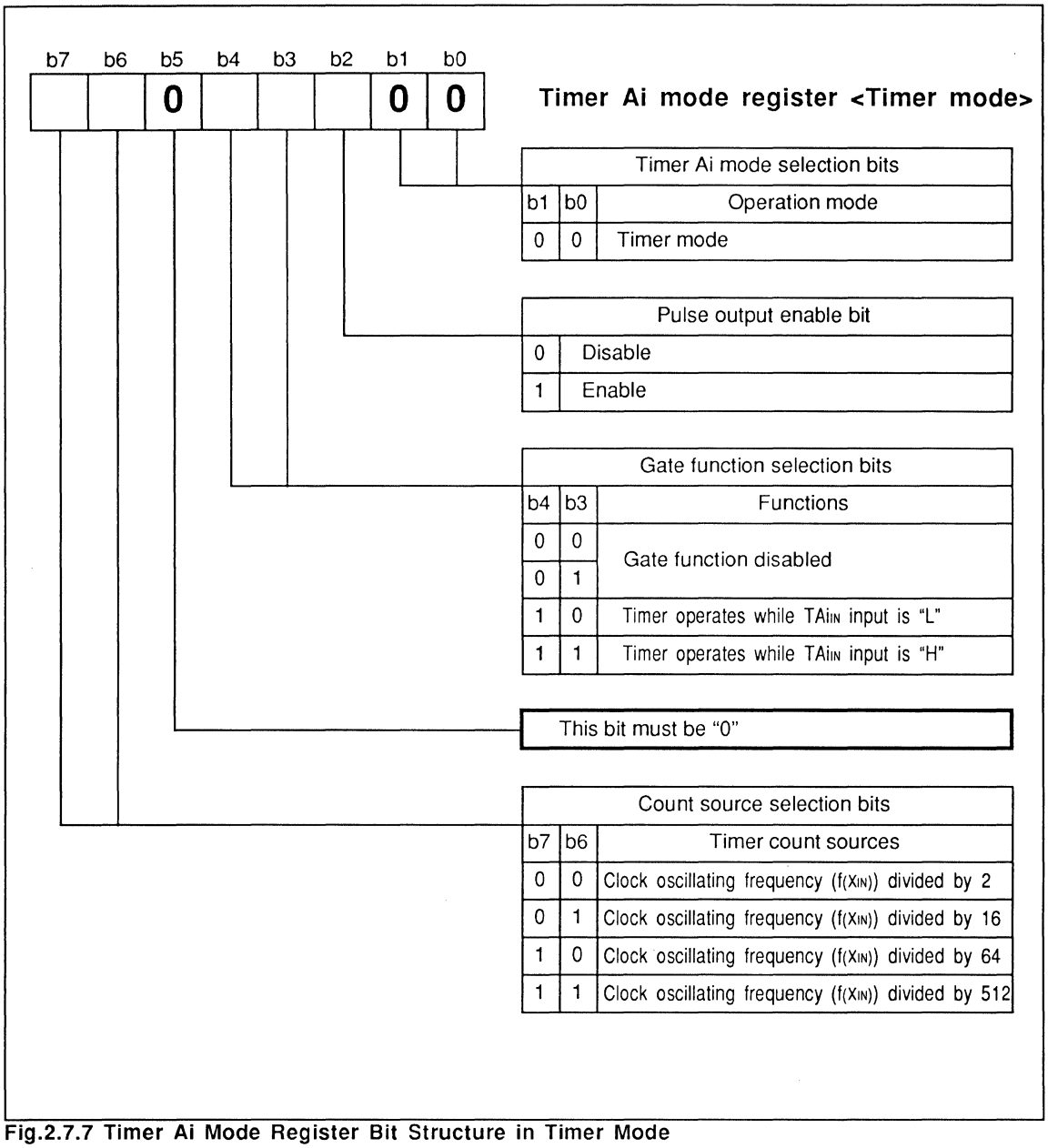


Fig.2.7.7 Timer Ai Mode Register Bit Structure in Timer Mode

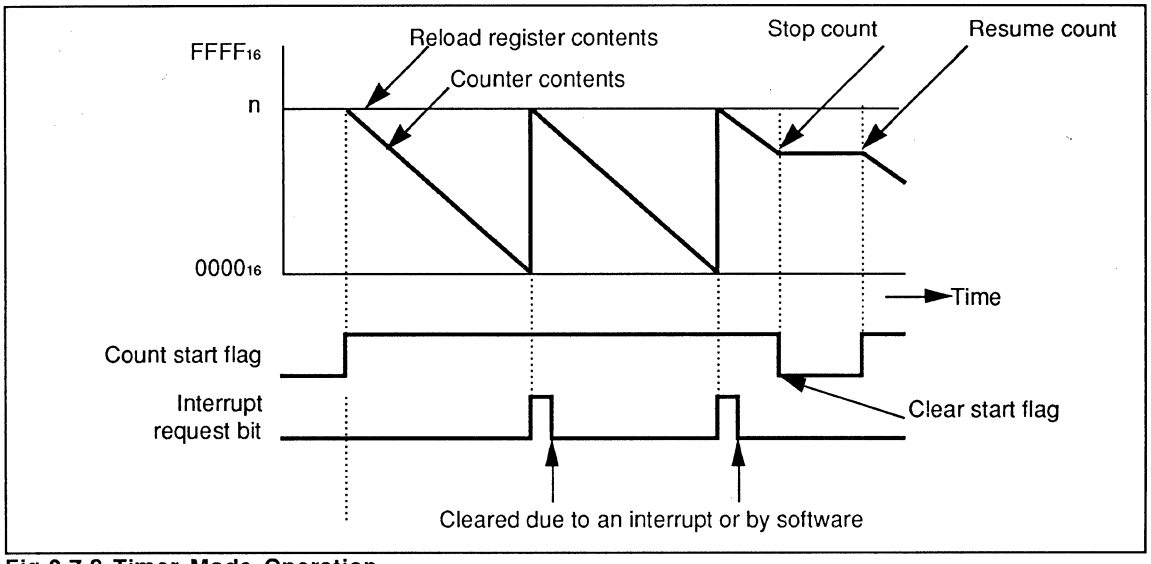


Fig.2.7.8 Timer Mode Operation

(3)Selection function

In timer mode, a gate function and a pulse output function can be selected by program. These functions can be used together.

●Gate function

When the gate function is enabled (timer Ai mode register bit 4 set to "1"), the starting and stopping of the timer count can be controlled by the level of the signal input to the TAiIN pin. The effective level is selected with the timer Ai mode register bit 3.

When the gate function is enabled, counting is performed only when the count start flag is "1" and the input to pin TAiIN is at the effective level. Counting stops if the input level is ineffectively. However, the content of the counter is preserved and counting can resume when the input level returns to an effective level.

Precautions when using the gate function

1.The TAiIN pin is in common with normal port pins. Therefore, the data direction register of the corresponding port must be set to input when using the gate function.

2.The pulse width of the TAiIN pin input signal during count interval and count halt interval must be at least 2 cycles of the timer count source.

●Pulse output function

When the pulse output function is enabled (timer Ai mode register bit 2 is set to "1"), a signal that changes polarity every time the content of the counter becomes "0000_{16}" is output from the TAi_{OUT} pin.

An "L" level is output from the TAi_{OUT} pin when the count start flag is "0" (count disabled).

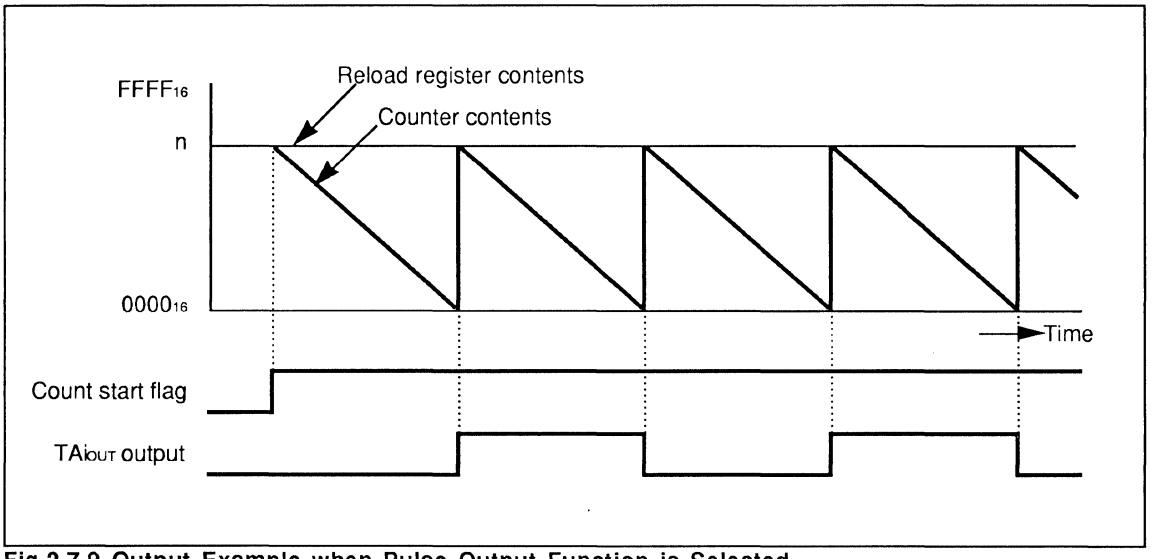


Fig.2.7.9 Output Example when Pulse Output Function is Selected

Precautions when using the pulse output function

1. When the content of the timer Ai register is changed while counting, the count start flag becomes "0" (count disabled) and the TAi_{OUT} pin level becomes "L".
2. The TAi_{OUT} pin is in common with normal port pins. When the pulse output function is enabled, the corresponding port is forced to output mode and functions as a timer output pin losing its programmable I/O port function. It can be used as a programmable I/O port once the pulse output function is disabled.

2.7.4 Event counter mode [01]

The event counter mode is selected by setting the timer Ai mode register bit 1 to "0" and bit 0 to "1". When this mode is selected, the timer Ai mode register bit 5 must be set to "0".
Figure 2.7.10 shows the bit configuration of the timer Ai mode register and the up-down flag during event counter mode.

(1)Functions

In event counter mode, the external clock input from the TAi_{IN} pin is counted and an interrupt occurs each time the counter overflows or underflows.
In event counter mode, the counter can be incremented or decremented. This selection is made with the contents of the up-down flag or the input signal to the TAi_{OUT} pin.

Timer dividing ratio Incremental counter.....1/(n+1)
Decremental counter.....$1/(FFFF_{16}-n+1)$
n: Value set in timer Ai register
(a value between 0000_{16} and $FFFF_{16}$)

In addition, in this mode, the pulse output function and two-phase pulse signal processing function can be selected from a program (two-phase pulse signal processing function is available only with timers A2, A3, and A4)

●Pulse output function

A pulse that changes phase every time the content of the counter becomes "0000_{16}" (decremental count) or "$FFFF_{16}$" (incremental count) is output from timer Ai output pin TAi_{OUT}.

●Two-phase pulse signal processing function

Whether to increment or decrement the counter can be selected using two signals with different phases. This function is selected with bits 5 to 7 of the up-down flag (address 44_{16}).

(2)Basic function

First the mode, up-down selection bit, effective edge of the count source, and whether to enable or disable pulse output function is selected with the timer Ai mode register. Then when, a value n ($n=0000_{16}$ to $FFFF_{16}$) is written in the timer Ai register, the count start flag is cleared to "0" (count disabled) and the value n is stored in the counter and the reload register.
Then, when the count start flag is set to "1" (count enabled), the external clock input to the TAi_{IN} pin is input to the counter. The counter counts at the falling (when the count polarity selection bit is "0") or rising (when the count polarity selection bit is "1") edge of the input clock.
Whether to increment or decrement the counter can be selected with the up-down flag or the level of the input signal to the TAi_{OUT} pin. The content of the up-down flag is used if the up-down switching factor selection bit (bit 4) in the timer Ai mode register is "0" and the TAi_{OUT} pin input signal level is used if it is "1".

●When using the content of the up-down flag

The counter is decremented when the bit corresponding to the timer Ai (bit i) in the up-down flag (address 44_{16}) is set to "0" and incremented when it is set to "1".

●When using the TAi_{OUT} input signal

The counter is decremented when the TAi_{OUT} pin input level is at "L" and incremented when it is at "H".

Precautions when externally controlling increment/decrement

1.The TAi_{OUT} pin is in common with normal port pins. Therefore, the data direction register of the corresponding port pin must be set to input when the input level of the TAi_{OUT} pin is used to control increment/decrement.
2.The pulse output function, which is described later, cannot be used when the input level to the TAi_{OUT} pin is used to control increment/decrement.

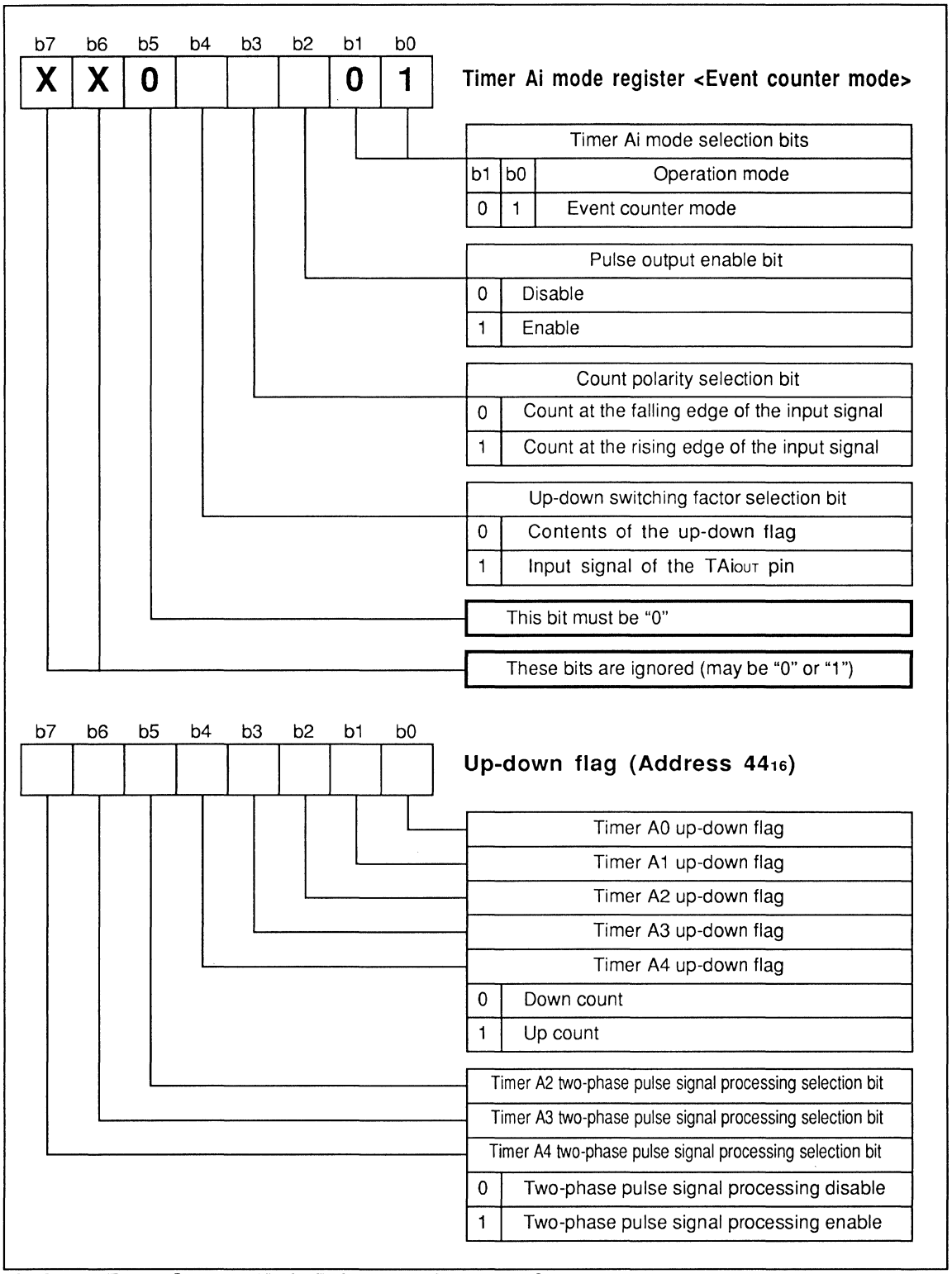


Fig 2.7.10 Event Counter Mode Related Registers Bit Structure

The count direction can be changed while counting. However, in this case, the increment interval and the decrement interval must be at least two cycles of the timer count source.
The content of the reload register (n) is loaded in the counter and an interrupt request bit is set to "1" at the next clock input when the content of the counter reaches "0000_{16}" (when decrementing) or "$FFFF_{16}$" (when incrementing). Counting continues and the interrupt request bit is repeatedly set at a certain interval. The interrupt request bit remains set until it is accepted or is cleared from a program.
The content of the counter can be read by reading the content of the timer Ai register, but the content of the reload register cannot be read.

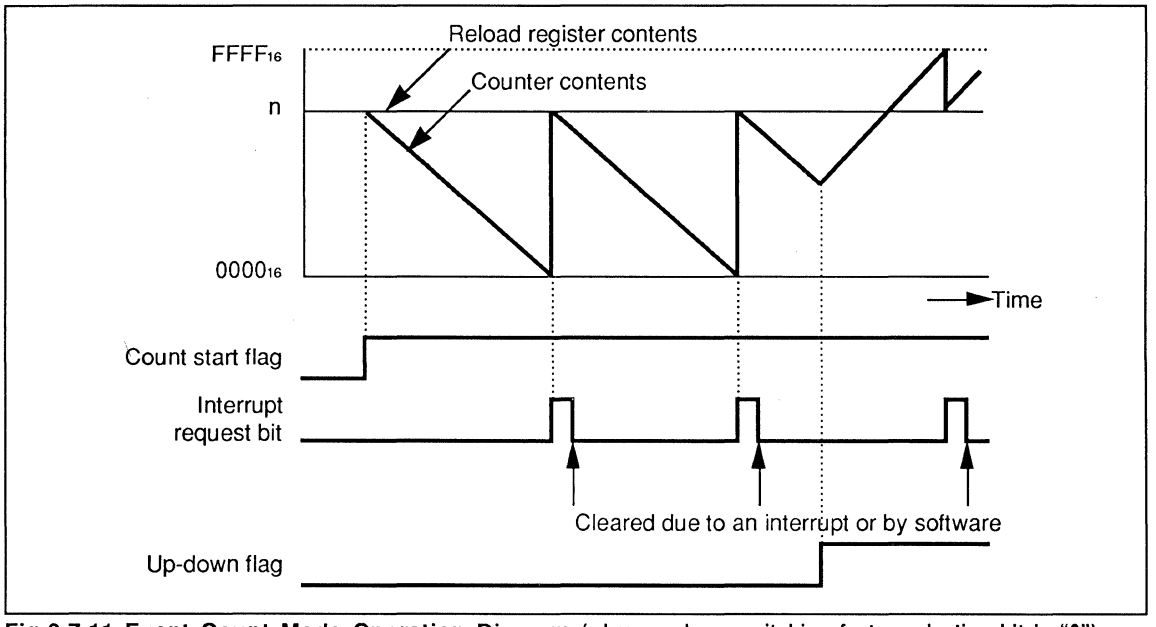


Fig.2.7.11 Event Count Mode Operation Diagram (when up-down switching factor selection bit is "0")

(3)Selection function

In event counter mode, a pulse output function and a two-phase pulse signal processing function can be selected by program. However, only timers A2, A3, and A4 can use the two-phase pulse signal processing function.

●Pulse output function

When the pulse output function is enabled (timer Ai mode register bit 2 is set to "1"), a signal that changes polarity every time the content of the counter becomes "0000_{16}" (decremental count) or "$FFFF_{16}$" (incremental count) is output from the TAi$_{OUT}$ pin.
An "L" level is output from the TAi$_{OUT}$ pin when the count start flag is "0" (count disabled).

Precautions when using the pulse output function

1.The counter up-down selection cannot be made externally because the pulse output function uses the TAi$_{OUT}$ pin.
2.When the content of the timer Ai register is written a value (if the same value) while counting, the count start flag becomes "0" (count disabled) and the TAi$_{OUT}$ pin level becomes "L".
3.The TAi$_{OUT}$ pin is in common with normal port pins. When the pulse output function is enabled, the corresponding port is forced to output mode and functions as a timer output pin losing its programmable I/O port function. It can be used as a programmable I/O port once the pulse output function is disabled.

●Two-phase pulse signal processing function

Timers A2 to A4, for which the event counter mode is selected, can use the two-phase pulse signal processing function which controls the counter increment/decrement with two input pulses shifted by 90°.

When using the two-phase pulse signal processing function, the high-order three bits of the up-down flag which is used timer (timer A2, A3, or A4) must be set to "1". And the timer Aj mode register (j=2, 3, 4) must be set as follows:

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

Timer Aj mode register (j=2, 3, 4)

("X" may be "0" or "1")

Fig.2.7.12 Setting value at Using Two-phase Pulse Signal Processing Function

After setting the timer Aj mode register, when a value n (n=0000_{16} to $FFFF_{16}$) is written in the timer Aj register, the count start flag is cleared to "0" and the value n is loaded in the counter and reload register. When using the two-phase pulse signal processing function, the reference pulse must be input to the TAjOUT pin and a pulse shifted by 90° from the reference pulse must be input to the TAjIN pin.

The counter is enabled when the count start flag is set to "1". After the input signal to the TAjOUT pin changes from "L" to "H", the counter is incremented when a rising edge is input to the TAjIN pin and is decremented when a falling edge is input.

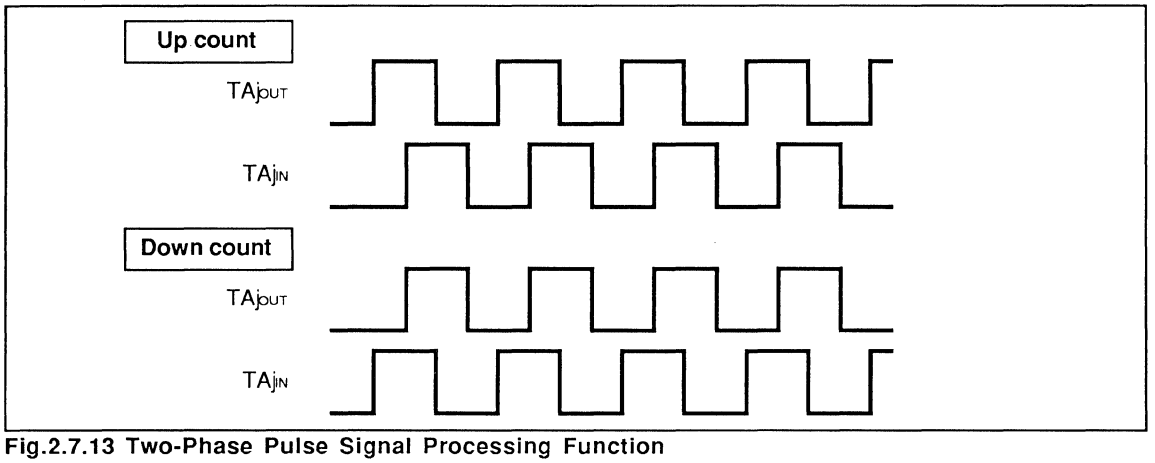


Fig.2.7.13 Two-Phase Pulse Signal Processing Function

Precautions when selecting two-phase signal processing function

1. When using the two-phase pulse signal processing function, the pulse output function cannot be used because the TAjOUT pin is used as the reference pulse input pin.
2. The phase difference between the reference pulse and the pulse input to the TAjIN pin must be between 80° and 100°.
3. If the input pulse changes direction as shown below, the following error occurs in the count value.

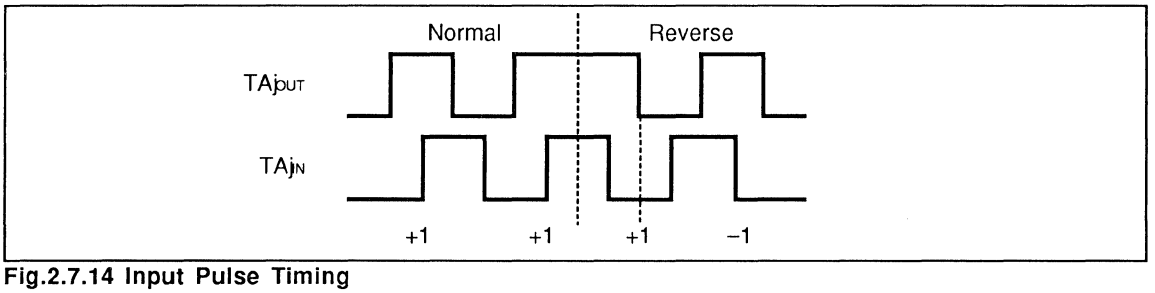


Fig.2.7.14 Input Pulse Timing

2.7.5 One-shot pulse mode [10]

The one-shot pulse mode is selected by setting the timer Ai mode register bit 1 to "1" and bit 0 to "0". When this mode is selected, the timer Ai mode register bit 5 must be set to "0" and bit 2 must be set to "1".

Figure 2.7.15 shows the bit configurations of the timer Ai mode register and one-shot start flag during one-shot pulse mode.

(1)Functions

In one-shot pulse mode, the level of the timer Ai output pin (TAi_{OUT}) is held at "H" for an arbitrary interval after a trigger.

The trigger can be either internal or external. The source of trigger is selected using bit 4 of the timer Ai mode register. A software trigger (internal trigger) is selected when this bit is "0" and a TAi_{IN} pin input signal (external signal) is selected when this bit is "1".

●Software trigger

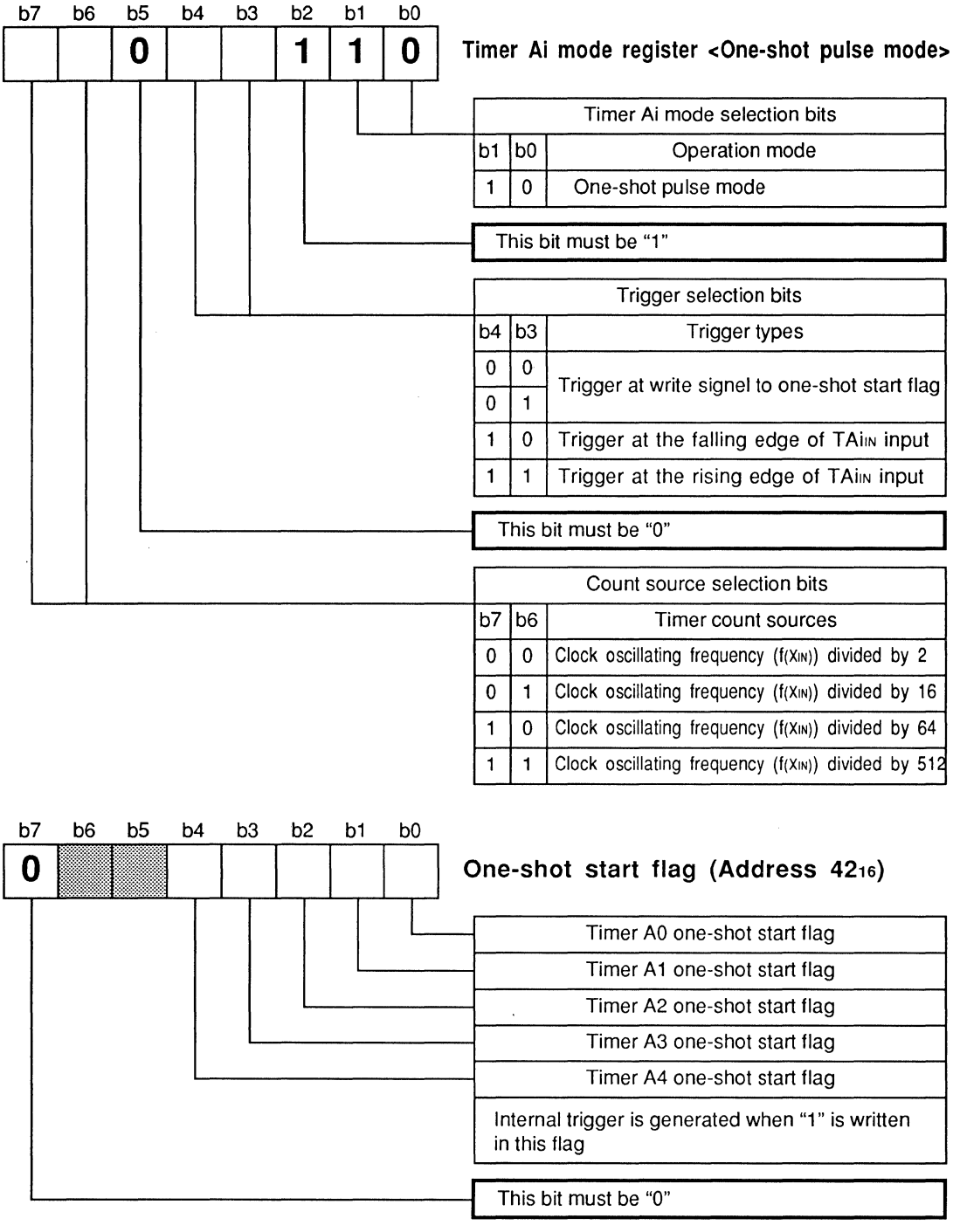
Internal trigger is generated by setting the bit corresponding to each timer in the one-shot start flag (address 42_{16}) to "1". Use **LDM** or **STA** instruction to write to the one-shot start flag (do not use read-modify-write type instructions such as **SEB**).

●External trigger (TAi_{IN} pin input pulse)

An input signal from the TAi_{IN} pin is used as the trigger. Whether to trigger at the rising edge or falling edge of the input signal is selected with bit 3 of the timer Ai mode register. A trigger occurs at the falling edge when this bit is "0" and at the rising edge when it is "1".

Precautions when using an external trigger

The TAi_{IN} pin is in common with normal port pins. Therefore, the data direction register of the corresponding port must be set to input when using an external trigger.



Note: Use **LDM** and **STA** instructions to write in the one-shot start flag (do not use the **SEB** instruction).

Fig.2.7.15 One-shot Pulse Mode Related Register Bit Structure

(2) One-shot pulse mode operation

First the mode, count source, and trigger source are selected with the timer Ai mode register bits. Next, when a value n (between 0000_{16} and $FFFF_{16}$) is written in the timer Ai register, the count start flag is cleared (count disabled) and n is stored in the counter and the reload register. Then count is enabled when the count start flag is set to "1", but the counter does not start until it is triggered. If bit 4 of the timer Ai mode register is "0", an internal trigger is generated by setting the bit corresponding to each timer in the one-shot start flag to "1". If bit 4 is "1", a trigger is generated at the rising edge or falling edge of the TAi_{IN} pin input signal (whether to trigger at the rising edge or falling edge is selected with bit 3 of the timer Ai mode register).

When triggered, the TAi_{OUT} pin level becomes "H" and the timer starts counting (however, if the timer Ai register contains "0000_{16}", the TAi_{OUT} pin level remains at "L" and counting does not start). The counter is decremented and when its content reaches "0001_{16}", the TAi_{OUT} pin level becomes "L", the content of the reload register is loaded in the counter, and counting stops. An interrupt occurs and the interrupt request bit is set to "1" when the TAi_{OUT} pin level changes from "H" to "L". The interrupt request bit remains set until the interrupt is accepted or it is cleared by program.

The count resumes at the next trigger and this operation is repeated.

The "H" width of the pulse output from the TAi_{OUT} pin is (count source cycle) × n.

If the count start flag is "0" (count disabled), the TAi_{OUT} pin output is at "L" level. Therefore, an arbitrary pulse width can be generated by setting a value in the timer Ai register before setting the timer Ai count start flag to "1".

If another trigger is received before a triggered operation completes, the content of the reload register is transferred to the counter and decrement continues from that value. In this case, the TAi_{OUT} pin level becomes "L" at n+1 count after the trigger. A trigger never causes the content of the reload register to be transferred to the counter except when it is received while a triggered operation is being executed. In this case, there should be at least one cycle of the timer count source between triggers.

Precautions when using one-shot pulse mode

If the low-order eight bits of the timer Ai register is set to "00_{16}" in one-shot pulse mode, the value must be reset and the count start flag must be set to "1" before the next trigger after a one-shot pulse output.

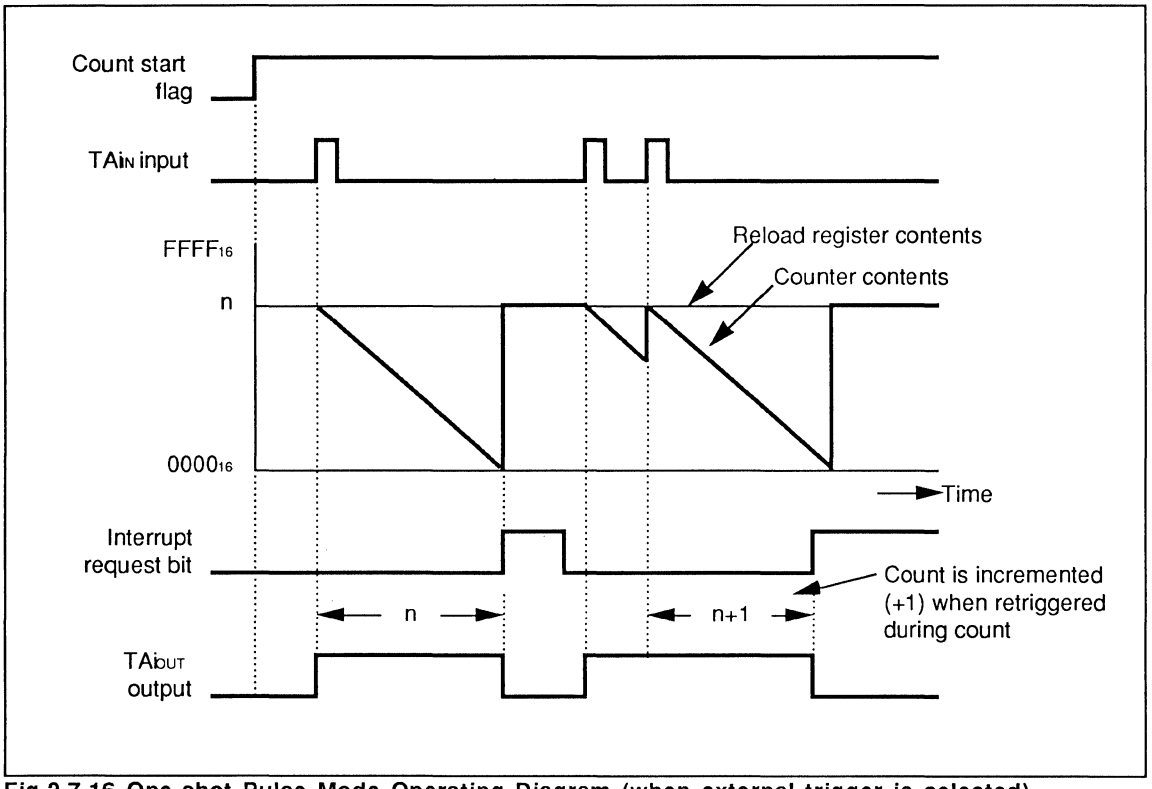


Fig.2.7.16 One-shot Pulse Mode Operating Diagram (when external trigger is selected)

2.7.6 PWM (Pulse Width Modulation) mode [11]

The PWM mode is selected by setting bits 1 and 0 of the timer Ai mode register to "1". When this mode is selected, bit 2 of the timer Ai mode register must be set to "1". Figure 2.7.17 shows the bit configuration of the PWM mode related registers.

(1)Functions

The PWM mode continuously outputs an arbitrary pulse width signal from the TAi$_{OUT}$ pin. A 16-bit PWM mode or an 8-bit PWM mode can be selected from a program.

●16-bit PWM mode

The counter functions as a 16-bit pulse width modulator.

●8-bit PWM mode

The reload register and the counter are both divided into 8-bit halves. The high-order 8-bits of the counter function as a pulse width modulator and the low-order 8-bits function as a prescaler.

(2) Operation

First the mode, count source, and trigger source are selected with the timer Ai mode register bits. Next, when data is written in the timer Ai register with the timer Ai start flag set to "0" (pulse width modulator operation halted), it is stored in the counter and the reload register. In PWM mode, the count start flag is not cleared when writing to the timer Ai register.
The trigger can be either internal or external. The source of trigger is selected with bit 4 of the timer Ai mode register. A software trigger is selected when this bit is "0" (internal trigger) and an external trigger is selected when it is "1".

●Software trigger

An internal trigger is generated by setting the count start flag to "1".

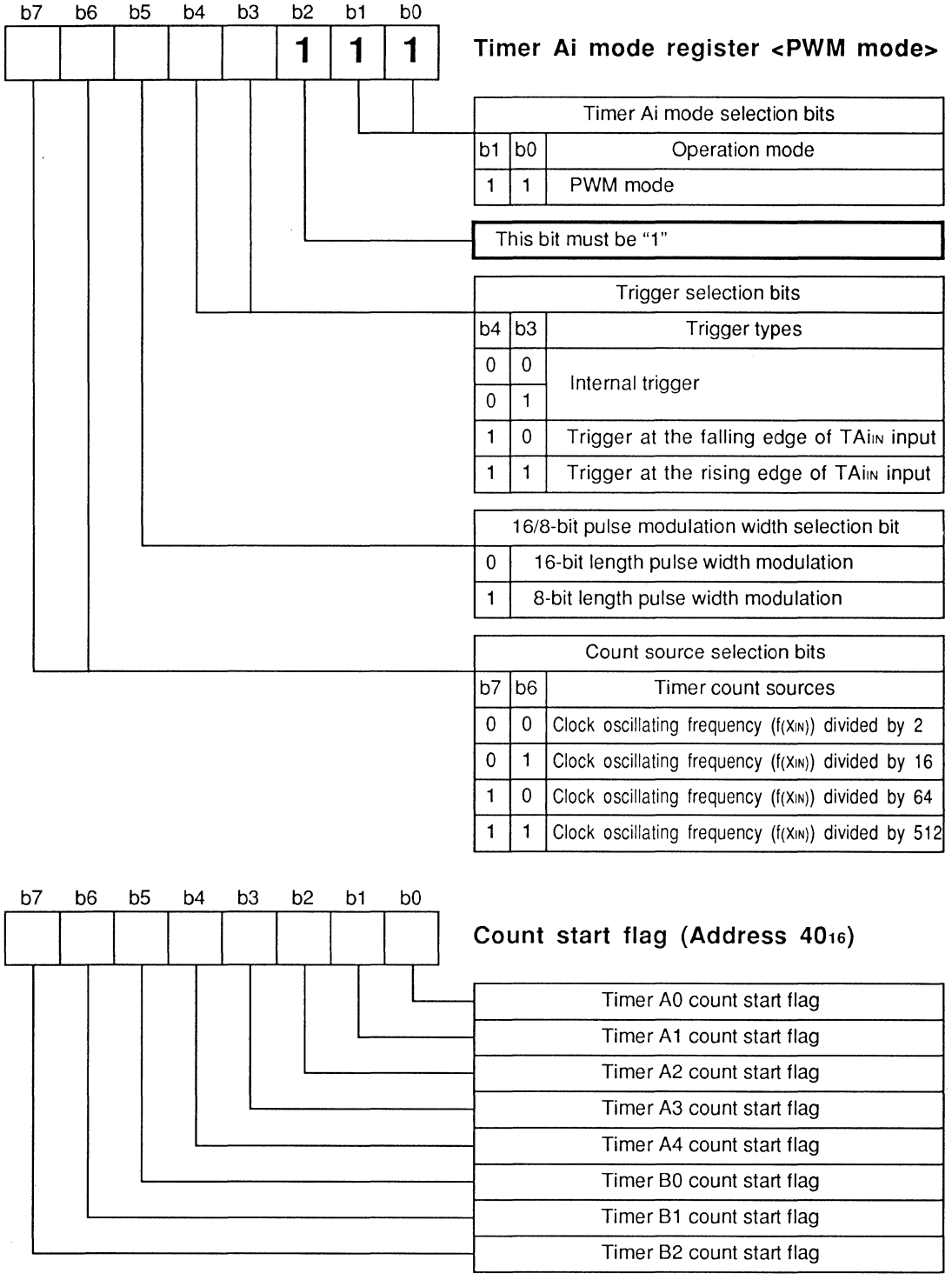
●External trigger (TAi$_{IN}$ pin input pulse)

After the count start flag is set to "1", a trigger is generated at the effective edge of the input signal to the TAi$_{IN}$ pin. The effective edge is selected with bit 3 of the timer Ai mode register. A trigger is generated at the falling edge if bit 3 is "0" and at the rising edge if it is "1".

Precautions when using an external trigger

The TAi$_{IN}$ pin is in common with normal port pins. Therefore, the data direction register of the corresponding port must be set to input when using an external trigger.

The pulse width modulator starts when triggered and outputs an arbitrary pulse from the TAi$_{OUT}$ pin. The pulse width modulator cannot be retriggered once it is started.



Note : In PWM mode, an internal trigger is generated by writing "1" in the count start flag. (when internal trigger is selected)

Fig.2.7.17 Pulse Width Modulation Mode Timer Ai Mode Register Bit Configuration

(3)Selection function

A 16-bit or an 8-bit pulse width modulation mode can be selected from a program. A 16-bit PWM mode is selected when bit 5 of the timer Ai mode register is "0" and an 8-bit PWM mode is selected when it is "1".

●16-bit PWM mode

In 16-bit PWM mode, the cycle and width of the pulse output from the TAiOUT pin are as follows:

Output pulse cycle ..... $(1/f_i) \times (2^{16}-1)$ [s]
Output pulse "H" width ..... $(1/f_i) \times m$ [s]
where
f_i: Clock frequency [Hz]
m: Value in the reload register

Figure 2.7.18 shows the output waveform during 16-bit PWM mode. An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set at each falling edge of the output pulse. To change the pulse width while it is being output, data must be written in the timer Ai register while the output pulse is at "L" (this also applies to 8-bit PWM mode). This data is written only in the reload register. The counter and count start flag are unaffected. The content of the reload register is transferred to the counter just before the rising edge of the next pulse and the output pulse is updated at the next pulse cycle.
The content of the reload register is obtained if the timer Ai register is read during PWM mode.

●8-bit PWM mode

When 8-bit PWM mode is selected, both the reload register and the counter are divided into 8-bit halves. The low-order 8 bits of the counter function as a prescaler and the high-order 8 bits function as an 8-bit pulse width modulator.
The prescaler counts the clock selected with bits 6 and 7 of the timer Ai mode register and a pulse is generated when its content reaches "00_{16}". At the same time, the content of the reload register is written in the prescaler and count continues. The counter counts the pulse generated by the prescaler. Therefore, the cycle and width of the pulse output from the TAiOUT pin are as follows:

Output pulse cycle ..... $(1/f_i) \times (n+1) \times (2^8-1)$ [s]
Output pulse "H" width ..... $(1/f_i) \times (n+1) \times m$ [s]
where
f_i: Clock frequency [Hz]
m: Value in the high-order 8 bits of the reload register
n: Value in the low-order 8 bits of the reload register

Figure 2.7.19 shows the output waveform during 8-bit PWM mode. The operation in this mode is similar to 16-bit PWM mode except that the bit length is 8-bits.

Precautions when using 8-bit PWM mode

In 8-bit PWM mode, if pulse output from the TAiOUT pin is started with a trigger, the pulse output starts after an "L" level is output for the specified "H" pulse width.

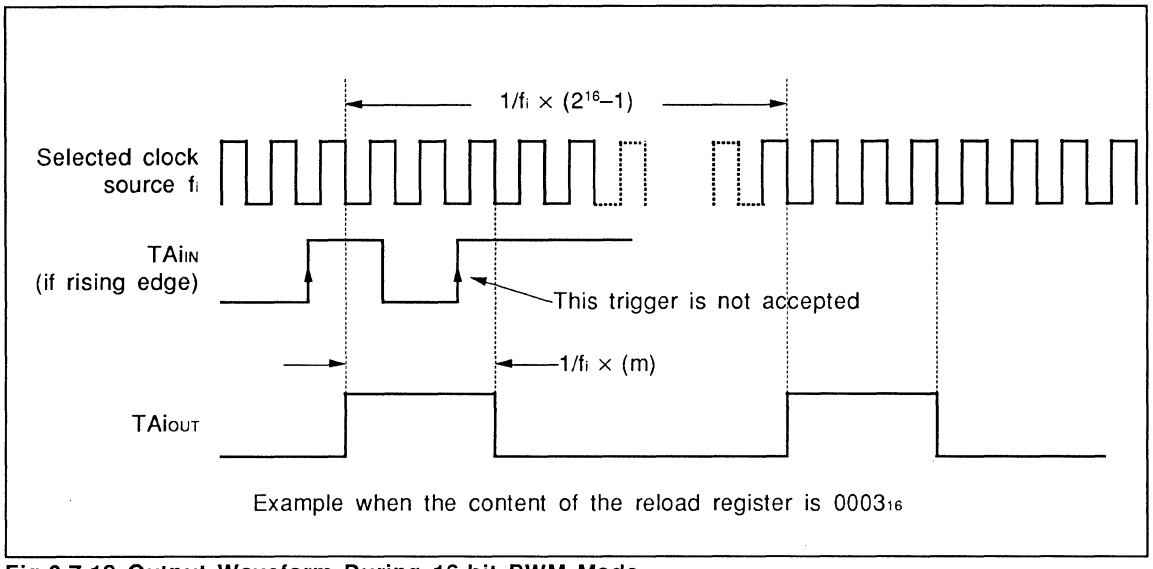


Fig.2.7.18 Output Waveform During 16-bit PWM Mode

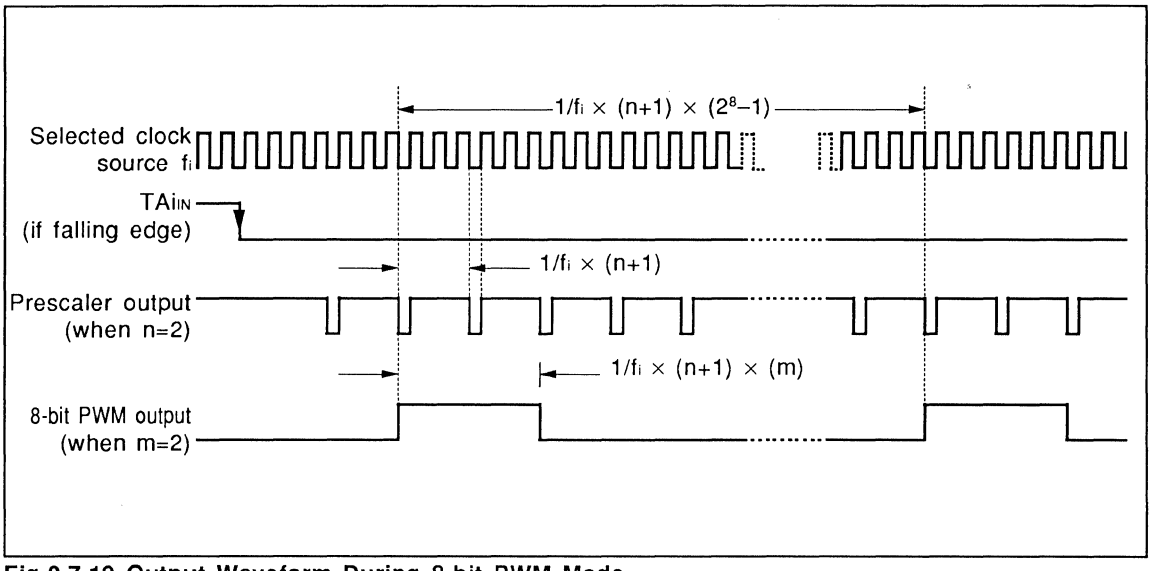


Fig.2.7.19 Output Waveform During 8-bit PWM Mode

2.8 Timer B

2.8.1 Timer B description

Timer B consists of three independent identical function timers TB0, TB1, and TB2. Three operation modes can be selected with the timer Bi mode selection bits of the timer Bi mode register (i=0, 1, 2).

●Timer mode [00]*

In this mode the selected internal clock is counted and an interrupt request is generated at an arbitrary frequency.

●Event counter mode [01]*

In this mode the external clock input to the TBi_{IN} pin is counted. The counter is decremented and an interrupt occurs when the timer underflows.

●Pulse cycle /pulse width measurement mode [10]*

In this mode, the frequency or the pulse width of the signal input to the TBi_{IN} pin is measured.

* The numbers in bracket indicate the timer Bi mode selection bits which are described later.

2.8.2 Block diagram

Figure 2.8.1 shows a block diagram of the timer Bi and the related registers.

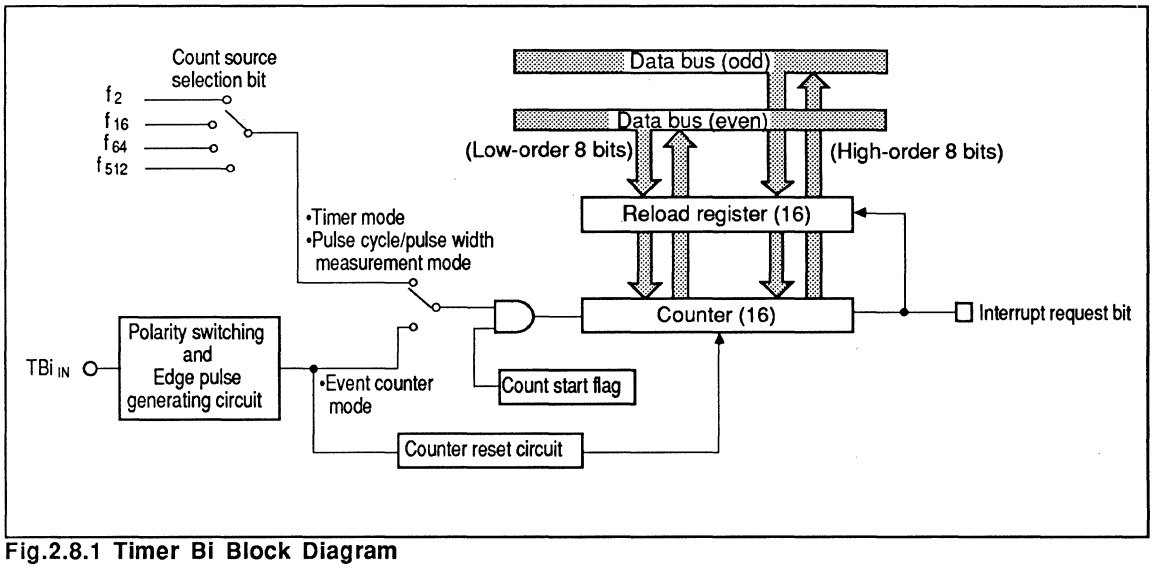


Fig.2.8.1 Timer Bi Block Diagram

(1)Counter and reload register

The counter and reload register consists of 16 bits. The counter counts the clock (count source) selected by the TBi mode register. It is decremented each time a clock is input. The reload register contains the initial value of the counter.

A value is set in the counter through the timer Bi register. The value written in the timer Bi register is written in the counter and the reload register. The content of the counter changes with clock input, but the content of the reload register remains unchanged.

(2)Count start flag

The count start flag consists of flags that starts and stops individual timers. Each bit controls the count operation of the corresponding timer. The count clock is input to the counter when this bit is "1" and count clock is inhibited when this bit is "0". These bits are cleared and the count is halted when a value is set in the timer (a value is written in the timer Bi register).

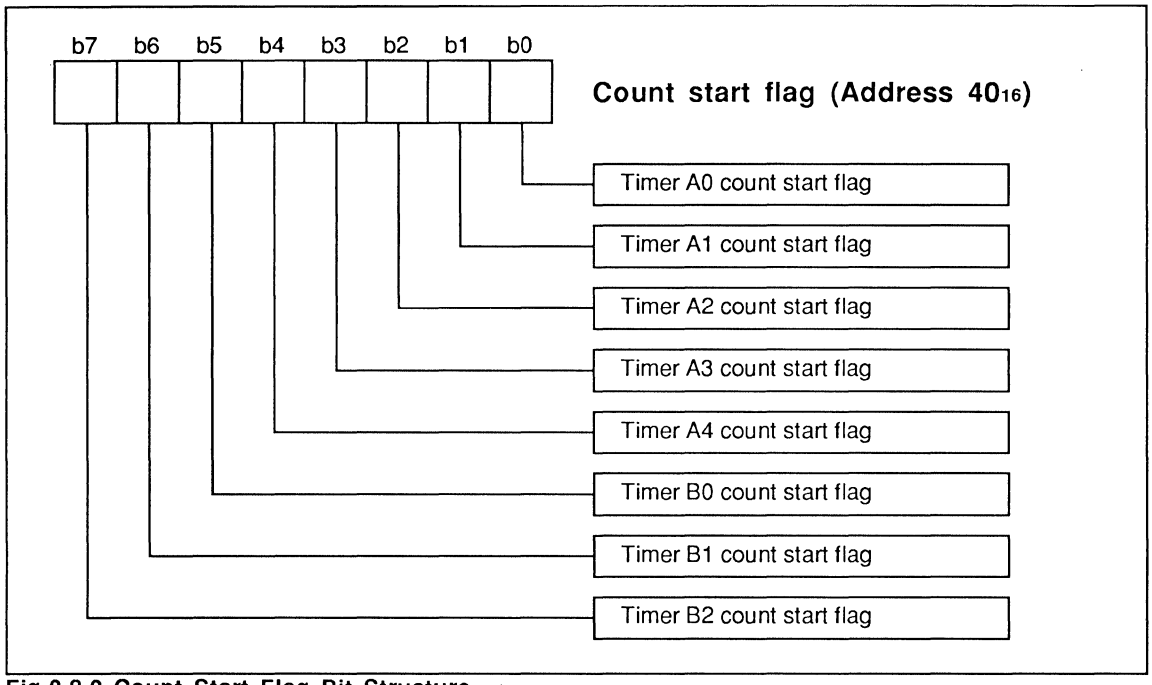


Fig.2.8.2 Count Start Flag Bit Structure

(3)Timer Bi register

The data written in this register is set in the counter and the reload register. The current value of the counter can be determined by reading this register.

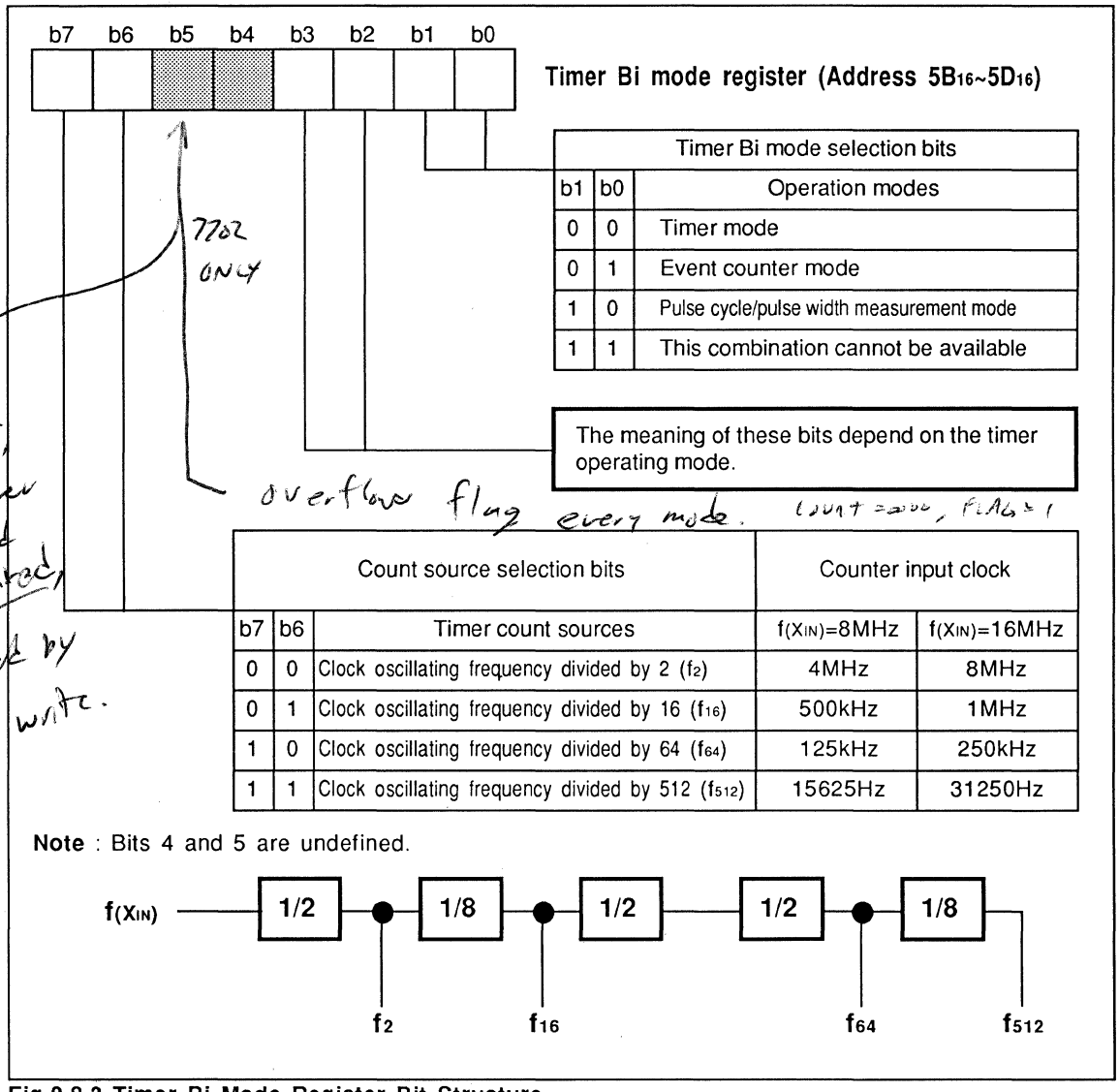
The timer Bi register is divided into high-order byte and low-order byte and data read while the counter is halted or data write can be performed either in byte or word unit. However, the Bi register must be read in word unit when the counter is operating.

Table 2.8.1 Timer Bi Register Address

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

(4)Timer Bi mode register

The timer Bi mode registers control the timer operating modes and counter source selection. Bits 1 and 0 control the timer operating modes. Note that the meaning of each bit differs according to the timer operating mode. Refer to the description of the respective operating mode for the bit configuration in each operating mode.



Timer Bi mode selection bits		
b1	b0	Operation modes
0	0	Timer mode
0	1	Event counter mode
1	0	Pulse cycle/pulse width measurement mode
1	1	This combination cannot be available

Count source selection bits			Counter input clock	
b7	b6	Timer count sources	f(XIN)=8MHz	f(XIN)=16MHz
0	0	Clock oscillating frequency divided by 2 (f_2)	4MHz	8MHz
0	1	Clock oscillating frequency divided by 16 (f_{16})	500kHz	1MHz
1	0	Clock oscillating frequency divided by 64 (f_{64})	125kHz	250kHz
1	1	Clock oscillating frequency divided by 512 (f_{512})	15625Hz	31250Hz

Fig.2.8.3 Timer Bi Mode Register Bit Structure

●Timer Bi mode selection bits

These bits are used to control the timer operating modes.

●Count source selection bits

These bits are used to select the counter source (except in event counter mode).

(5)Timer Bi interrupt control register

The timer Bi interrupt control register consists of interrupt priority level selection bits and interrupt request bits.

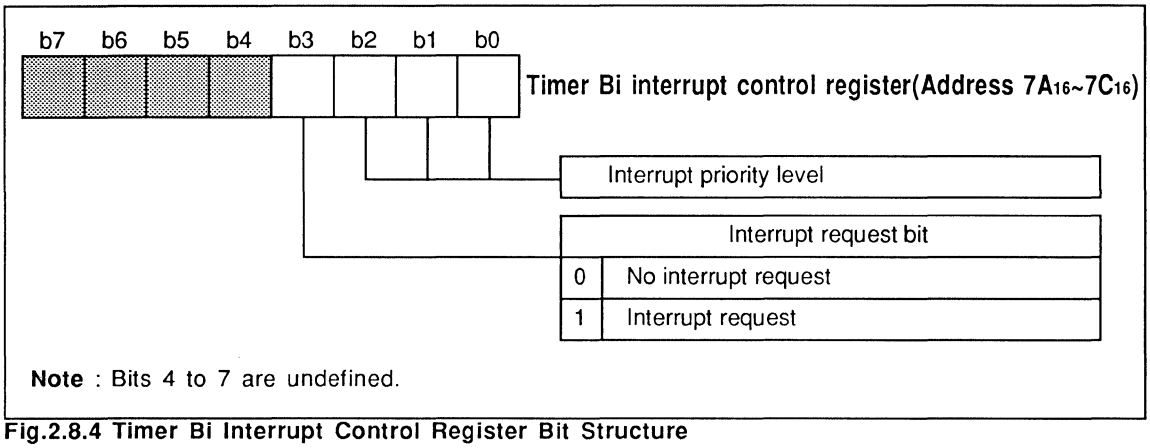


Fig.2.8.4 Timer Bi Interrupt Control Register Bit Structure

●Interrupt priority level selection bit

This bit is used to select the interrupt priority level. It should be set to a level between 1 and 7 when using timer Bi interrupt. An interrupt is allowed only when this level is greater than the processor interrupt priority level (IPL) in the processor status register (PS). (When interrupt disable flag I is "0".) Set these bits to "000_2" to disable the timer Bi interrupt.

Table 2.8.2 Interrupt Priority Level

Interrupt control register			Interrupt priority level	Priority
b2	b1	b0		
0	0	0	Level 0 (Interrupt disabled)	—
0	0	1	Level 1	Low
0	1	0	Level 2	↑
0	1	1	Level 3	↕
1	0	0	Level 4	↕
1	0	1	Level 5	↕
1	1	0	Level 6	↕
1	1	1	Level 7	High

●Interrupt request bit

This bit is set to "1" when a timer Bi interrupt request occurs. This bit can be set or cleared from a program.

2.8.3 Timer mode [00]

A timer mode is selected by setting timer Bi mode register bits 1 and 0 to "0". When this mode is selected, bits 2 and 3 of the timer Bi mode register are ignored.
Figure 2.8.5 shows the bit configuration of the timer Bi mode register in timer mode.

(1)Functions

In timer mode, the selected internal clock is decremented and an interrupt occurs when the counter underflows.

Timer dividing ratio ..... 1/(n+1)
n: Value set in timer Bi register
(a value between 0000_{16} and $FFFF_{16}$)

(2)Timer mode operation

First, the mode and count source are selected with the timer Bi mode register. Then when a value n (n=0000_{16} to $FFFF_{16}$) is written in the timer Bi register, the count start flag is cleared (count disabled) and n is stored in the counter and the reload register.
When the count start flag is set to "1" (count enabled), the internal clock selected with the source selection bit is input to the counter. The content of the counter is decremented by 1 each time a clock is input. At the next clock input after the content of the counter reaches 0_{16}, the content of the reload register is loaded in the counter and the interrupt request bit is set to "1". Count operation continues in this manner and the interrupt request bit is set to "1" each time the content of the counter changes from 0_{16} to n. Therefore, a timer Bi interrupt request occurs at every n+1 count of the clock input. The interrupt request bit remains set until the interrupt is accepted or it is cleared from a program.
The content of the counter can be read at any time by reading the content of the timer Bi register, but the content of the reload register cannot be read.

Note : Interrupts must be enabled in order to use timer Bi interrupt. See "Section 2.6 Interrupts" for more information.

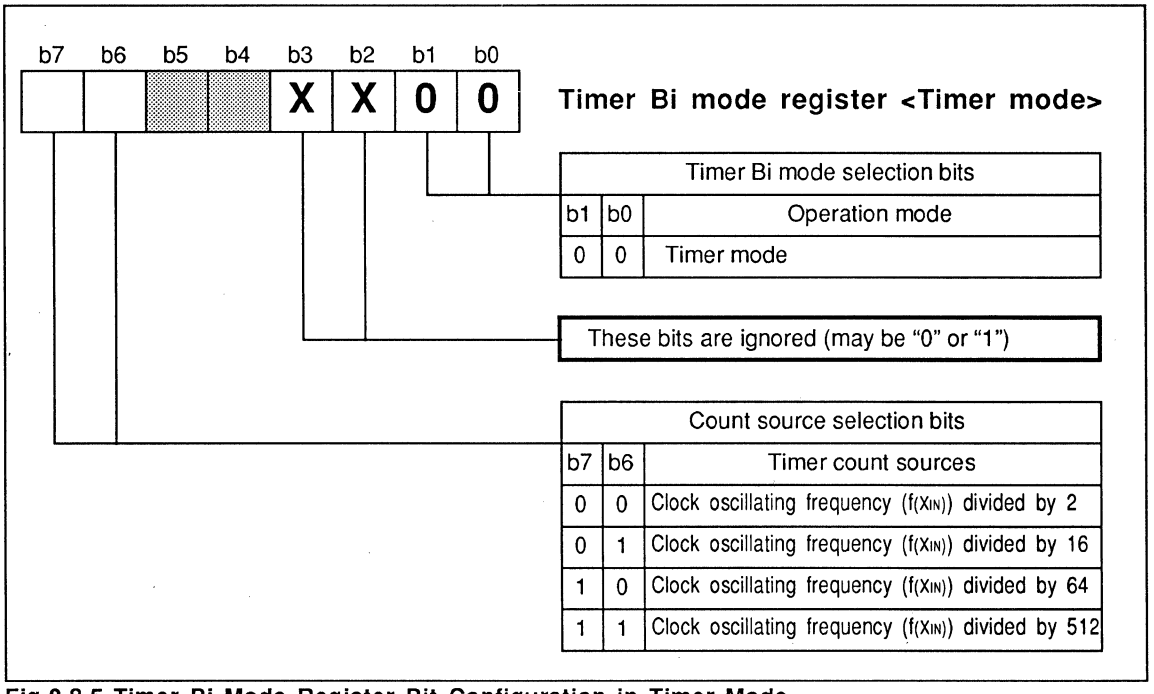


Fig.2.8.5 Timer Bi Mode Register Bit Configuration in Timer Mode

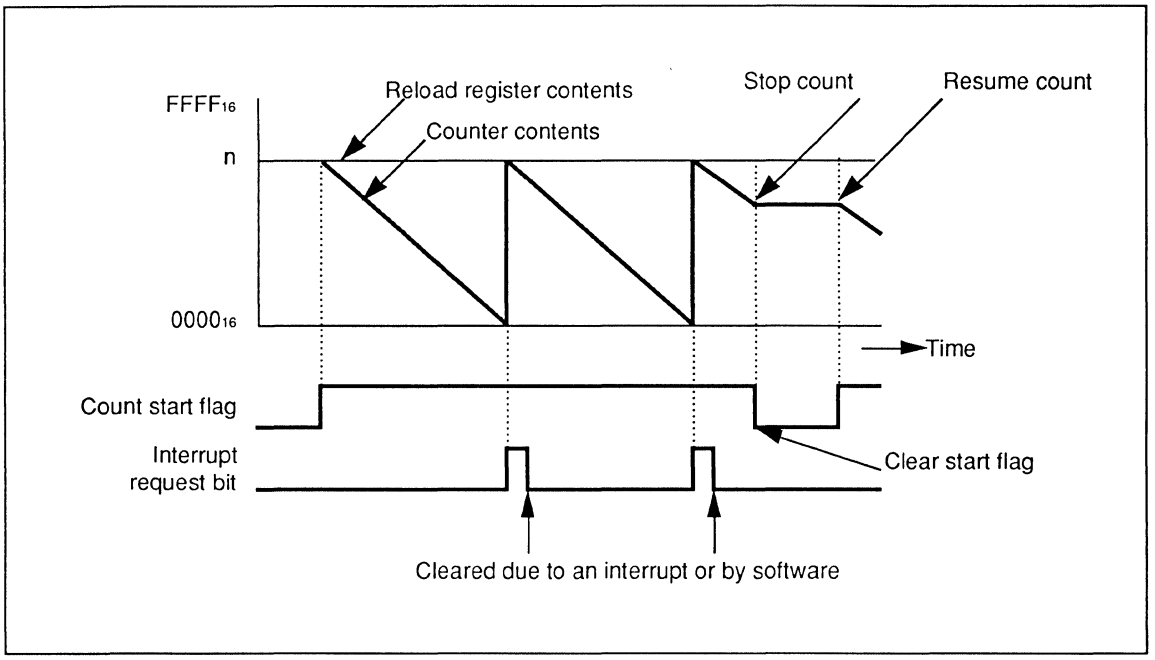


Fig.2.8.6 Timer Mode Operation

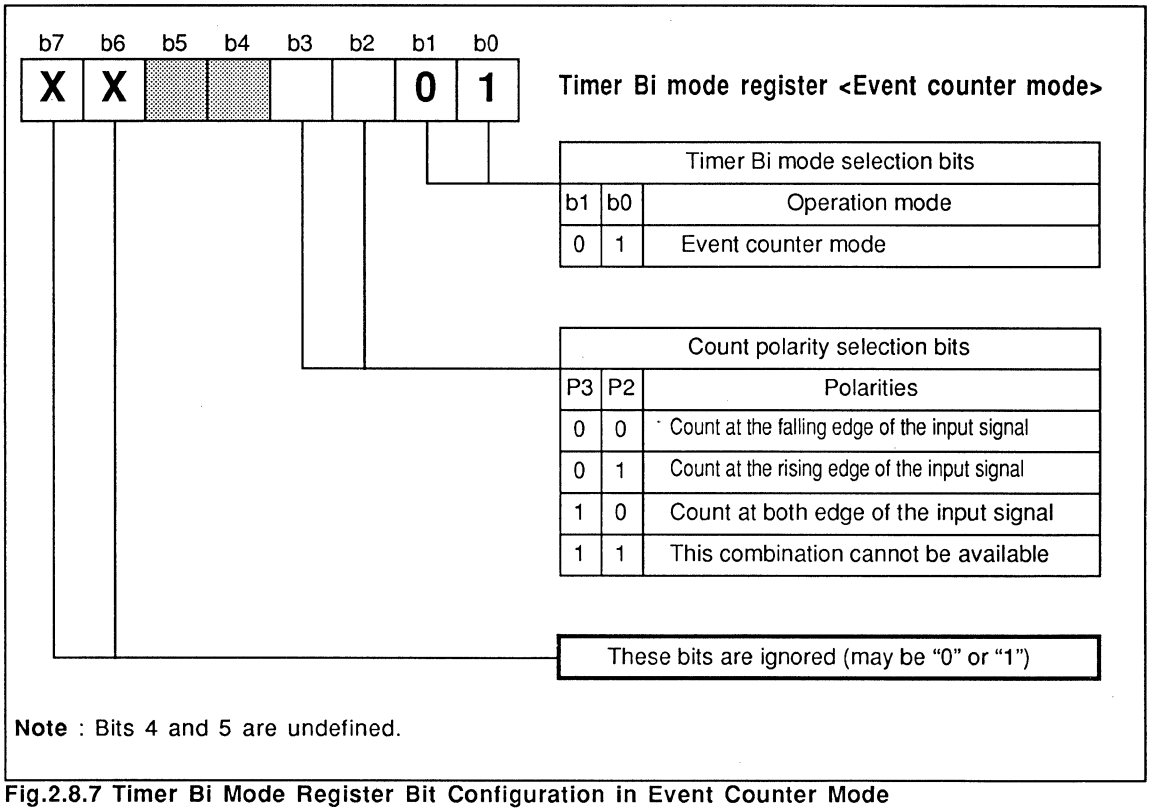
2.8.4 Event counter mode [01]

The event counter mode is selected by setting the timer Bi mode register bit 1 to "0" and bit 0 to "1". When this mode is selected, the timer Bi mode register bits 7 and 6 are ignored.
Figure 2.8.7 shows the bit configuration of the timer Bi mode register during event counter mode.

(1)Functions

In event counter mode, the external clock input from the TBi_{IN} pin is counted and an interrupt occurs each time the counter underflows.

Timer dividing ratio ..... 1/(n+1)
n: Value set in timer Bi register
(value between 0000_{16} and $FFFF_{16}$)



Note : Bits 4 and 5 are undefined.

Fig.2.8.7 Timer Bi Mode Register Bit Configuration in Event Counter Mode

(2)Event counter mode operation

First, the mode and count polarity are selected with the timer Bi mode register. Then when a value n ($n=0000_{16}$ to $FFFF_{16}$) is written in the timer Bi register, the count start flag is cleared (count disabled) and n is stored in the counter and the reload register.

When the count start flag is set to "1" (count enabled), the effective edge of the signal input to the TBi_{IN} pin is detected and counted. When the count edge selection bits (timer Bi mode register bits 3 and 2) are "00" , count is made at the falling edge of the input signal, when the bits are "01", count is made at the rising edge, and when they are "10", count is made at both edge of the input signal.

The content of the counter is decremented by 1 each time an effective edge is detected. The content of the reload register is loaded in the counter and the interrupt request bit is set to "1" at the next clock input after the content of the counter reaches 0_{16}. Count operation continues in this manner and the interrupt request bit is set to "1" each time the content of the counter changes from 0_{16} to n. Therefore, a timer Bi interrupt request occurs at every n+1 count of the clock input. The interrupt request bit remains set until the interrupt is accepted or it is cleared by program.

The content of the counter can be read at any time by reading the content of the timer Bi register, but the content of the reload register cannot be read.

The operation in event counter mode is identical to that of the timer mode except that an externally input clock is counted.

Precautions when using the event counter mode

When the event counter mode is selected, the data direction register bit of the port corresponding to the TBi_{IN} pin must be set to input ("0").

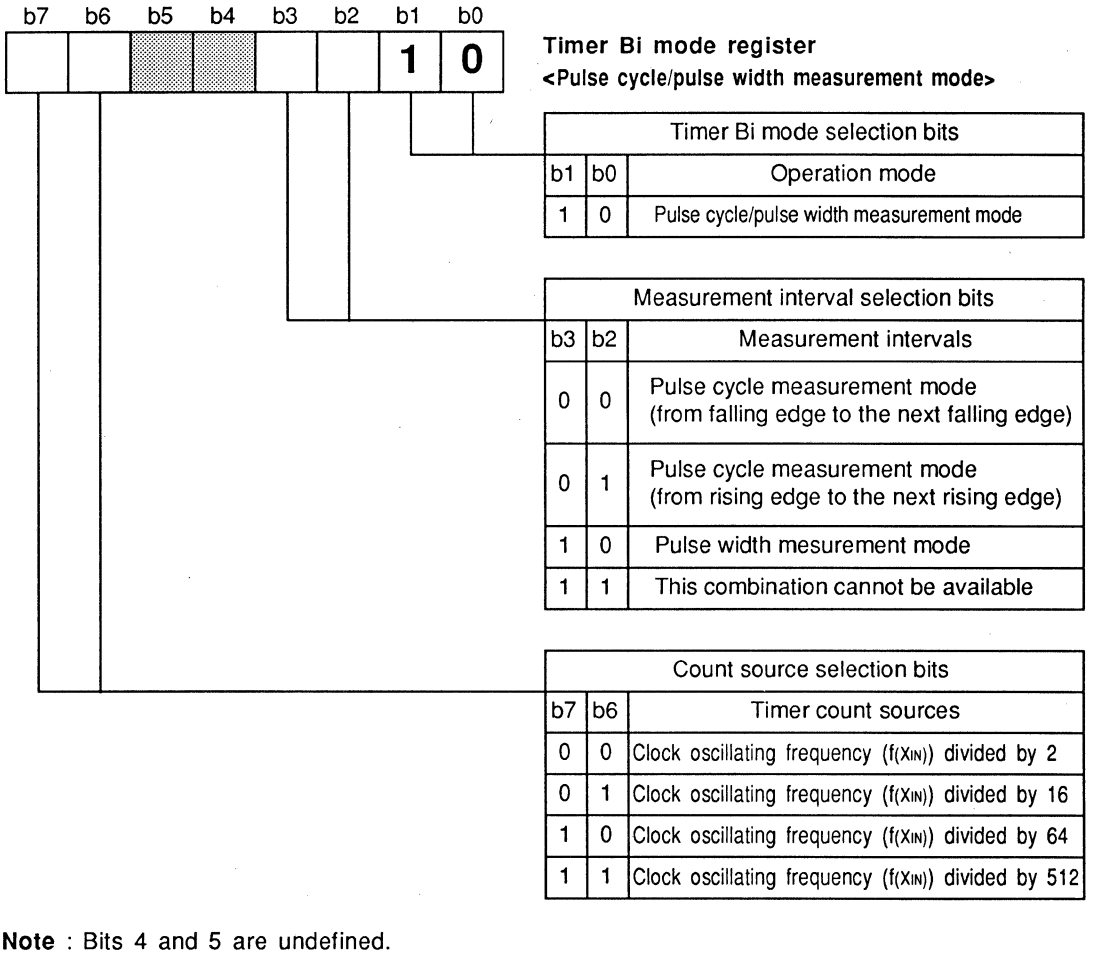
2.8.5 Pulse cycle/pulse width measurement mode [10]

The pulse cycle/pulse width measurement mode is selected by setting the timer Bi mode register bit 1 to "1" and bit 0 to "0".

Figure 2.8.8 shows the timer Bi mode register bit configuration during pulse cycle/pulse width measurement mode.

(1)Functions

This mode measures the cycle or width of the TBiIN pin input signal.



Timer Bi mode selection bits		
b1	b0	Operation mode
1	0	Pulse cycle/pulse width measurement mode

Measurement interval selection bits		
b3	b2	Measurement intervals
0	0	Pulse cycle measurement mode (from falling edge to the next falling edge)
0	1	Pulse cycle measurement mode (from rising edge to the next rising edge)
1	0	Pulse width mesurement mode
1	1	This combination cannot be available

Count source selection bits		
b7	b6	Timer count sources
0	0	Clock oscillating frequency (f(XIN)) divided by 2
0	1	Clock oscillating frequency (f(XIN)) divided by 16
1	0	Clock oscillating frequency (f(XIN)) divided by 64
1	1	Clock oscillating frequency (f(XIN)) divided by 512

Note : Bits 4 and 5 are undefined.

Fig.2.8.8 Timer Bi Mode Register Bit Configuration in Pulse Cycle/Pulse Width Measurement Mode

(2)Pulse Cycle /pulse width measurement mode description

First, the mode, count source, and whether to measure the pulse cycle or width are selected with the timer Bi mode register. The operation of the pulse cycle measurement and pulse width measurement are the same except for the effective edge of the TB_{iIN} at which the count is triggered. When the count start flag is set to "1" (count enabled), the counter starts and the selected count source is input to the counter. The counter is an incremental counter with its content incremented (+1) each time a count source clock is input.

When the effective edge (Note 2) of the TB_{iIN} input signal is detected, the content of the counter (measurement result) is transferred to the reload register.

Note 1 : In this mode, the reload register functions as a buffer register. The content of the reload register can be read by reading the timer Bi register.

Note 2: The effective edge is selected with timer Bi mode register bits 3 and 2.

When the content of the counter is transferred to the reload register, it is cleared and counting continues. This operation is repeated each time an effective edge is detected.

A timer Bi interrupt request occurs when the content of the counter is transferred to the reload register. When an interrupt request occurs, the timer Bi interrupt request bit is set to "1". The interrupt request bit remains set until the interrupt is accepted or the bit is cleared by program. However, an interrupt request does not occur when the content of the counter is transferred to the reload register at the first effective edge detected after the count start flag is set to "1".

Table 2.8.3 Effective Edge Types

b3	b2	Effective edge	Measurement Mode
0	0	Falling edge (from "H" to "L")	Pulse cycle
0	1	Rising edge (from "L" to "H")	Pulse cycle
1	0	Both edges (level change)	Pulse width

Precaution when using pulse cycle/pulse width measurement mode

1.The TB_{iIN} pin is used as the pulse input pin. Therefore, the data direction register of the corresponding port must be set to input mode.

2.When measuring signals other than 50% duty in pulse width measurement mode, whether the content of the reload register is measured at "H" level or "L" level must be determined by program. In addition, the count interval must be at least 2 cycles of the count source.

3.In this mode, an interrupt occurs when the timer overflows. Therefore, an appropriate count source must be selected to prevent timer overflow.

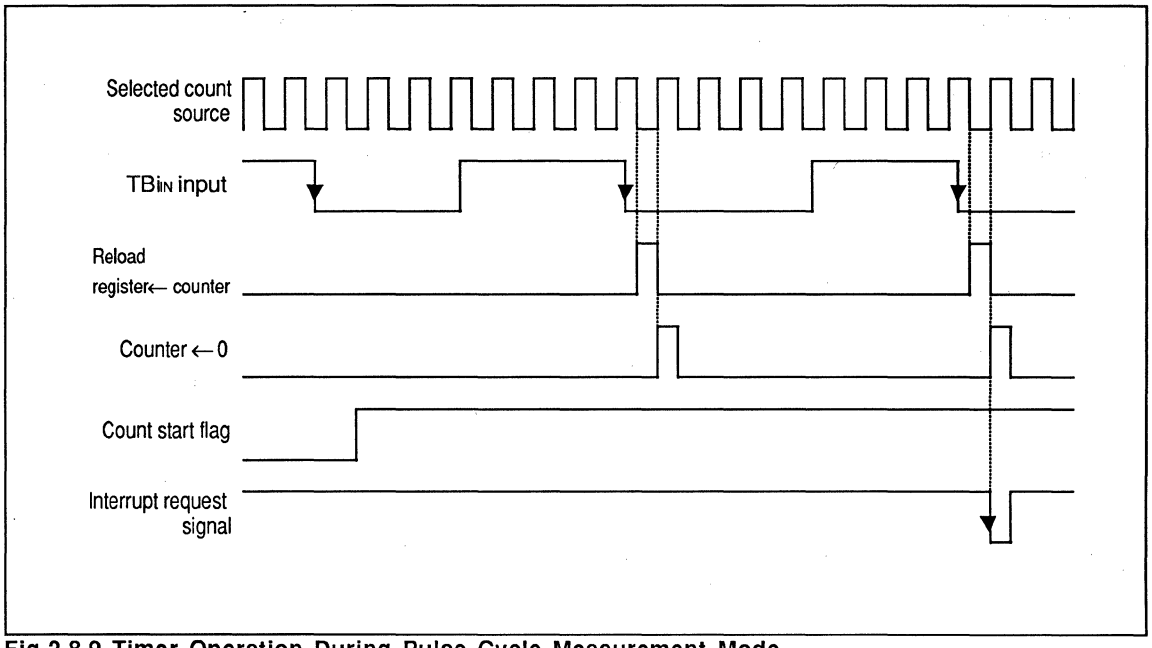


Fig.2.8.9 Timer Operation During Pulse Cycle Measurement Mode

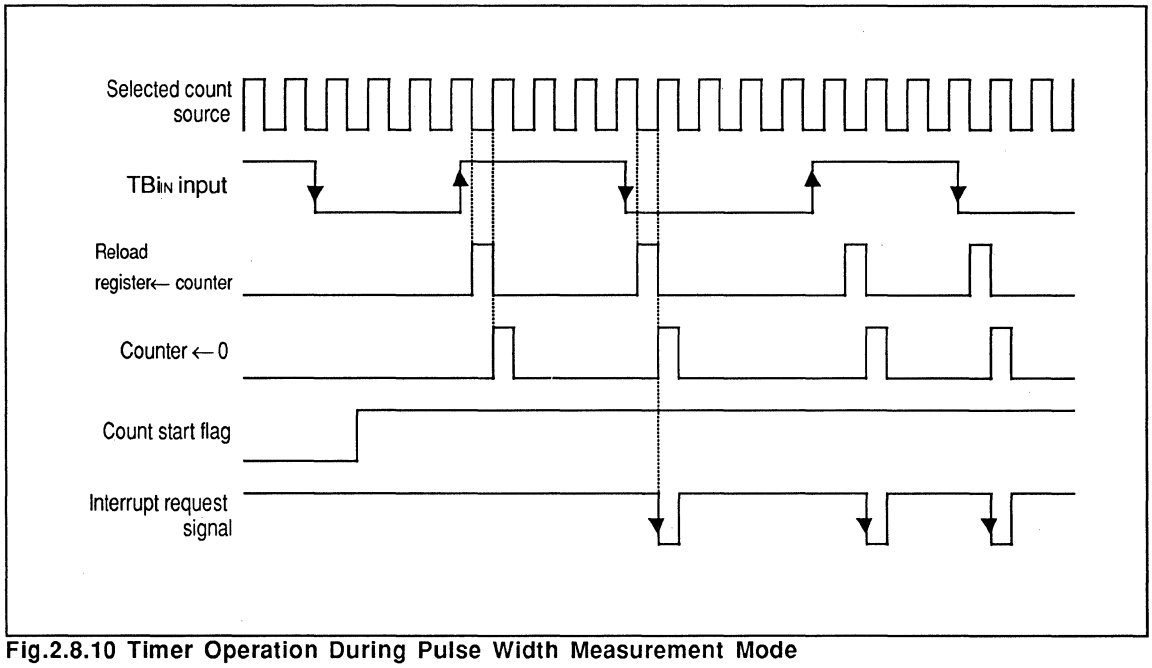


Fig.2.8.10 Timer Operation During Pulse Width Measurement Mode

2.9 Serial I/O

2.9.1 Serial I/O description

The M37700 has two serial I/O ports that can operate either as clock synchronous serial I/O port or asynchronous serial I/O (UART) port. These two ports are independent, but have identical functions. Each serial I/O port has a transfer clock generation timer (baud rate generator) and can be set a variety of data transfer rate.

Each serial I/O has four operating modes. The following modes are available:

●Clock synchronous serial I/O [001]*

In this mode, both the transmission side and receiving side use the same clock to transfer data.

●7-bit UART [100]*

In this mode, the data is transferred at an arbitrary rate and data format. The data (character) length is 7 bits.

●8-bit UART [101]*

This mode is identical to 7 bit UART except that the data length is 8 bits.

●9-bit UART [110]*

This mode is identical to 7 bit UART except that the data length is 9 bits.

*The number in brackets are the content of the serial I/O mode selection bits which are described later.

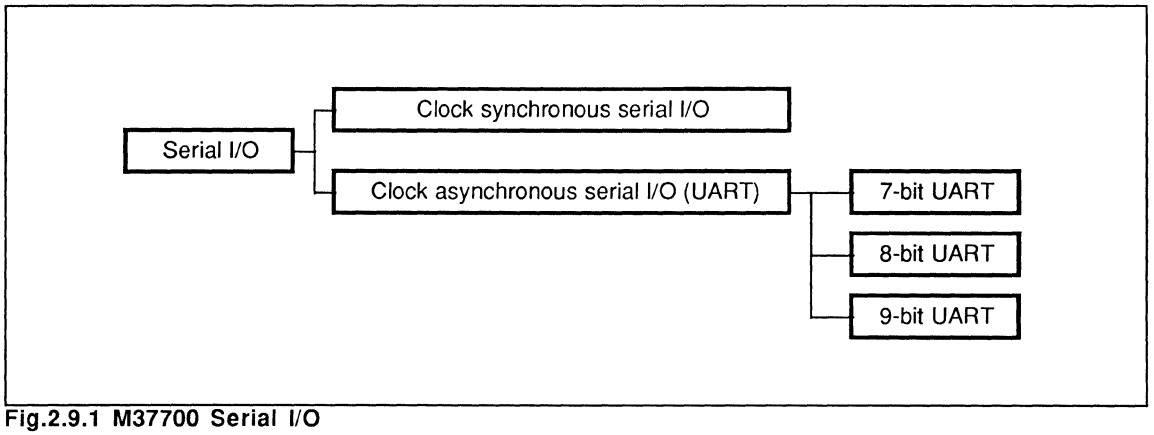


Fig.2.9.1 M37700 Serial I/O

2.9.2 Block description

Figure 2.9.2 shows a block diagram of serial I/O. The function of each block is described below.

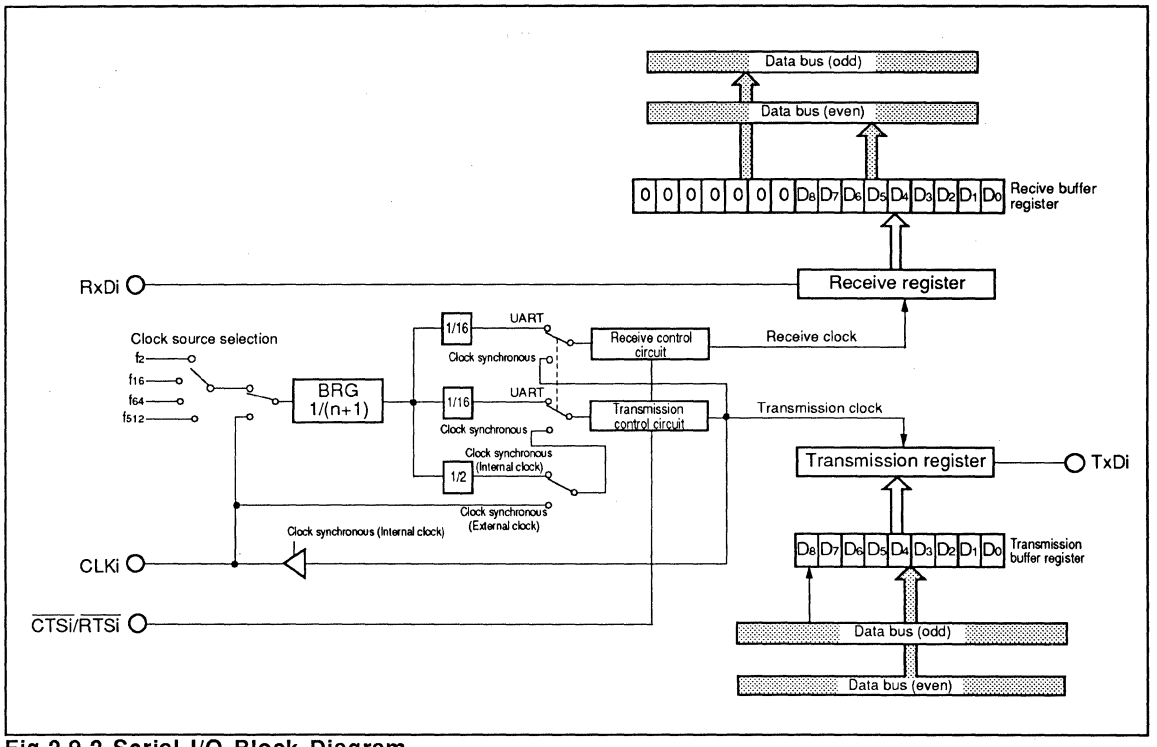


Fig.2.9.2 Serial I/O Block Diagram

(1)UARTi transmit/receive mode register

The UARTi (i=0, 1) transmit/receive mode register consists of 8 bits. This register is used to set the serial I/O mode and transfer format. Figure 2.9.3 shows the bit configuration of the UARTi transmit/receive mode register.

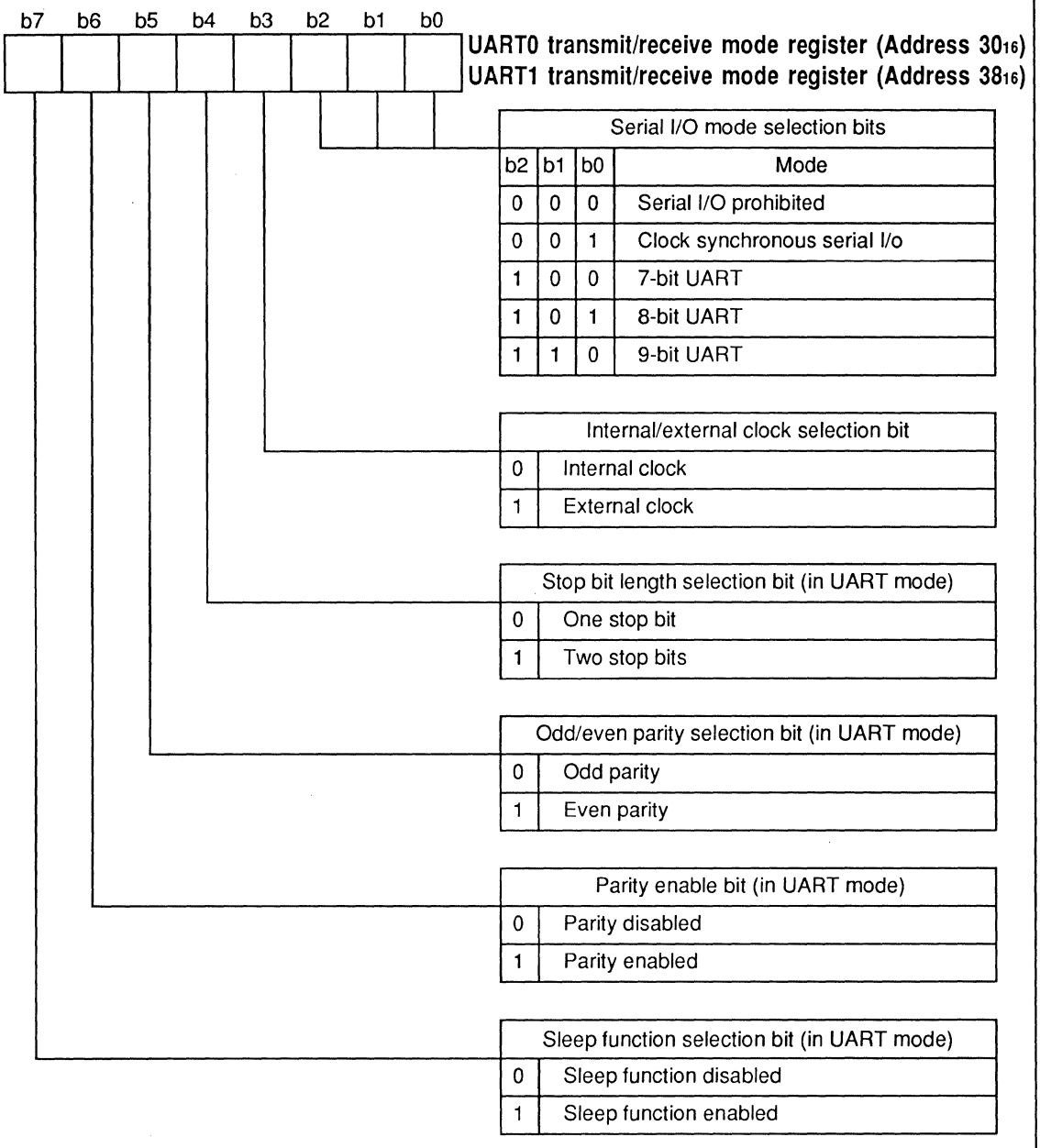
●Bits 0 to 2—Serial I/O mode selection bits

These bits are used to enable/disable serial I/O and select the function of port P8. When these bits are set to "000", serial I/O is disabled and ports $P8_0$ to $P8_3$ and $P8_4$ to $P8_7$ function as programmable I/O ports. When one of the serial I/O modes is selected, port P8 has the function shown in Table 2.9.1 and loses its programmable I/O port function (except some pins in UART mode).

Table 2.9.1 Function of Port 8 when Serial I/O is Selected

Using UART0	Using UART1	Function
$P8_0$	$P8_4$	$\overline{CTS}/\overline{RTS}$ (transmission control signal I/O pin)
$P8_1$	$P8_5$	CLK (transfer clock I/O pin) (Note)
$P8_2$	$P8_6$	RxD (serial data input pin)
$P8_3$	$P8_7$	TxD (serial data output pin)

Note:This depends on the internal/external clock selection bit as follows:
When external clock is selected : Clock input pin
When internal clock is selected : Clock output pin in clock synchronous mode and normal I/O port in UART mode
When the $\overline{CTS}$ input ($P8_0$ or $P8_4$), external transfer clock ($P8_1$ or $P8_5$), and RxD ($P8_2$ or $P8_6$) are used, the corresponding data direction register must be set to "0" (input mode).



Note:Bits 4 to 6 are ignored in clock synchronous mode.
Bit 7 must be set to "0" when using clock synchronous mode.
This register is cleared to "00₁₆" at reset.

Fig.2.9.3 UARTi Transmit/Receive Mode Register Bit Configuration

●**Bit 3—Internal/external clock selection bit**

[Clock synchronous mode]

This bit is used to select either an internal clock or an external clock as the synchronous clock (shift clock) for data transfer.

When this bit is set to "0" to select an internal clock, the later described BRGi generated clock divided by 2 is used as the shift clock. In addition, the CLKi pin becomes the output pin and the shift clock is output from this pin.

When this bit is set to "1" to select an external clock, The CLKi pin becomes the input pin and data transfer is synchronized with the clock input to this pin.

[UART mode]

This bit is used to select either an internal clock or an external clock as the input to the BRGi which is described later.

When this bit is set to "0" to select an internal clock, the clock selected with the BRG count source selection bit in the UARTi control register becomes the BRG input clock. In this case, the CLKi pin can be used as a programmable I/O pin.

When this bit is set to "1" to select an external clock, the CLKi pin becomes the clock input pin and the clock input to this pin becomes the BRGi input clock.

●**Bit 4—Stop bit length selection bit**

[Clock synchronous mode]

This bit is ignored. (It can be either "0" or "1".)

[UART mode]

This bit is use to select between 1 and 2 bits as the stop bit to indicate the end of data.

●**Bit 5—Odd/even parity selection bit**

[Clock synchronous mode]

This bit is ignored (it can be either "0" or "1").

[UART mode]

This bit is used to select between even parity and odd parity. This bit is ignored if the parity enable bit is set to "0" (disabled).

●**Bit 6—Parity enable bit**

[Clock synchronous mode]

This bit is ignored (it can be either "0" or "1").

[UART mode]

This bit is used to specify whether to add a parity bit at the end of transfer data. Whether to use odd parity or even parity is specified with bit 5.

●**Bit 7—Sleep function selection bit**

[Clock synchronous mode]

This bit must be set to "0".

[UART mode]

This bit is used to enable or disable the sleep function (see "2.9.5 (6) Sleep mode"). If the sleep function is enabled, the data is ignored when the most significant bit (MSB) of the received data is "0". This function is used when multiple microcomputers are connected through the serial I/O port.

(2)UARTi transmit/receive control register 0

The UARTi transmit/receive control register 0 consists of bits to select the BRG count source and $\overline{CTS}/\overline{RTS}$ pin function, and a flag that indicates the transmission register status. Figure 2.9.4 shows the bit configuration of the UARTi transmit/receive control register 0.

●Bit 0 to 1—BRG counter source selection bits

This bit is used to select the count source of the baud rate generator (BRG) when an internal clock is selected. The count source can be either 1/2 (f_2), 1/16 (f_{16}), 1/64 (f_{64}), or 1/512 (f_{512}) of the source oscillating frequency $f(X_{IN})$.

●Bit 2—$\overline{CTS}/\overline{RTS}$ function selection bit

This bit is used to specify whether to use the $P8_0$ pin (when using UART0) or $P8_4$ (when using UART1) as $\overline{CTS}$ input pin or $\overline{RTS}$ output pin.

When this bit is "0", $P8_0$ or $P8_4$ becomes a $\overline{CTS}$ input pin and this pin must be at "L" level in order for transmission to start.

When this bit is "1", $P8_0$ or $P8_4$ becomes an $\overline{RTS}$ output pin and "H" level is output when receive is disabled (receive enable flag is "0") and "L" level is output when receive is enabled (receive enable flag is "1"). It returns to "H" when receive starts.

●Bit 3—Transmit register empty flag

This bit is set to "0" when the content of the transmit buffer is transferred to the transmission register. It is set to "1" when transmission completes and the transmission register becomes empty.

●Bits 4 to 7

These bits are undefined because no memory is allocated.

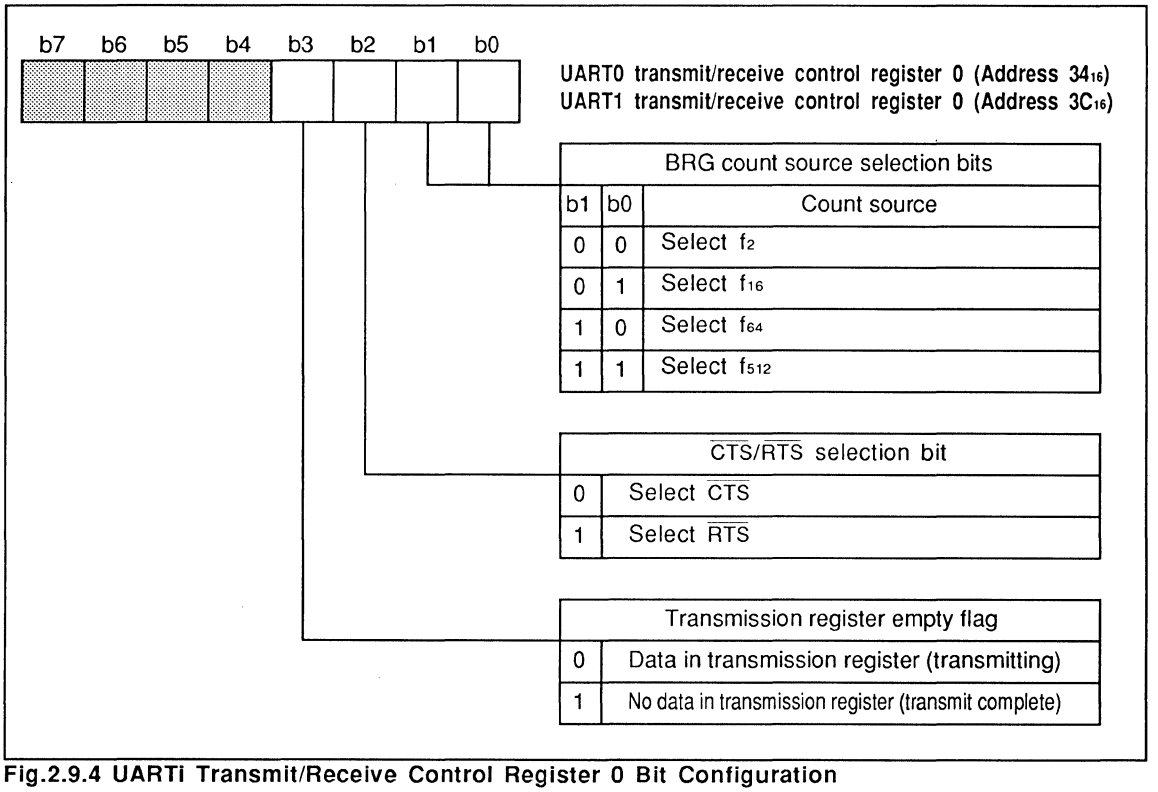
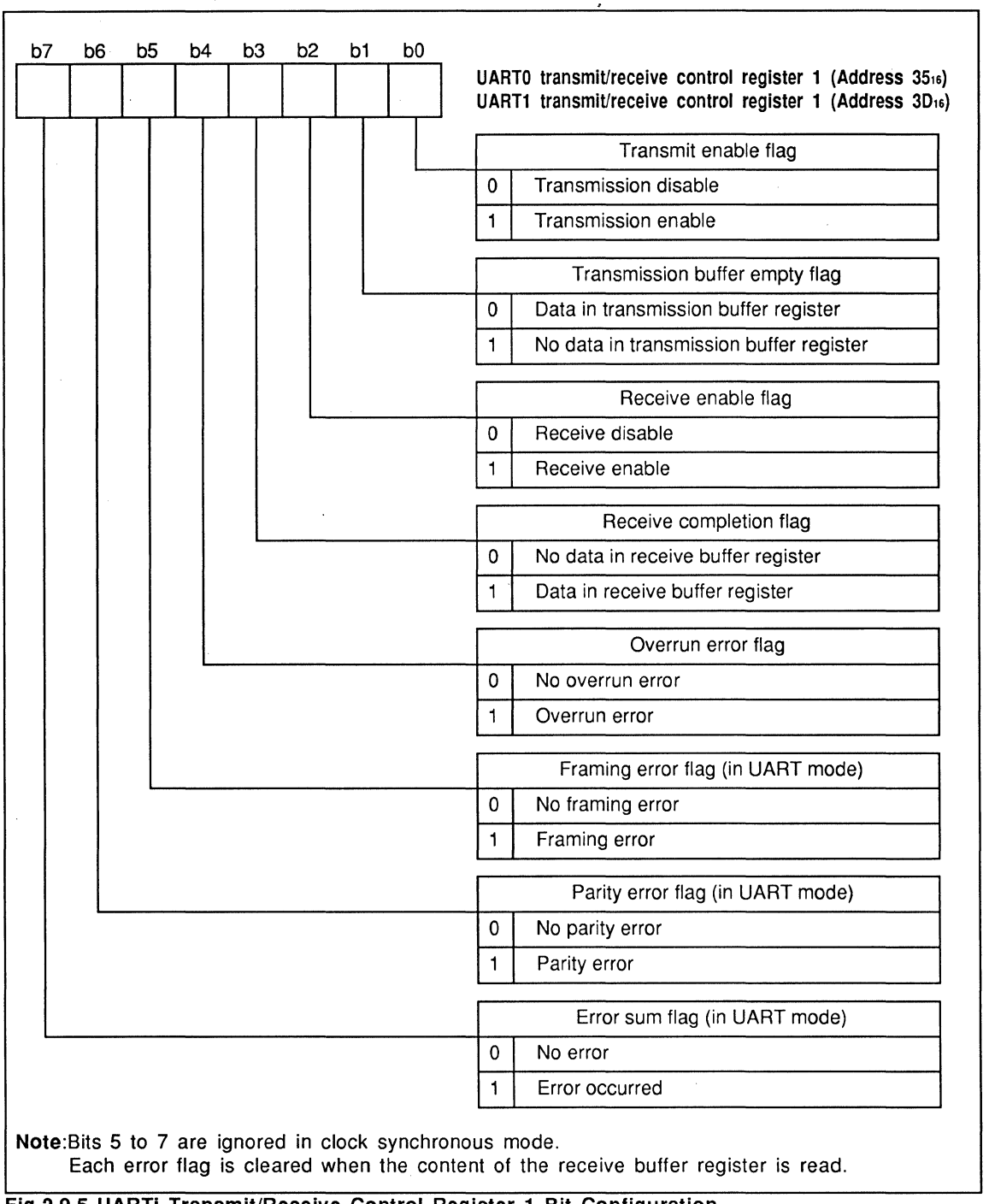


Fig.2.9.4 UARTi Transmit/Receive Control Register 0 Bit Configuration

(3) UARTi transmit/receive control register 1

The UARTi transmit/receive control register 1 consists of serial I/O enable bit, serial I/O status flag, and serial I/O error flags. Figure 2.9.5 shows the bit configuration of the UARTi transmit/receive control register 1.



Note:Bits 5 to 7 are ignored in clock synchronous mode.
Each error flag is cleared when the content of the receive buffer register is read.

Fig.2.9.5 UARTi Transmit/Receive Control Register 1 Bit Configuration

●Bit 0—Transmission enable bit
Serial I/O transmission is enabled when this bit is set to "1".

●Bit 1—Transmission buffer empty flag
This bit indicates the status of the transmission buffer register. It is set to "1" when the content of the transmission buffer is sent to the transmission shift register. It is automatically cleared when data is written in the transmission buffer register.

●Bit 2—Receive enable flag
Serial I/O receive is enabled when this flag is set to "1". If the $\overline{RTS}$ function is selected, the $\overline{RTS}$ pin level becomes "L" when this flag is set to "1".

●Bit 3—Receive completion flag
This flag is set to "1" when the data in the receive register is transferred to the receive buffer register (receive completion). This bit is cleared when the low-order byte of the receive buffer register is read or when the receive enable flag is set to "0" (receive disabled).

●Bit 4—Overrun error flag
This flag is set to "1" when receiving of the next data completes and the content of the receive buffer register is updated while there is data remaining in the receive buffer register (before the content of the receive buffer register is read).
This flag is cleared when the low-order byte of the receive buffer register is read or when the receive enable flag is set to "0" (receive disabled).

●Bit 5—Framing error flag
[Clock synchronous mode]
This bit is ignored.

[UART mode]
This flag is set to "1" when the number of stop bits is not the number specified with bit 4 of the UARTi transmission mode register. This flag is cleared when the low-order byte of the receive buffer register is read or when the receive enable flag is set to "0" (receive disabled).

●Bit 6—Parity error flag
[Clock synchronous mode]
This bit is ignored.

[UART mode]
This flag is set to "1" when there is a parity error. This flag is cleared when the low-order byte of the receive buffer register is read or when the receive enable flag is set to "0" (receive disabled).

●Bit 7—Error sum flag
[Clock synchronous mode]
This bit is ignored.

[UART mode]
This flag is set when either an overrun error, a framing error, or a parity error occurs. This flag is cleared when the low-order byte of the receive buffer register is read or when the receive enable flag is set to "0" (receive disabled).

(4)Transmission register and transmission buffer register

When transmit conditions are satisfied, the transmit data written in the transmission buffer register is transferred to the transmission register and is synchronously transmitted from the TxD pin with the specified clock. In clock synchronous mode and 7 or 8 bit UART mode, only the low-order byte of the transmission buffer register is used. In 9 bit UART mode, bit 8 of the transmit data is written in bit 0 of the high-order byte, and the remaining 7 to 0 bits are written in the low-order byte.
The transmission buffer register becomes empty after the data is transferred to the transmission register. Therefore, the next transmit data can be written during transmission.
The content of the transmission buffer register cannot be read because it is a write only register.

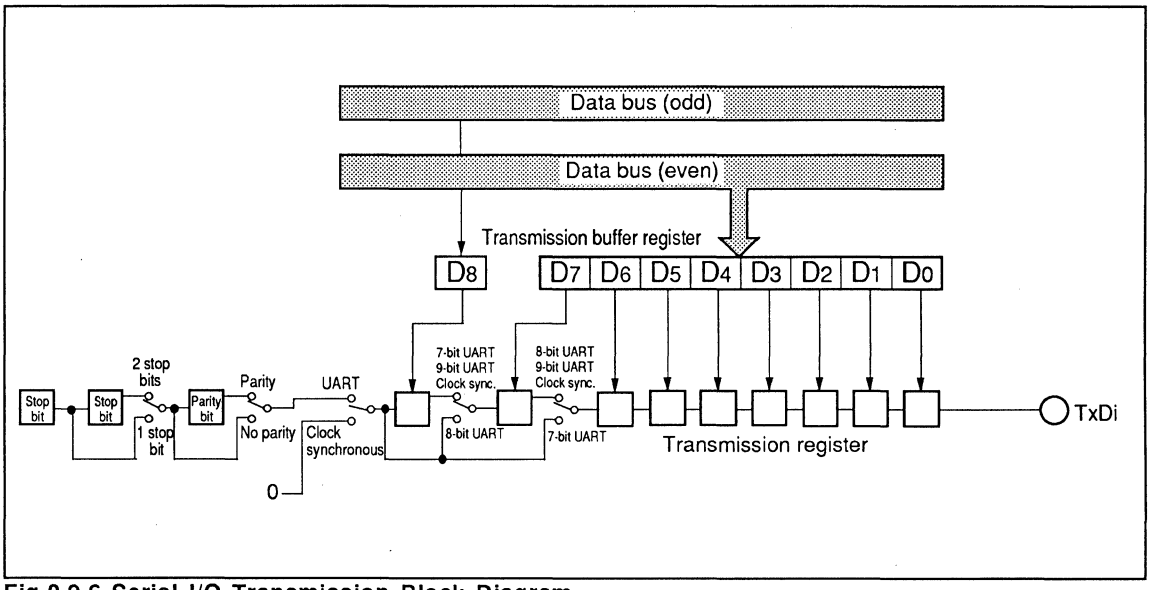


Fig.2.9.6 Serial I/O Transmission Block Diagram

(5)Receive register and receive buffer register

The receive register converts serial data input from the RxD pin to parallel data. The RxD pin level is moved bit by bit to the receive register synchronized with the rising edge of the synchronous clock. The contents of the high-order 7 bits of the receive buffer register can be always read "0". And, the unused bits (D_7 and D_8 in 7-bit UART mode and D_8 in 8-bit UART mode) of the low-order 9 bits can be read the same data as the MSB (most significant bit) of effective receive data.
Note that the content of the receive buffer register will be updated if the next receive data becomes available before the receive buffer register is read.

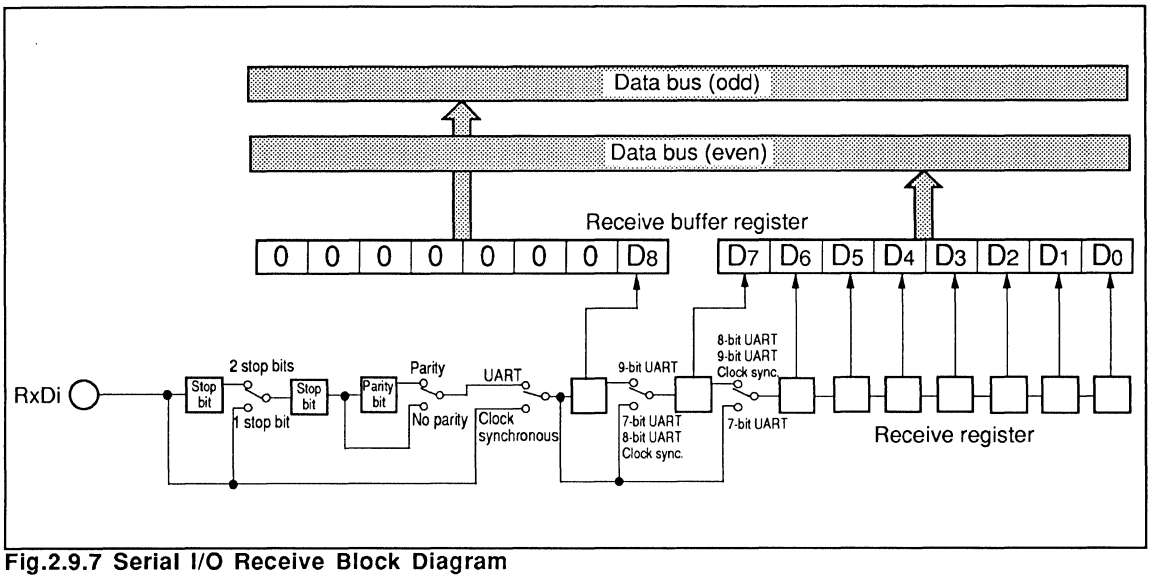


Fig.2.9.7 Serial I/O Receive Block Diagram

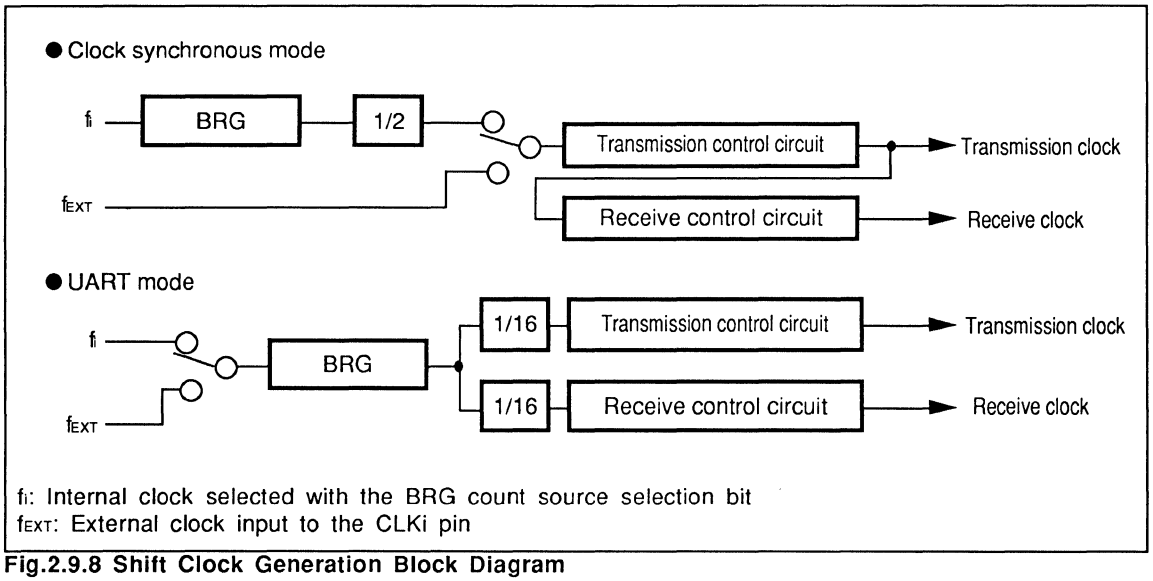
(6)Baud rate generator

The baud rate generator (BRG) is a timer used exclusively for serial I/O. It is equipped with a reload register and consists of 8 bits. The BRG divides the input clock by (n+1), where n is the value set in the BRG register. This register can contain a value between 0 (00_{16}) and 255 (FF_{16}).

In clock synchronous serial I/O mode, BRG becomes effective when an internal clock is selected and the BRG output divided by two becomes the transmit/receive clock.

In UART mode, the BRG is effective regardless of the clock type and the BRG output divided by 16 becomes the transmit/receive clock.

The content of the BRG register cannot be read because it is a write only register.



fi: Internal clock selected with the BRG count source selection bit
fEXT: External clock input to the CLKi pin

Fig.2.9.8 Shift Clock Generation Block Diagram

(7) UARTi transmission interrupt control register/UARTi receive interrupt control register

Transmission interrupt and receive interrupt can be used when the serial I/O function is selected. Each of these interrupts has an interrupt control register which is used to set the enable condition (priority level) and check the existence of an interrupt request.

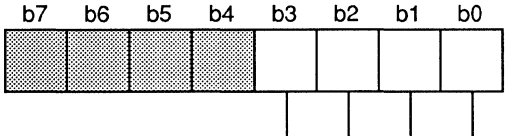
●Interrupt priority level

These bits specify an interrupt priority level between 0 and 7. When an interrupt occurs, this level is compared with the IPL in the processor status register. The interrupt is enabled when this priority level is higher than the IPL (see "Section 2.6 Interrupts").

●Interrupt request bit

The transmission interrupt request bit is set to "1" when data is transferred from the transmission buffer register to the transmission register for data transmission.

The receive interrupt request bit is set to "1" when data receive completes and data is transferred from the receive register to the receive buffer register.



UART0 transmission interrupt control register (Address 71_{16})
UART1 transmission interrupt control register (Address 73_{16})

Interrupt priority level selection bits			
b2	b1	b0	Level
0	0	0	Level 0 (interrupt disabled)
0	0	1	Level 1
0	1	0	Level 2
0	1	1	Level 3
1	0	0	Level 4
1	0	1	Level 5
1	1	0	Level 6
1	1	1	Level 7

Interrupt request bit	
0	No interrupt request
1	Interrupt request

Note:The bit configuration of the UARTi receive interrupt control register is the same as the UARTi transmission interrupt control register (i=0, 1).

Fig.2.9.9 UARTi Transmission Interrupt Control Register Bit Configuration

2.9.3 Serial I/O operation mode selection

In order to use the serial I/O function, the serial I/O operation mode must be selected first. The operation mode is selected with bits 2 to 0 (serial I/O mode selection bits) of the UARTi transmit/receive mode register (see Table 2.9.3). When a serial I/O mode is selected, the serial I/O function becomes effective and port P8 ($P8_0$ to $P8_3$ when using UART0 and $P8_4$ to $P8_7$ when using UART1) changes to serial I/O pin. When a port changes to a serial I/O port, it loses its programmable I/O port function and the corresponding data direction register and port register are ignored.

The clock synchronous serial I/O and the clock asynchronous serial I/O (UART) functions are described below.

Table 2.9.2 Serial I/O Mode Selection Registers

Serial I/O mode	Operation mode selection register
UART0	UART0 transmit/receive mode register (address 30_{16})
UART1	UART1 transmit/receive mode register (address 38_{16})

Table 2.9.3 Relation between Serial I/O Mode Selection Bits and Operation Mode

b2	b1	b0	Operation Mode	Port P8 Function
0	0	0	Serial I/O disabled	Programmable I/O port
0	0	1	Clock synchronous serial I/O	Serial I/O function pins
0	1	0	This cannot be available	—
0	1	1	This cannot be available	—
1	0	0	7-bit UART	Serial I/O function pins
1	0	1	8-bit UART	Serial I/O function pins
1	1	0	9-bit UART	Serial I/O function pins
1	1	1	This cannot be available	—

Note: The serial I/O mode selection bits must be set to "000" when serial I/O is not used. These bits must not be set to "010", "011", or "111".

2.9.4 Clock Synchronous serial I/O

Table 2.9.4 shows the performance of clock synchronous mode serial I/O.

Table 2.9.4 Clock Synchronous Serial I/O Description

Parameter		Function
Data format		8 bit fixed, LSB first
Transmission speed	Internal clock	BRG output/2
	External clock	2Mbps maximum (at $f(X_{IN})$=8 MHz) 4Mbps maximum (at $f(X_{IN})$=16MHz)
Transmit/receive control		$\overline{CTS}$ input or $\overline{RTS}$ output can be selected by a program.

(1)Synchronous clock (shift clock)

The serial I/O data transfer rate is determined by the synchronous clock (shift clock). The M37700 can select whether to generate this clock internally or to use an external clock. The synchronous clock is generated internally when the transmission mode register bit 3 is set to "0" and externally when it is set to "1".

●Internal generation of synchronous clock

When the clock internal/external selection bit is set to "0", the BRG output divided by 2 is used as the synchronous clock. In this case, the CLK pin becomes output mode and the transmit synchronous clock is output from the CLK pin.

The BRG is a serial I/O timer consisting of 8 bits and is used as a frequency divider to generate the desired frequency. The BRG divides the clock selected with UARTi transmit/receive control register 0 bits 1 and 0 by n+1 and then by 2. "n" is the value set in the BRG register. It can be a value between 0_{16} and FF_{16}.

Synchronous clock frequency ..... $f_i/(2(n+1))$

f_i : BRG input frequency

Eight synchronous clocks are generated by activating the transmitter.

●Using external input clock as synchronous clock

When an external clock is selected, the CLK pin becomes the input pin and the clock input to this pin becomes the synchronous clock.

Precautions when using clock synchronous serial I/O

In clock synchronous mode, the synchronous clock used for data transfer is generated by activating a transmitter. Therefore, the transmitter must be activated even when performing receive only.

(2)Serial I/O data transmission

The data transmission method in clock synchronous serial I/O mode is described below.

[Setting the control registers]

Set each serial I/O control register for transmission.

●Transmit/receive mode register

- Operation mode
  Set the serial I/O mode selection bits to 001.
- Synchronous clock
  Select either an internal clock ("0") or an external clock ("1") with the synchronous clock selection bit.
- Set bit 7 to "0" (disable sleep mode).

●Transmit/receive control register 0

- Enable/disable $\overline{CTS}$ function
  $\overline{CTS}$ is used when the start of serial data transmission is controlled externally. One of the conditions of starting transmission is setting the $\overline{CTS}$ pin to "L" when using the $\overline{CTS}$ function.

●Transmit/receive control register 1

- Set transmission enable flag to "1".

After initialization, write the data to be transmitted in the low-order byte of the transmission buffer register. At the same time, the transmission buffer register empty flag becomes "0".

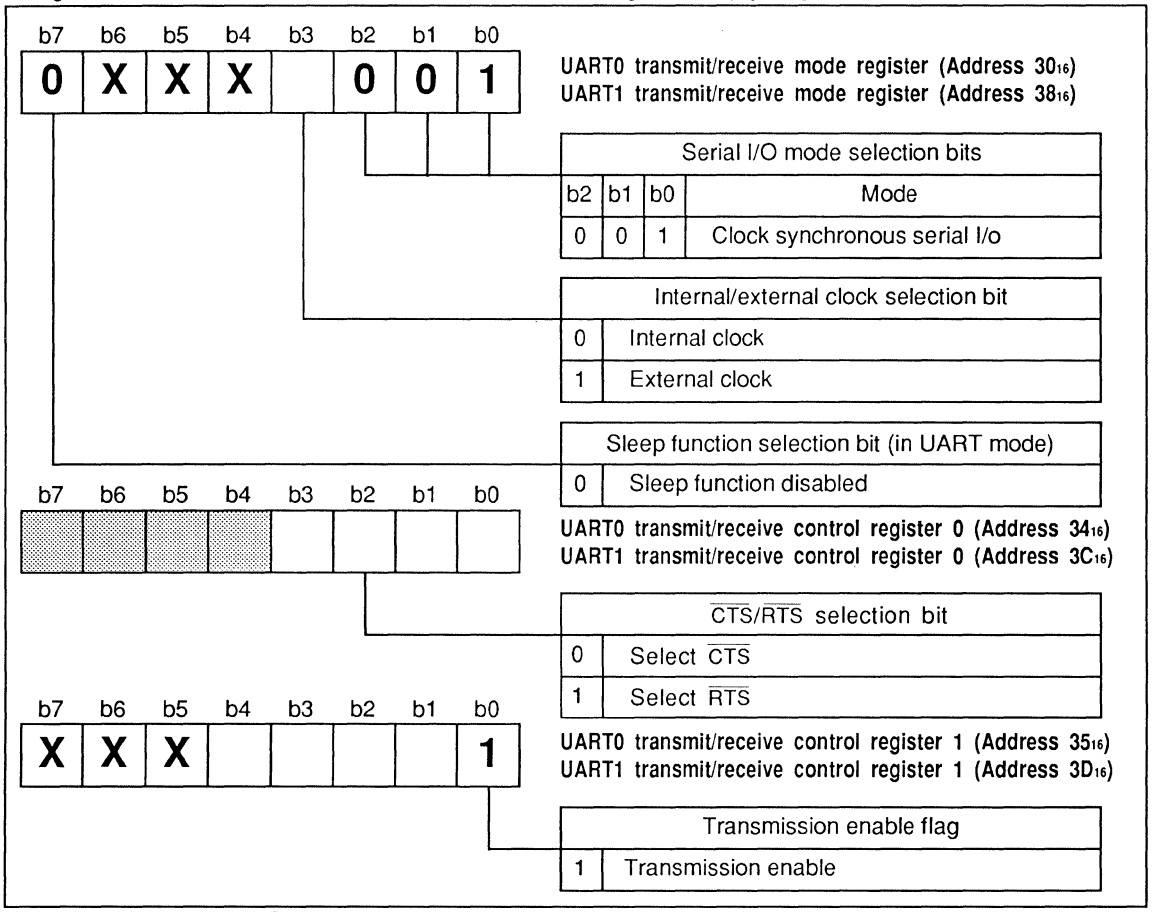


Fig.2.9.10 Setting the Control Registers

[Transmit operation]
The transmission of serial data starts when the following conditions are satisfied.

①Transmission enabled (transmission enable flag is "1").
②Transmit data is available in the transmission buffer (transmission buffer empty flag is "0").
③$\overline{CTS}$ pin input level is "L".
(**Note**: This condition is ignored if the $\overline{CTS}$ function is not selected.)

When the above three conditions are satisfied (two if $\overline{CTS}$ function is not selected), the content of the transmission buffer register is transferred to the transmission register and eight shift clocks are generated. At this point, the transmission buffer empty flag is set to "1" and the transmission register empty flag is cleared to "0". The shift clock is input to the transmission control circuit and the data in the transmission register is transmitted bit by bit from the TxD pin (starting at the low-order bit) at each falling edge of this clock. When the 1-byte data transmission is completed by the eighth shift clock, the transmission register empty flag is set to "1".

The synchronous clock is generated continuously if the conditions for the next data are satisfied when a transmission completes. Therefore, to transmit data continuously, the next data should be written in the transmission buffer register while data is being transmitted (when the transmission register empty flag is 0). If the conditions to transmit the next data are not satisfied, the synchronous clock halts at "H".

[Transmission interrupt]
The transmission interrupt request bit is set to "1" when the content of the transmission buffer register is transferred to the transmission register. To use the transmit interrupt, the interrupt priority level in the UARTi transmission interrupt control register must be set to 1 or greater before an interrupt occurs. In addition, the interrupt disable flag must be cleared to "0". (See "Section 2.6 interrupts" for more information on interrupt enable conditions.) The transmission interrupt request flag is cleared when an interrupt is accepted. It can also be cleared from a program.

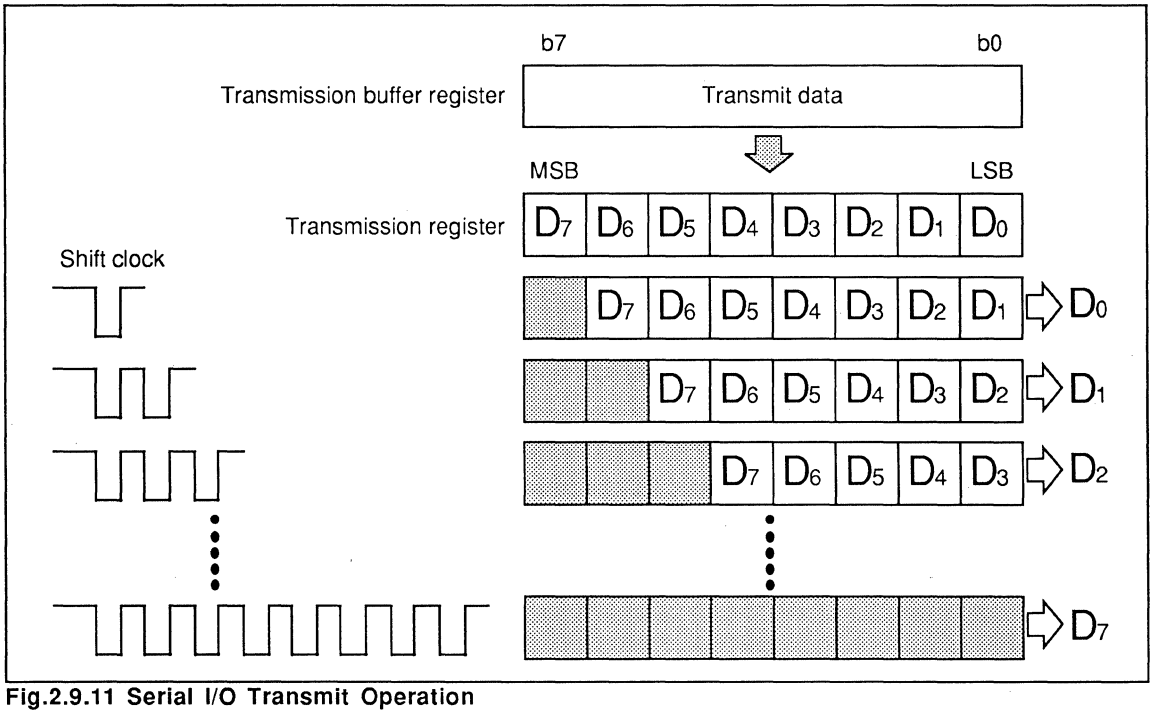
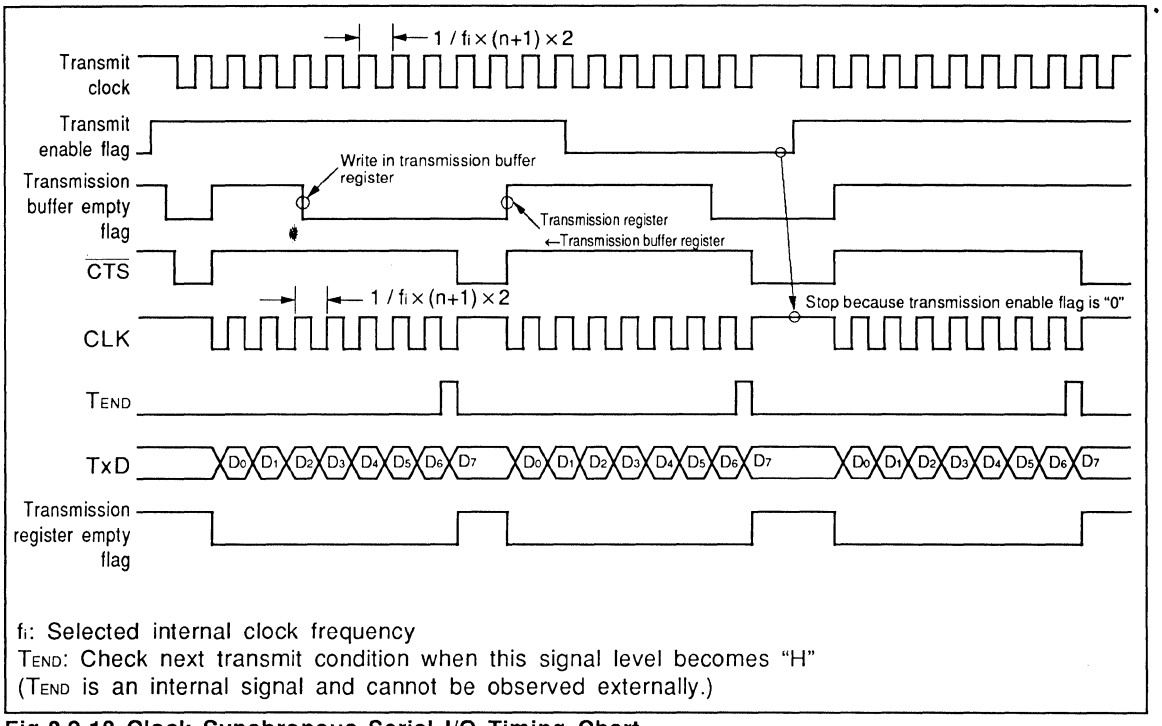


Fig.2.9.11 Serial I/O Transmit Operation



fi: Selected internal clock frequency
TEND: Check next transmit condition when this signal level becomes "H"
(TEND is an internal signal and cannot be observed externally.)

Fig.2.9.12 Clock Synchronous Serial I/O Timing Chart

(3) Serial data receive

The data receive method in clock synchronous serial I/O mode is described below.

[Setting the control registers]

●Transmit/receive mode register

- Operation mode
  Set the operation mode selection bits to 001.
- Synchronous clock
  Select either an internal clock ("0") or an external clock ("1") with the synchronous clock selection bit.
- Set bit 7 to "0" (disable sleep mode).

●Transmit/receive control register 0

- Enable/disable $\overline{\text{CTS}}$ function
  $\overline{\text{CTS}}$, $\overline{\text{RTS}}$ selection bit
  "0": $\overline{\text{RTS}}$ function disabled ($\overline{\text{CTS}}$ function selected).
  "1": $\overline{\text{RTS}}$ function selected.
  The $\overline{\text{RTS}}$ function is used to notify externally when ready to receive serial data. The $\overline{\text{RTS}}$ pin is normally at "H" level. It becomes "L" when the receive enable flag is set to "1".

●Transmit/receive control register 1

- Set transmission enable flag to "1" (transmission enabled).
- Set receive enable flag to "1" (receive enabled).

●Write data in transmission buffer register

Internal clock → Shift clock is generated and receiving starts.
External clock → Starts receiving as soon as clock is input to CLK pin.

Precautions when using clock synchronous serial I/O receive

In clock synchronous serial I/O, the shift clock is generated by activating a transmitter. Therefore, a transmit operation must be performed even when performing receive only. Note that in this case, a dummy data is output from the TxD pin.

When an internal shift clock is selected, the shift clock is generated by enabling transmission and writing a dummy data in the transmission buffer register. When an external shift clock is selected, receive starts as soon as the shift clock is input to the CLK when the transmission enable bit is set to "1" and a dummy data is written in the transmission buffer register.

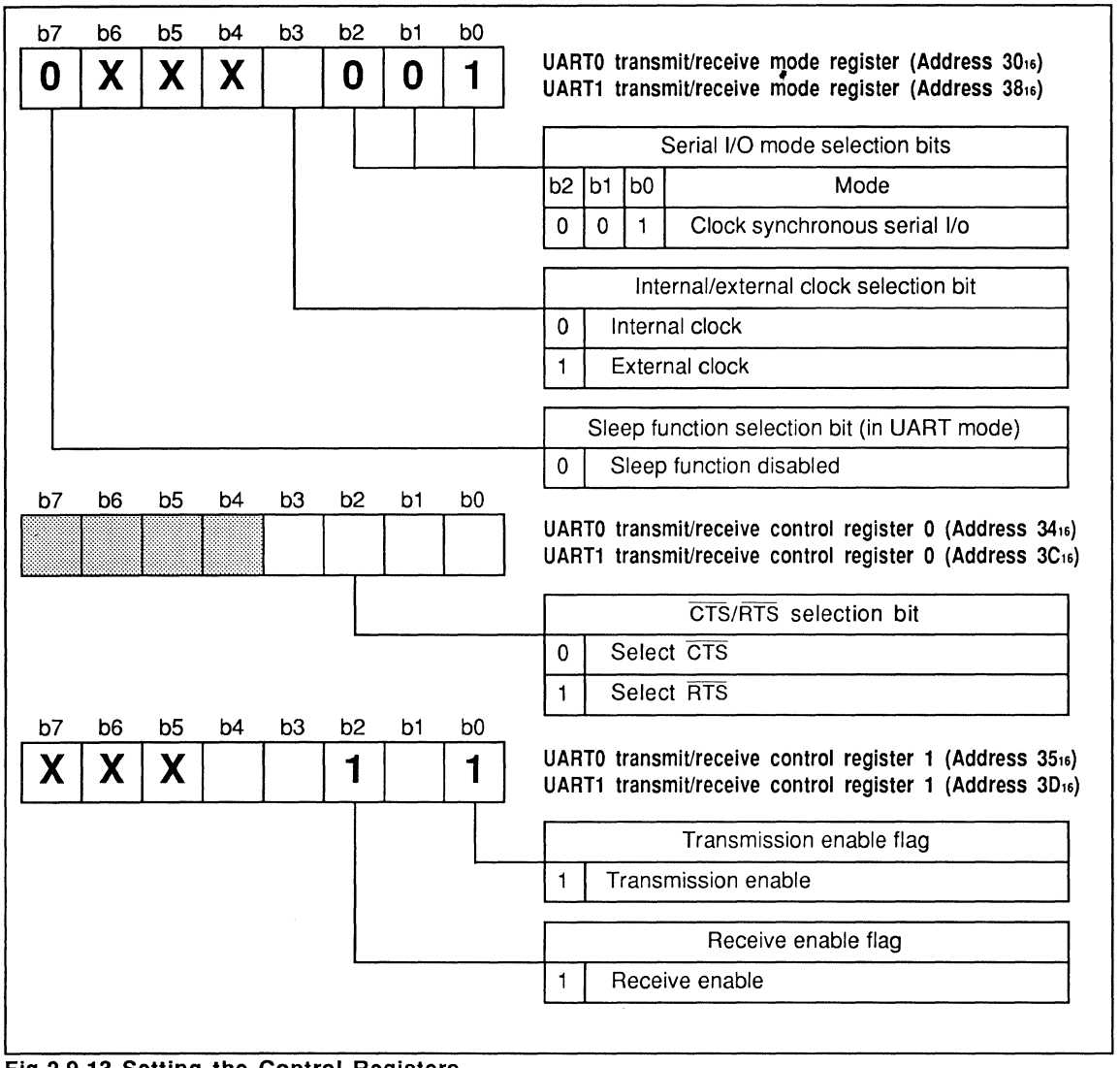


Fig.2.9.13 Setting the Control Registers

[Receive operation] (When using an external clock)
Serial data receive is enabled by enabling transmission and setting the receive enable flag to "1". When the receive enable flag is set to "1" the $\overline{RTS}$ pin level becomes "L" to indicate externally that the microprocessor is ready to receive serial data (when $\overline{RTS}$ function is selected). The transmit and receive timing can be synchronized by connecting the $\overline{RTS}$ output to the $\overline{CTS}$ pin on the transmit side. The RxD pin level is used to establish the most significant bit of the receive register at the rising edge of the shift clock input to the CLK pin and the content of the receive register is shifted 1 bit to the right. This operation is repeated each time a clock is input. When 1-byte data is accumulated in the receive register after eight transmit clocks, the content of the receive register is transferred to the receive buffer. At the same time the receive completion flag and the receive interrupt request bit are set to "1". The receive completion flag is cleared when the receive buffer register is read.
When receiving data continuously, an overrun error occurs and bit 4 of the UART transmit/receive control register 1 (overrun error flag) is set if the next receive data becomes available in the receive register while the receive completion flag is "1" (before reading the content of the receive buffer register). In this case, the next data is written in the receive buffer register. Therefore, if an overrun occurs, the transmit and receive programs must make arrangements to re-transmit the data.

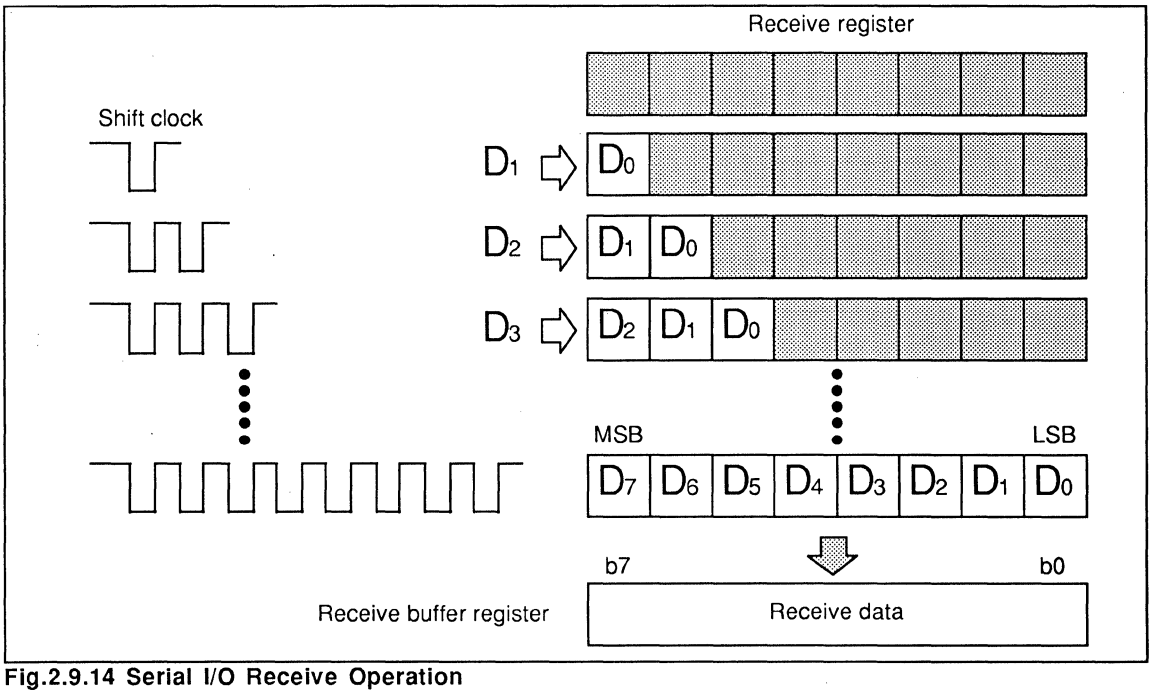


Fig.2.9.14 Serial I/O Receive Operation

[Receive interrupt]
A receive interrupt occurs and the receive interrupt request bit is set to "1" when receiving of one byte completes and the data is transferred from the receive register to the receive buffer register.
To use the receive interrupt, the interrupt priority level in the UARTi receive interrupt control register must be set to 1 or greater before an interrupt occurs. In addition, the interrupt disable flag must be cleared to "0". (See "Section 2.6 interrupts" for more information on interrupt enable conditions.) The receive interrupt request flag is cleared when an interrupt is accepted. It can also be cleared from a program.

2.9.5 Clock asynchronous serial I/O (UART)

(1)UART description

Table 2.9.5 shows the serial I/O characteristics in UART mode.

Table 2.9.5 UART Description

Parameter		Function
Data format	Start bit	1 bit
	Data bit (character length)	7 bits, 8 bits, or 9 bits
	Parity bit	0 bit or 1 bit (odd or even selectable)
	Stop bit	1 bit or 2 bit
Baud rate	Internal clock	BRG output/16
	External clock	125Kbps maximum (8MHz version) 250Kbps maximum (16MHz version)
Error detection		4 types (overrun, parity, framing, error sum) (Error sum can be used to check existence of error.)

In UART mode the baud rate and the data format must be set beforehand. The setting of the baud rate and the transfer format are described below.

(2)Transmission rate

The serial data transfer rate is determined by the baud rate (frequency of the clock used for transmission and receive). With the M37700, the baud rate is set by the BRG. The BRG is a frequency divider that consists of 8 bits. The BRG input clock can be either an internal clock or an external clock input to the CLK pin depending on the internal/external clock selection bit.

When an internal clock is selected, 1/2, 1/16, 1/64, or 1/512 of the $f(X_{IN})$ is selected with the count source selection bit. When an external clock is selected, the clock input from the CLK pin is input to the BRG.

The clock input to BRG is divided by (n+1) and then by 16 to obtain the baud rate.

Table 2.9.6 Baud Rate Selection Table

Baud Rate (bps)		Selected	Value set in BRG register	
Rated	Actual	clock	$f(X_{IN})$=8MHz	$f(X_{IN})$=16MHz
75	75.12	f_{512}	12 ($0C_{16}$)	25 (19_{16})
110	110.04	f_{64}	70 (46_{16})	141($8D_{16}$)
134.5	134.70	f_{64}	57 (39_{16})	115 (73_{16})
150	150.24	f_{64}	51 (33_{16})	103 (67_{16})
300	300.48	f_{64}	25 (19_{16})	51 (33_{16})
600	600.96	f_{64}	12 ($0C_{16}$)	25 (19_{16})
1200	1201.92	f_{16}	25 (19_{16})	51 (33_{16})
2400	2403.85	f_{16}	12 ($0C_{16}$)	25 (19_{16})
4800	4807.69	f_2	51 (33_{16})	103(67_{16})
9600	9615.39	f_2	25 (19_{16})	51 (33_{16})
19200	19230.77	f_2	12 ($0C_{16}$)	25 (19_{16})
31250	31250.00	f_2	7 (07_{16})	15 ($0F_{16}$)
62500	62500.00	f_2	3 (03_{16})	7 (07_{16})
125000	125000.00	f_2	1 (01_{16})	3 (03_{16})
250000	250000.00	f_2	0 (00_{16})	1 (01_{16})
500000	500000.00	f_2	—	0 (00_{16})

(3) Transfer format

The format of the transfer data is set in the UARTi transmit/receive mode register. In M37700 UART mode, data can be transferred in the following modes.

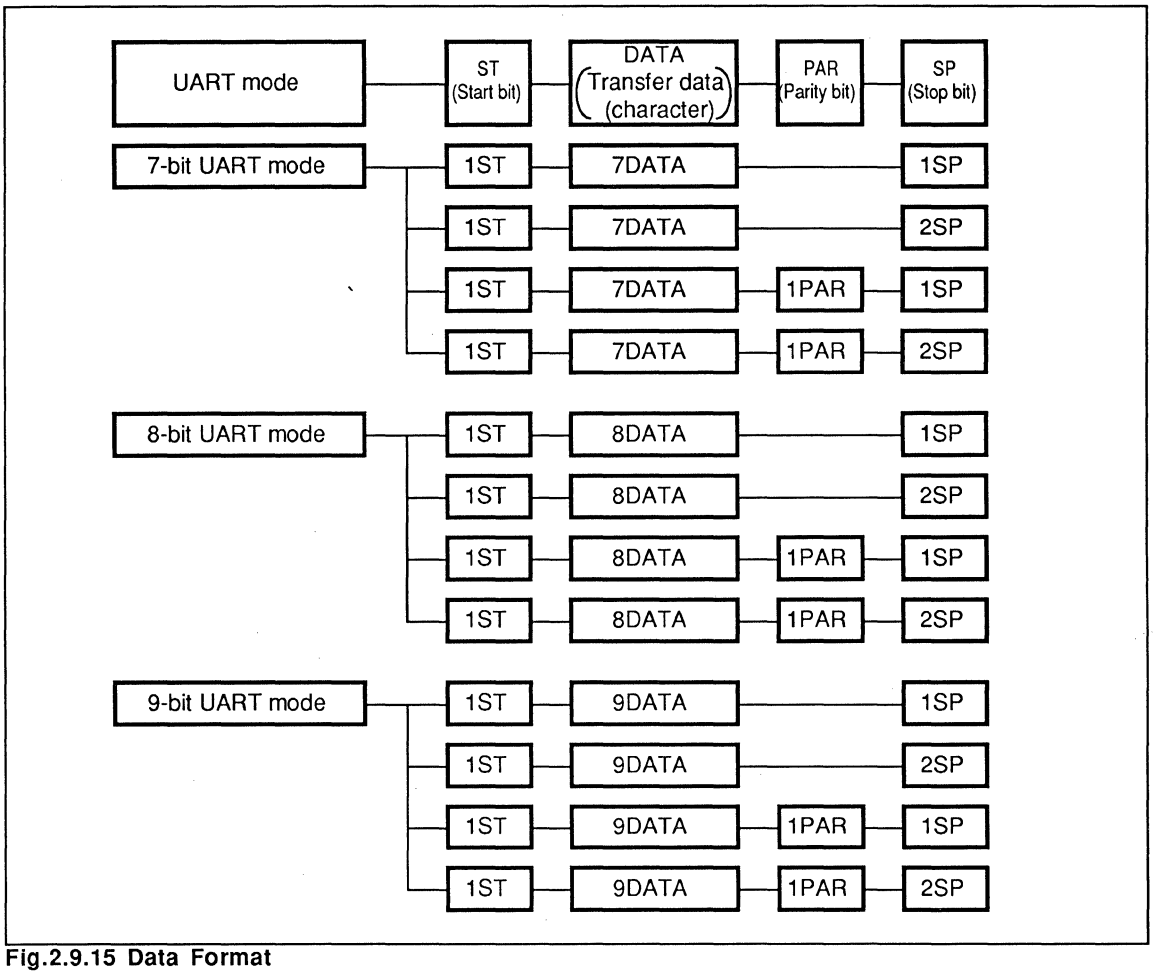


Fig.2.9.15 Data Format

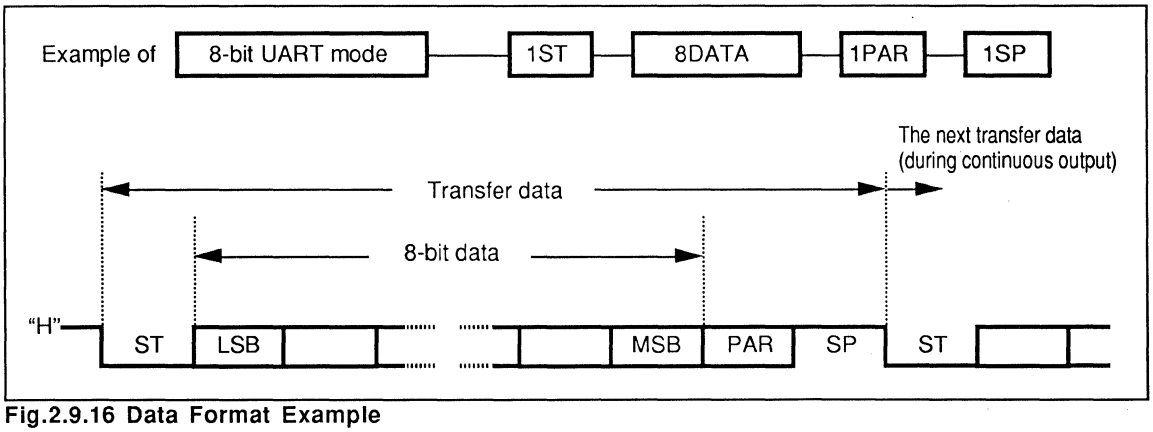


Fig.2.9.16 Data Format Example

Table 2.9.7 Transfer Data in UART Mode

Item	Function
ST (Start bit)	This bit indicates the start of data transmission. A 1 bit "L" signal is appended in front of the transmission data.
DATA (Character)	This is the transmission data written in the transmission buffer register.
SP (Stop bit)	This bit appends after the data (or after the parity bit if it is included) to indicate the end of transmission. An 1 or 2 bit "H" signal is output as a stop bit.
PAR (Parity bit)	This bit appends to the end of data to indicate the data parity. This bit is appended so that the number of 1s in the data including the parity bit is always even or odd.

(4)Serial data transmission

The UART mode data transmission method is described below.

[Setting control registers]

Set each serial I/O control register for transmission.

●Transmit/receive mode register

- Set operation mode.
  Select the data length with the serial I/O mode selection bits.
- Set transfer format.
  Select stop bit length, parity enable/disable, and odd/even parity if parity is enabled.
- Set baud rate.
  Select whether to use an internal clock ("0") or external clock ("1") as the BRG count source with the clock selection bit.
- Enable/disable sleep mode.
  (See "2.9.5 (6) Sleep mode" for detail information concerning sleep mode.)

●Transmit/receive control register 0

- Select $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function
  $\overline{\text{CTS}}$ is used when externally controlling the start of serial data transmission. One of the conditions of starting transmission is setting the $\overline{\text{CTS}}$ pin to "L" when using the $\overline{\text{CTS}}$ function.

●Transmit/receive control register 1

- Set the transmission enable flag to "1".
  After initializing the control registers for transmission, write the data to be transmitted in the low-order byte of the transmit buffer register in 7-bit or 8-bit UART mode. In 9-bit UART mode, write the data into the high-order 1 bit and the low-order 1 byte of the transmit buffer register. When the data is written, the transmission buffer register empty flag becomes "0".

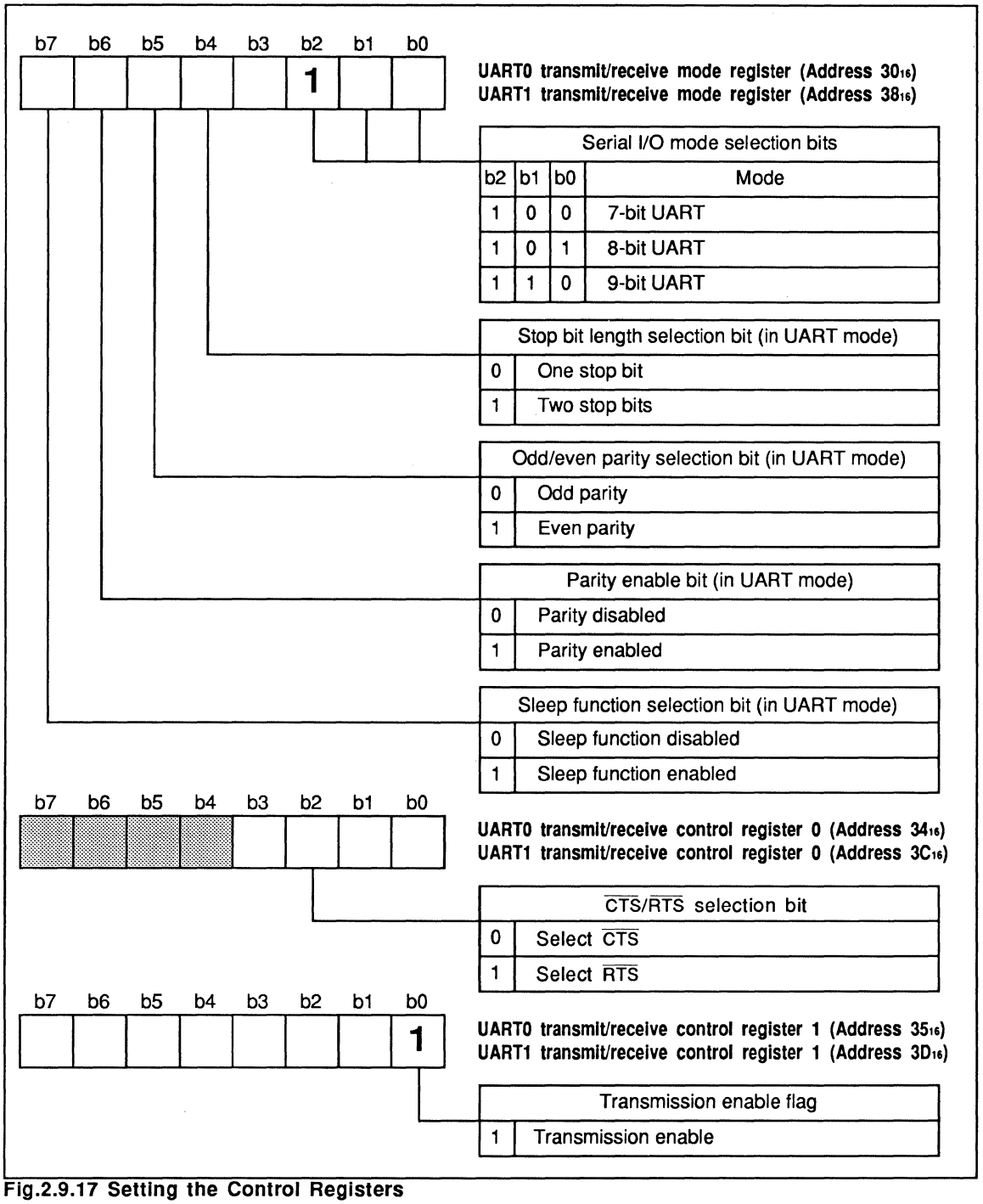


Fig.2.9.17 Setting the Control Registers

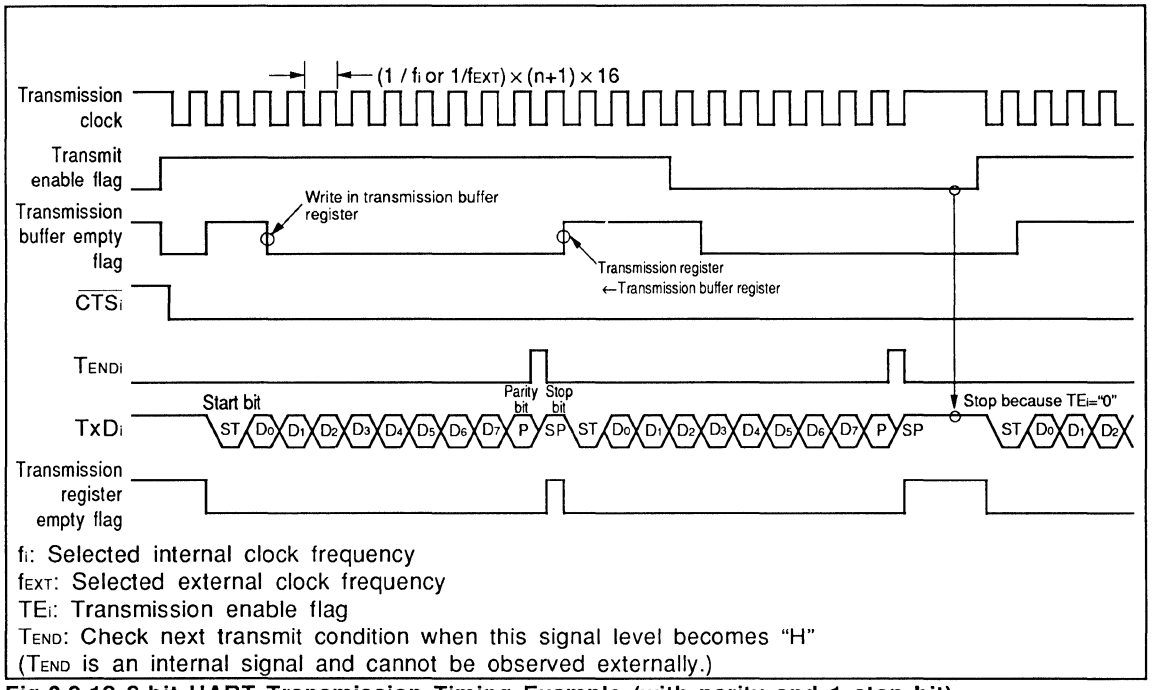
[Transmit operation]
The only difference between 7-bit UART, 8-bit UART, and 9-bit UART is the length of the transmitted data.The low-order byte of the transmssiont buffer register is used for 7-bit and 8-bit UART and the low-order byte and bit 0 of the high-order byte is used for 9-bit UART.
The transmission of serial data starts when the following conditions are satisfied.

①Transmission is enabled (transmission enable flag is "1")
②Transmit data is available in the transmission buffer (transmission buffer empty flag is "0")
③$\overline{CTS}$ pin input level is "L"
(**Note**: This condition is ignored if the $\overline{CTS}$ function is not selected.)

When the above three conditions are satisfied (two if $\overline{CTS}$ function is not selected), the content of the transmission buffer register is transferred to the transmission register and data transmission starts from the TxD pin. At this point, bit 1 of the transmit/receive control register 1 (transmission buffer empty flag) is set to "1" and bit 3 of the transmit/receive control register 0 (transmission register empty flag) is cleared to "0".
When transmission starts, data is output from the TxD pin in the format specified by the transmit/receive mode register. The data is output bit by bit in the order;
ST→DATA(LSB)→...→DATA(MSB)→PAR→SP.

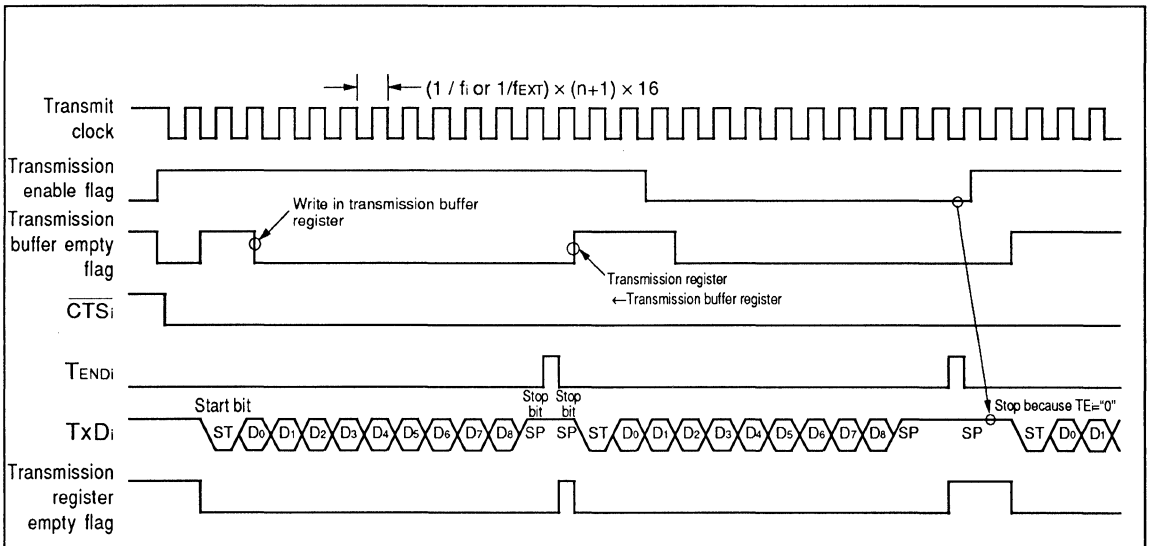
After the stop bit has been output, the transmission register empty flag is set to "1" to indicate that the transmission has completed. If the next data is available when transmission completes, a start bit is generated following the stop bit and the next data is transmitted. In order to continuously transfer data, the next transmission data should be written in the transmission buffer register during the transmit operation (when the transmission register empty flag is "0"). If the transmit condition for the next data is not satisfied, "H" level is output from the TxD pin.

[Transmission interrupt]
The transmission interrupt request bit is set to "1" when the content of the transmission buffer is transferred to the transmission register. To use the transmission interrupt, the interrupt priority level in the UARTi transmission interrupt control register must be set to 1 or greater before an interrupt occurs. In addition, the interrupt disable flag must be cleared to "0". (See "Section 2.6 interrupts" for more information on interrupt enabling conditions.) The transmission interrupt request flag is cleared when an interrupt is accepted. It can also be cleared from a program.



fi: Selected internal clock frequency
fEXT: Selected external clock frequency
TEi: Transmission enable flag
TEND: Check next transmit condition when this signal level becomes "H"
(TEND is an internal signal and cannot be observed externally.)

Fig.2.9.18 8-bit UART Transmission Timing Example (with parity and 1 stop bit)



fi: Selected internal clock frequency
fEXT: Selected external clock frequency
TEi: Transmission enable flag
TEND: Check next transmit condition when this signal level becomes "H"
(TEND is an internal signal and cannot be observed externally.)

Fig.2.9.19 9-bit UART Transmission Timing Example (no parity and 2 stop bits)

(5)Receiving serial data

The UART mode data receiving method is described below.

[Setting control registers]

●Transmit/receive mode register

Match the format with the transmitting side.

•Set operation mode.

Select the data length with the serial I/O mode selection bits.

•Set transfer format.

Select stop bit length, parity enable/disable, and odd/even parity if parity is enabled.

•Set baud rate.

Select whether to use an internal clock ("0") or external clock ("1") as the BRG count source with the clock selection bit.

•Enable/disable sleep mode.

(See "2.9.5 (6) Sleep mode" for detail information concerning sleep mode.)

●Transmit/receive control register 0

•Select $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function.

$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ selection bit

"0": $\overline{\text{CTS}}$ function selected

"1": $\overline{\text{RTS}}$ function selected

The $\overline{\text{RTS}}$ function is used to indicate externally that data can be received.

●Transmit/receive control register 1

•Set the receive enable flag to "1" (receive enabled).

[Receive operation] (when using an external clock)

Serial data receive is enabled by setting the receive enable flag to "1". When the receive enable flag is set to "1", the $\overline{\text{RTS}}$ pin level becomes "L" to indicate externally that the microprocessor is ready to receive serial data (when $\overline{\text{RTS}}$ function is selected). The transmit and receive timing can be synchronized by connecting the $\overline{\text{RTS}}$ output to the $\overline{\text{CTS}}$ pin on the transmit side.

When the RxD pin detects a start bit, a receive clock is generated and data receive starts. At the same time the $\overline{\text{RTS}}$ pin level returns to "H". The RxD pin level is received in the receive register at the rising edge of the receive clock and the content of the receive register is shifted 1 bit to the right. This operation is repeated to receive the entire data from ST to SP. Then the content of the receive register is transferred to the receive buffer register. At the same time the receive completion flag and the receive interrupt request bit are set. The receive completion flag is cleared when the receive buffer register is read.

When receiving data continuously, an overrun error occurs and bit 4 of the UART control register 1 (overrun error flag) is set if the next receive data becomes available in the receive register while the receive completion flag is "1" (before reading the content of the receive buffer register). In this case, the next data is written in the receive buffer register. Therefore, if an overrun occurs, the transmit and receive programs must make arrangements to re-transmit the data.

[Receive interrupt]

A receive interrupt occurs and the receive interrupt request bit is set to "1" when a stop bit indicating the end of data is detected.

To use the receive interrupt, the interrupt priority level in the UARTi receive interrupt control register must be set to 1 or greater before an interrupt occurs. In addition, the interrupt disable flag must be cleared to "0". (See "Section 2.6 interrupts" for more information on interrupt enable conditions.)

The receive interrupt request flag is cleared when an interrupt is accepted. It can also be cleared from a program.

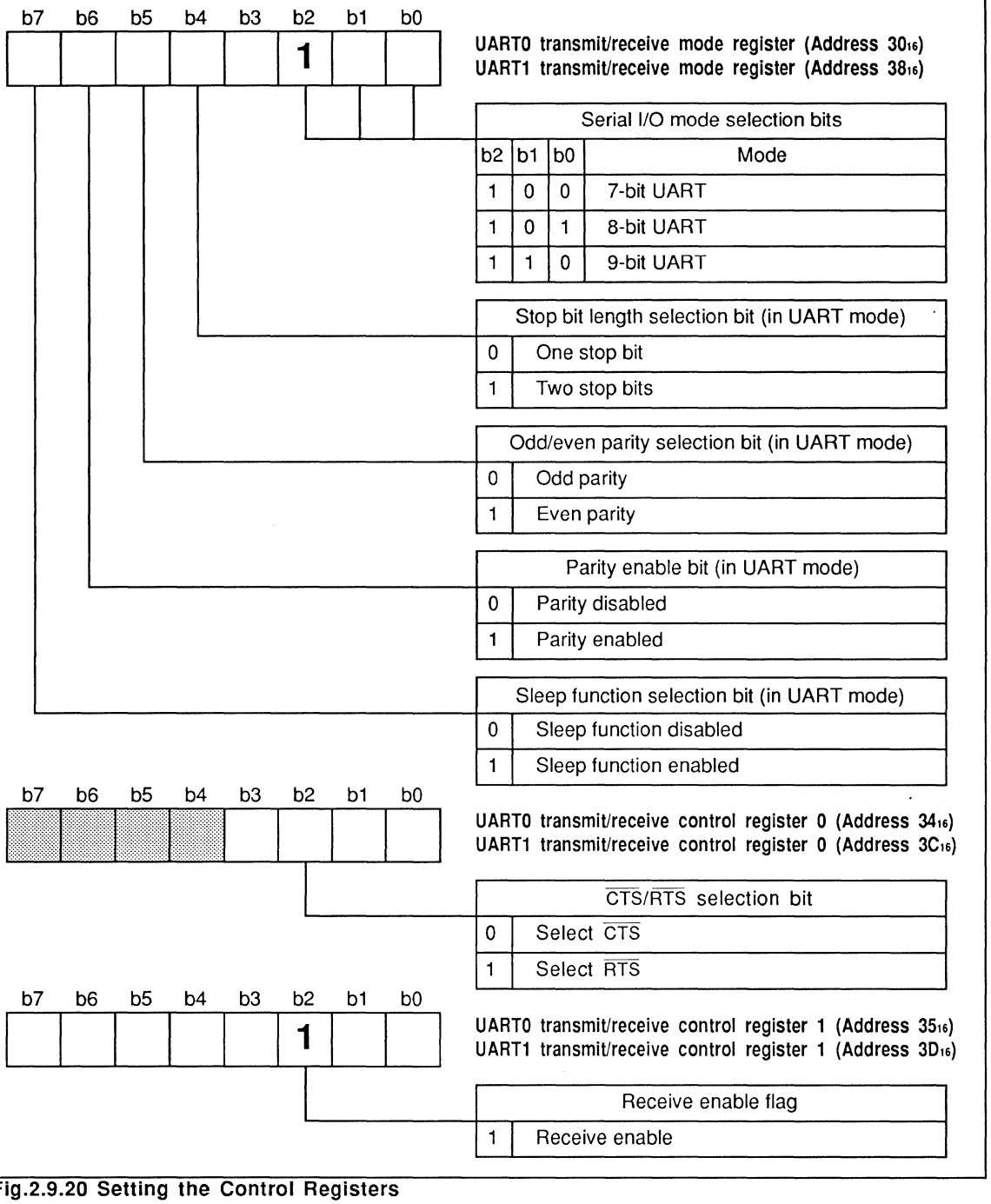


Fig.2.9.20 Setting the Control Registers

[Error flag]
During UART mode operation, transmission data errors can be detected using four error flags. These errors are detected when transferring data from the receive register to the receive buffer register. The error flags are cleared when the low-order byte of the receive buffer register is read or when the receive enable flag is set to "0".

●Overrun error
An overrun error occurs and the overrun error flag is set to "1" when the next receive data becomes available before the content of the receive buffer register is read.

●Framing error
A framing error occurs and the framing error flag is set to "1" when there is insufficient number of stop bits.

●Parity error
A parity error occurs and the parity error flag is set to "1" when parity checking is enabled and the number of 1s in the data including the parity bit conflicts with the parity specified by bit 5 of the UARTi transmit/receive mode register.

●Sum error
The error sum flag is set to "1" when either an overrun error, a framing error, or a parity error occurs. The existence of errors can be determined by checking the error sum flag.

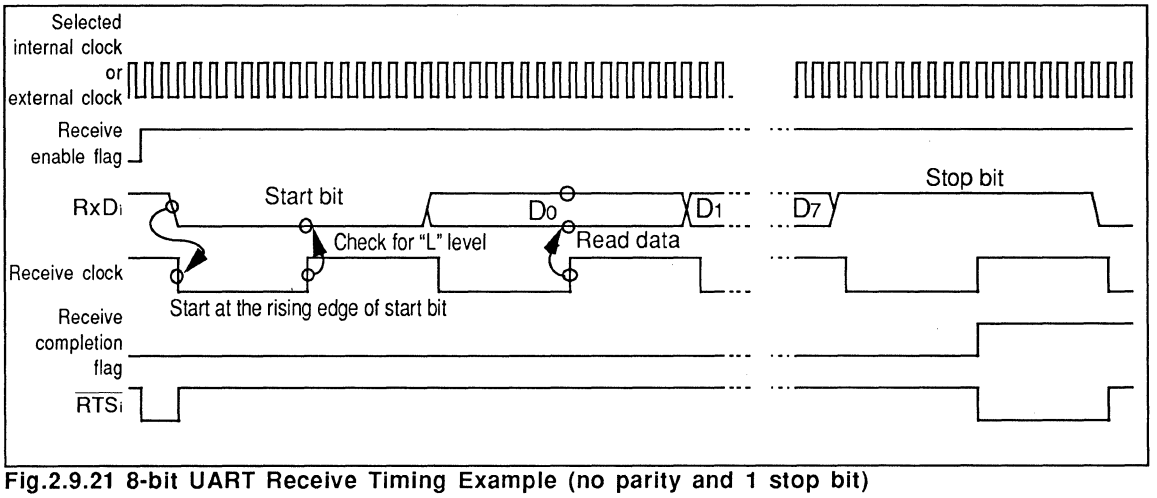


Fig.2.9.21 8-bit UART Receive Timing Example (no parity and 1 stop bit)

(6) Sleep mode

Sleep mode is used for communication between certain computers when multiple microcomputers are connected through serial I/O.

Sleep mode is entered by setting the UARTi transmit/receive mode register bit 7 to "1". In sleep mode, the content of the receive register is not transferred to the receive buffer register when the most significant bit (bit 7 if 8-bit UART mode, bit 6 if 7-bit UART mode, and bit 8 if 9-bit UART mode) of the received data is "0". In this case, the receive completion flag and the error flags remain unchanged and no receive interrupt occurs. Normal receive operation is performed only when the most significant bit of the received data is "1".

The following is a description of sleep mode usage in 8-bit UART mode. The main microcomputer first sends a data with bit 7 set to "1" and the remaining bits 0~6 forming the address of the destination microcomputer. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data and sets the sleep selection bit to "0" if the address matches its own address and to "1" if otherwise. Next the main microcomputer starts sending data with bit 7 set to "0". Then only the microcomputer with the sleep selection bit set to "0" will receive this data. This enables communication between the main microcomputer and a specific subordinate microcomputer.

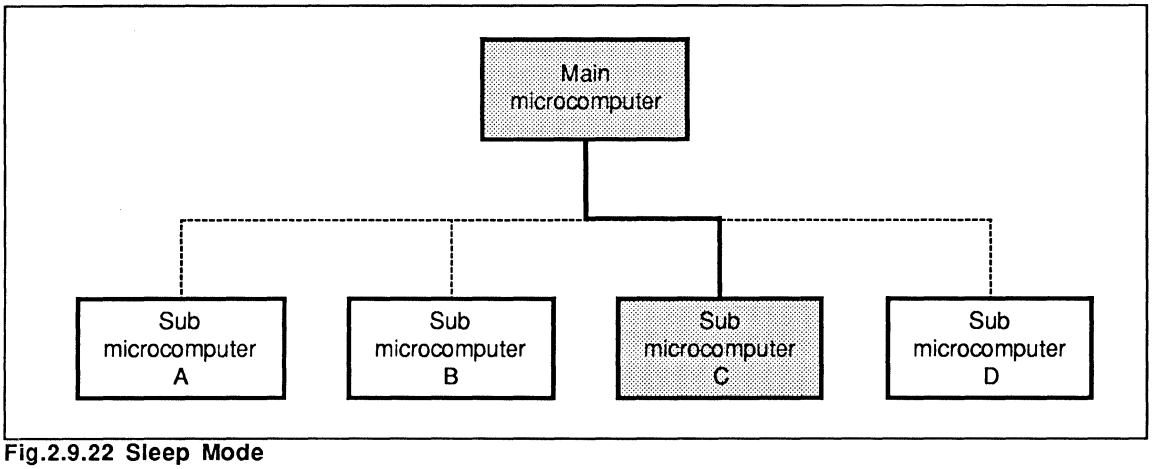


Fig.2.9.22 Sleep Mode

2.10 A-D Converter

2.10.1 A-D converter description

Table 2.10.1 describes the characteristics of the A-D converter and Figure 2.10.1 show its block diagram.

Table 2.10.1 A-D Converter Characteristics

Parameter	Description
Analogi Input pin	8 pins (AN_0 to AN_7)
A-D conversion mode	One-shot mode
	Repeat mode
	Single sweep mode
	Repeat sweep mode
A-D conversion method	Successive approximation
Resolution	8 bits
Absolute accuracy	±3 LSB
Conversion speed	57ϕ_{AD} cycles, ϕ_{AD}: A-D converter operating clock (for 1 analog input pin)

The M37700 A-D converter provides the following four A-D conversion modes.

●One-shot mode

The input voltage to the selected analog input pin is converted. After conversion, the result is stored in the corresponding A-D register and an A-D conversion interrupt is occurred.

●Repeat mode

The input voltage to the selected analog input pin is repeatedly converted. The results are stored in the corresponding A-D register, but no A-D conversion interrupt is occurred.

●Single sweep mode

Inputs to analog input pins AN_0, AN_1,...AN_7 are converted in this order and an A-D conversion interrupt is occurred when conversion of AN_7 completes. The result is stored in the corresponding A-D register when each pin is converted.

●Repeat sweep mode

This is similar to single sweep mode except that conversion is repeated without occurring an interrupt after converting the AN_7 pin.

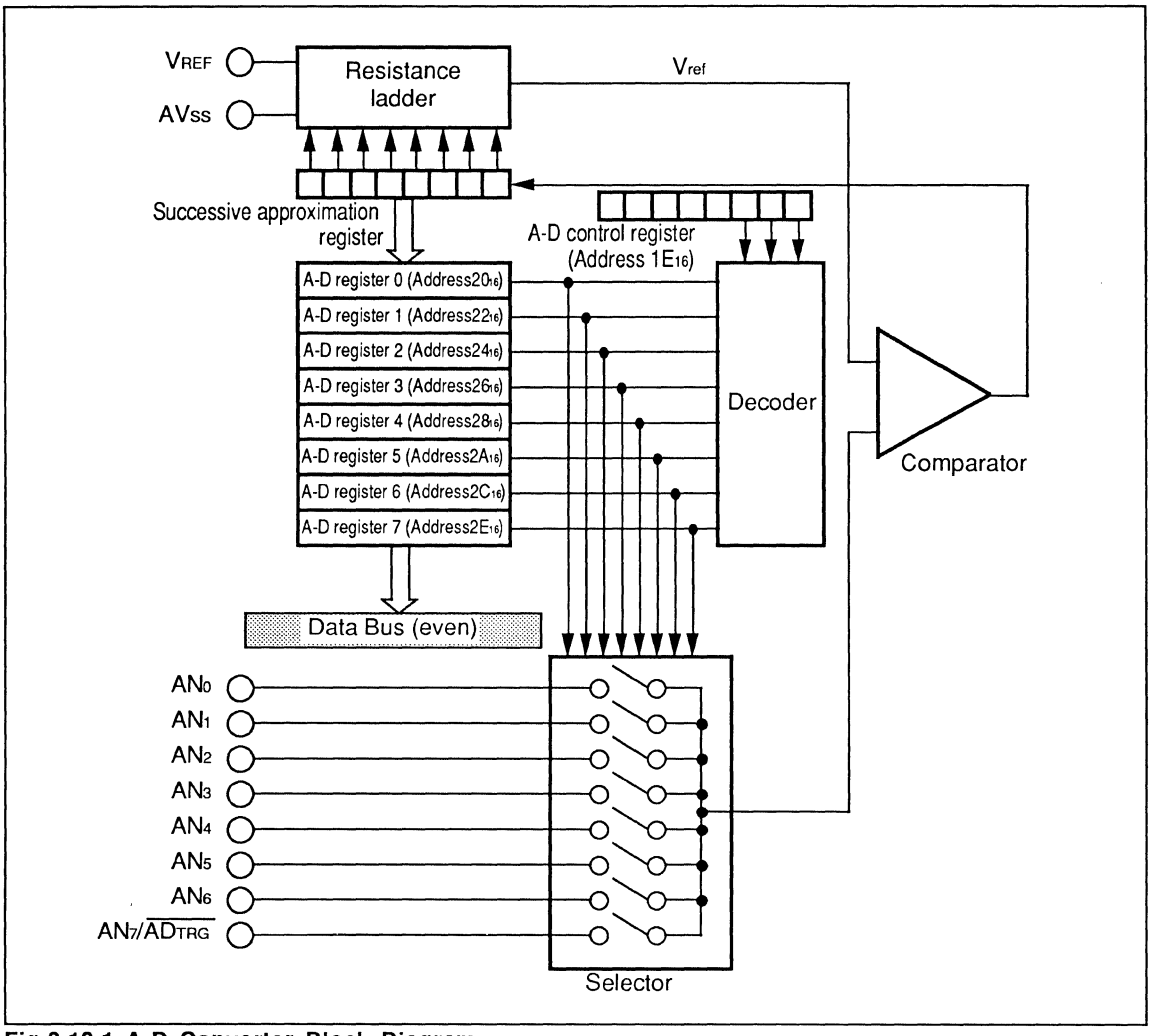
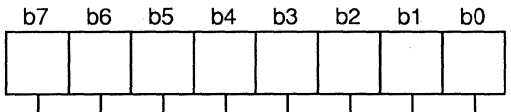


Fig.2.10.1 A-D Converter Block Diagram

2.10.2 Block description

(1)A-D control register

The A-D control register is used to control the A-D converter.



A-D control register (Address $1E_{16}$)

Analog input selection bits			
b2	b1	b0	Selected port
0	0	0	Select AN_0
0	0	1	Select AN_1
0	1	0	Select AN_2
0	1	1	Select AN_3
1	0	0	Select AN_4
1	0	1	Select AN_5
1	1	0	Select AN_6
1	1	1	Select AN_7 (Note)

A-D mode selection bits		
b4	b3	A-D conversion mode
0	0	One-shot mode
0	1	Repeat mode
1	0	Single sweep mode
1	1	Repeat sweep mode

Trigger selection bit	
0	Software trigger (internal trigger)
1	$\overline{AD_{TRG}}$ input trigger (external trigger)

A-D conversion start flag	
0	Stop A-D conversion
1	Start A-D conversion

A-D conversion frequency (ϕ_{AD}) selection flag	
0	Select $f(X_{IN})/8$
1	Select $f(X_{IN})/4$

Note: If an external trigger is used, pin AN_7 cannot be used as analog voltage input pin.

Fig.2.10.2 A-D Control Register Bit Configuration

●Analog input selection bits

This bit is used to select the analog input pin in one-shot mode and repeat mode. These bits are ignored in other modes.

Precautions when selecting analog pin

The analog input pin is in common with programmable I/O port P7. Therefore, when using a pin as analog input pin and $\overline{ADTRG}$ input pin, the corresponding bit in the port 7 data direction register must be set to "0" (input mode). Note that pin AN_7 cannot be used as analog input pin when using an external trigger.

●A-D mode selection bits

The A-D mode selection bits are used to select among the four A-D conversion modes.

Table 2.10.2 A-D Mode Selection Bit

b4	b3	A-D mode selection bit
0	0	One-shot mode
0	1	Repeat mode
1	0	Single sweep mode
1	1	Repeat sweep mode

●Trigger selection bit

An A-D conversion operation is started by a trigger. The trigger selection bit is used to select between an internal trigger and an external trigger. An internal trigger (software trigger) is selected when this bit is "0" and an external trigger ($\overline{ADTRG}$ pin input signal) is selected when this bit is "1".

<Internal trigger>

A trigger is generated and A-D conversion starts when bit 6 (A-D conversion start flag) of the A-D control register is set to "1".

<External trigger>

A trigger is generated when the signal input to the $\overline{ADTRG}$ pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is "1". When an external trigger is selected, a retrigger can be available during A-D conversion. In this case, the conversion is repeated from the beginning.

Precautions when using an external trigger

The $\overline{ADTRG}$ pin is common with pin $P7_7/AN_7$. Therefore, the AN_7 pin cannot be used as analog input pin when external trigger is selected in one-shot mode or repeat mode. In addition, when external trigger is selected, the data direction register bit (bit 7 at address 11_{16}) for the $P7_7$ pin must be set to input mode ("0").

●A-D conversion start flag

The A-D conversion start flag can be used to start or stop A-D conversion.

<Internal trigger>

An internal trigger is generated and A-D conversion starts when the A-D conversion start flag is set to "1". A-D conversion stops when it is cleared to "0". This bit is automatically cleared after A-D conversion in one-shot mode and single sweep mode. It is not cleared in other modes.

<External trigger>

The A-D conversion start flag must be set to "1" before generating an external trigger. If external trigger is selected, this flag is not cleared after conversion.

NEW REGISTER [illegible] (Addr 1Fh)

[illegible] channels in sweep mode

00 An0, An1
01 An0 – An3
10 An0 – An5
11 An0 – An7

●A-D conversion Frequency (ϕ_{AD}) selection flag

This flag is used to select the A-D converter operating frequency (ϕ_{AD}). When this flag is "0", the clock frequency $f(X_{IN})$ divided by 8 is selected. When this flag is "1", the clock frequency $f(X_{IN})$ divided by 4 is selected.

In one-shot mode and repeat mode, A-D conversion completes after 57 × ϕ_{AD} cycles from the beginning of A-D conversion. In single-sweep and repeat sweep mode, A-D conversion of the AN_7 pin completes after 456 × ϕ_{AD} cycles from the beginning of A-D conversion.

Table 2.10.3 A-D Converter Operation Frequency and Conversion Time

Frequency selection flag		"0"	"1"
A-D converter operating clock		$\phi_{AD}=f(X_{IN})/8$	$\phi_{AD}=f(X_{IN})/4$
Conversion time (Note)	$f(X_{IN})$=8MHz	57.0μs	28.5μs
	$f(X_{IN})$=16MHz	28.5μs	14.25μs

Note: Conversion time per analog input pin

Precautions when selecting ϕ_{AD}

The A-D converter operating clock ϕ_{AD} during A-D conversion must be no less than 250kHz because the comparator in the A-D conversion circuit consists of capacity coupling amplifiers.

(2) A-D register i (i=0 to 7)

The A-D registers are 8-bit read only registers. The conversion results are stored in these registers. There are eight A-D registers numbered from 0 to 7 with each corresponding to an analog input pin. The content of the A-D register can be read during A-D conversion. However, if the A-D register corresponding to the analog pin being converted is read, the previous conversion result is obtained.

Table 2.10.4 Combination between Analog Input Pin and Register Containing the Result

Analog input pin	Register containing the result	Address
AN_0	A-D register 0	20_{16}
AN_1	A-D register 1	22_{16}
AN_2	A-D register 2	24_{16}
AN_3	A-D register 3	26_{16}
AN_4	A-D register 4	28_{16}
AN_5	A-D register 5	$2A_{16}$
AN_6	A-D register 6	$2C_{16}$
AN_7	A-D register 7	$2E_{16}$

(3)Comparator and successive approximation register

When A-D conversion is triggered, the following operation starts and an analog value is converted to a digital value.

①Initialization of successive approximation register

The successive approximation register is cleared to "00_{16}"

②Setting the most significant bit (bit 7)

The successive approximation register bit 7 is set to "1". Then the reference voltage V_{ref} is compared with the input voltage V_{IN} and bit 7 changes as follows:

Unchanged if $V_{ref} < V_{IN}$
Cleared to "0" if $V_{ref} > V_{IN}$

Note : The compare reference voltage V_{ref} depends on the value in the successive approximation register. Table 2.10.5 shows the relationship between V_{ref} and the value in the successive approximation register.

Step ② above is repeated for all bits from bit 7 to bit 0 and the value in the successive approximation register (digital equivalent of the analog input voltage) is stored in the A-D register when comparison of bit 0 completes.

Table 2.10.5 Relationship between the Content of the Successive Approximation Register and V_{ref}

Content of successive approximation register	0	1~255
Comparison reference voltage V_{ref} (V)	0	$V_{REF}/256 \times (n-0.5)$ n is the content of the successive approximation register

Table 2.10.6 Change in Successive Approximation Register and Compare Voltage During A-D Conversion

	Successive approximation register	Compare voltage V_{REF}
	b7 b0	
Conversion start	0 0 0 0 0 0 0 0	0 [V]
First comparison	1 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{512}$ [V]
Second comparison	n_7 1 0 0 0 0 0 0 1st comparison result	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{512}$ [V]
Third comparison	n_7 n_6 1 0 0 0 0 0 2nd comparison result	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{512}$ [V]
Eighth comparison	n_7 n_6 n_5 n_4 n_3 n_2 n_1 1	
Conversion end	n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0	

2.10.3 A-D conversion mode operation

Four different A-D conversion modes can be selected with A-D mode selection bits. In each mode, the trigger selection bit is used to determine whether to use a software trigger (internal trigger) or an external input signal (external trigger).
Each conversion mode is described below for case using an internal trigger and an external trigger.

(1)One-shot mode [00]*

The input voltage to the analog input pin selected with the analog input pin selection bit of the A-D control register is converted and an A-D conversion interrupt is generated when conversion completes. In this mode, the analog input pin must be selected before the A-D conversion trigger. The pins not used as analog input pin can be used as normal I/O ports.

●When an internal trigger is used to start A-D conversion

When the A-D start flag is set to "1", an internal trigger is generated and A-D conversion starts. After 57 cycles of ϕ_{AD}, A-D conversion ends, the content of the successive approximation register (converted result) is transferred to the A-D register, and the A-D interrupt request bit is set to "1" (→ generate A-D interrupt request). Then the A-D start flag is cleared to "0" and A-D converter stops.

●When an external trigger is used to start A-D conversion

A-D conversion starts when the A-D start flag is set to "1" and the input level of the $\overline{ADTRG}$ pin changes from "H" to "L" (→ external trigger). When A-D conversion completes after 57 cycles of ϕ_{AD}, the content of the successive approximation register (converted result) is transferred to the A-D register and the A-D interrupt request bit is set to "1" (→ generate A-D interrupt request). At this point, the A-D start flag is not cleared and A-D conversion can be repeated by generating another trigger. A trigger can also be generated during A-D conversion.

Precautions when using an external trigger

When an external trigger is selected, the AN_7 pin must not be selected as an analog input pin because it is used as the trigger input pin ($\overline{ADTRG}$) to the A-D converter.

(2)Repeat mode [01]*

In this mode, the input voltage to the analog input pin selected with the analog input selection bit of the A-D control register is repeatedly converted. No interrupt request is generated and the A-D conversion flag is not cleared. The conversion of the selected pin is repeated while the A-D start flag is "1". The A-D conversion result can be read at any time. In this mode, the analog input pin must be selected before the A-D trigger is received. Pins not used as analog input pin can be used as normal I/O ports.

●When an internal trigger is used to start A-D conversion

When the A-D start flag is set to "1", an internal trigger is generated and A-D conversion starts. Each time a conversion is completed, the content of the successive approximation register (converted result) is transferred to the A-D register. The A-D converter does not stop at this point and conversion is repeated.

●When an external trigger is used to start A-D conversion

A-D conversion starts when the A-D start flag is set to "1" and the $\overline{ADTRG}$ pin input level changes from "H" to "L" (→ external trigger). When A-D conversion completes, the content of the successive approximation register (conversion result) is transferred to the A-D register. The A-D converter does not stop at this point and conversion is repeated.

Precautions when using an external trigger

When an external trigger is selected, the AN_7 pin cannot be used as analog input pin because it is used as the trigger input pin ($\overline{ADTRG}$) to the A-D converter.

(3)Single sweep mode [10]*

In this mode, inputs from pins AN_0 to AN_7 are converted and the results are stored in the respective A-D register. An A-D interrupt occurs when the result of converting pin AN_7 is stored in the A-D register. In this mode, the data direction register for port P7 must be set to "00_{16}" (input mode) before an A-D trigger is received.

●When an internal trigger is used to start A-D conversion

When the A-D start flag is set to "1", an internal trigger is generated and A-D conversion of pin AN_0 starts. When conversion of pin AN_0 ends, the result is stored in A-D register 0. Then pin AN_1 is converted and the result is stored in A-D register 1. This is repeated up to pin AN_7. When the result of converting pin AN_7 is stored in A-D register 7, the A-D interrupt request bit is set to "1" (→ generate A-D interrupt request). At this point the A-D start flag is cleared and the A-D converter stops.

●When an external trigger is used to start A-D conversion

Pins AN_0 to AN_7 are converted when the A-D start flag is set to "1" and the $\overline{ADTRG}$ pin input level changes from "H" to "L" (→ external trigger). The result is stored in the corresponding A-D register each time a pin is converted and an A-D interrupt request bit is set to "1" when conversion of pin AN_7 completes. At this point the A-D start flag is not cleared to "0". Therefore, conversion can be repeated from pin AN_0 by generating another trigger. A trigger can also be generated during A-D conversion.

Precautions when using an external trigger

When an external trigger is selected, the result of converting the trigger input is stored in A-D register 7 because pin AN_7 is used as the trigger input pin ($\overline{ADTRG}$) to the A-D converter.

(4)Repeat sweep mode [11]*

Inputs from pins AN_0 to AN_7 are converted and stored in the respective A-D register as with single sweep mode. However, conversion does not stop after converting all pins. Instead, it is repeated from pin AN_0. In this mode, the data direction register of port P7 must be set to "00_{16}" (input mode) before the A-D trigger is received.

●When an internal trigger is used to start A-D conversion

When the A-D start flag is set to "1", an internal trigger is generated and A-D conversion starts. The content of the successive approximation register (conversion result) is transferred to the A-D register each time a pin (AN_0 to AN_7) is converted. This is repeated until the A-D start flag is cleared to "0".

●When an external trigger is used to start A-D conversion

A-D conversion starts when the A-D start flag is set to "1" and the $\overline{ADTRG}$ pin input level changes from "H" to "L" (→ external trigger). The content of the successive approximation register (conversion result) is transferred to the A-D register each time a pin (AN_0 to AN_7) is converted. Conversion is repeated until the A-D start flag is cleared to "0".

Precautions when using an external trigger

When an external trigger is used, the result of converting the trigger input is stored in A-D register 7 because the AN_7 pin is used as the trigger input pin ($\overline{ADTRG}$) to the A-D converter.

* The numbers in brackets are the contents of the A-D mode selection bits.

2.11 Watchdog Timer

2.11.1 Watchdog timer description

The watchdog timer is a 12-bit timer that is used to detected unexpected execution sequence caused by software run-away. It is also used to stabilize the oscillator when returning from a **STP** instruction. Figure 2.11.1 shows a block diagram of the watchdog timer.

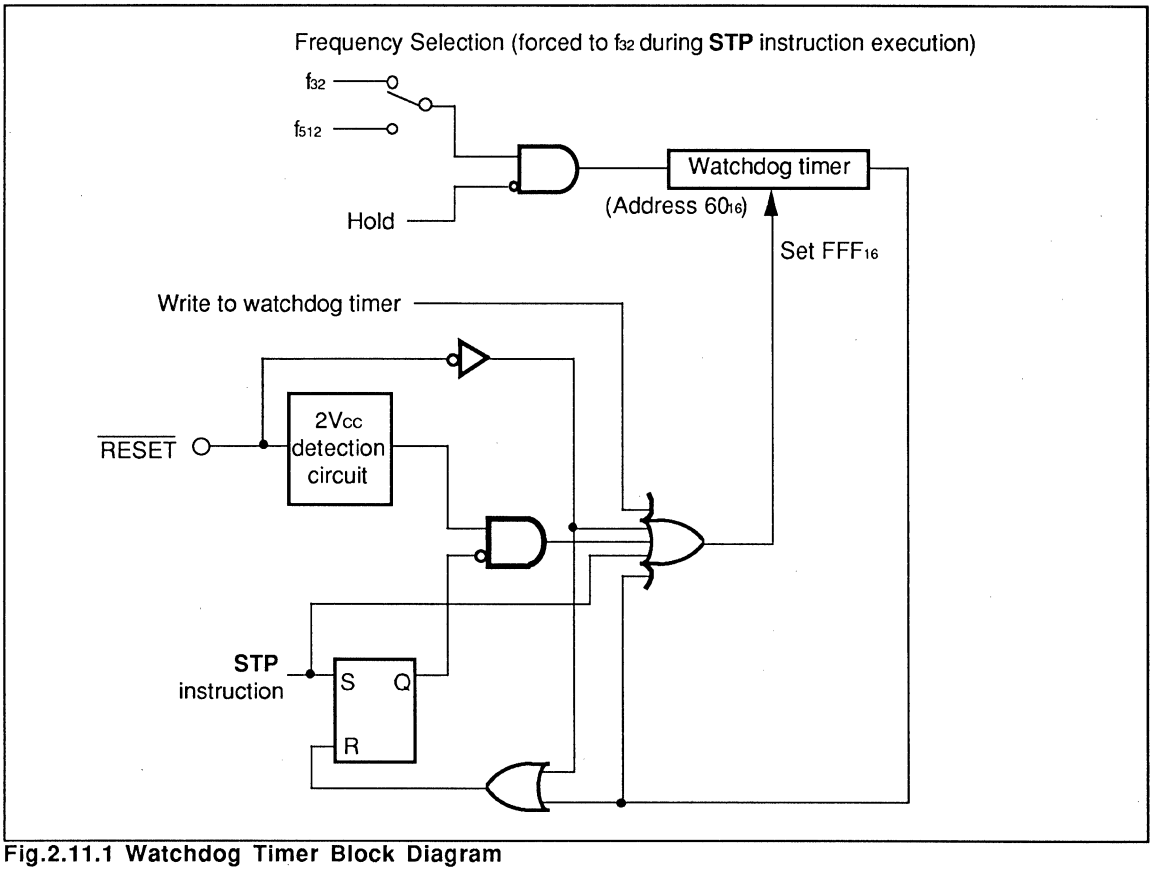


Fig.2.11.1 Watchdog Timer Block Diagram

2.11.2 Operation description

The watchdog timer consists of 12 bits and its content is decremented (−1) each time the clock selected with the watchdog timer frequency selection bit is input to the watchdog timer. The watchdog timer frequency selection bit is assigned to bit 0 at address 61_{16} and f_{512} (source oscillating frequency $f(X_{IN})$/512) is selected as the watchdog timer count source after a reset. Thereafter, it can also be set to f_{32} (source oscillating frequency $f(X_{IN})/32$) by changing the watchdog timer frequency selection flag by program.

When there is a reset, "FFF_{16}" is set in the watchdog timer. Then the count source f_{512} is counted. The content of the watchdog timer is decremented each time a clock is input. An interrupt is raised when the most significant bit of the watchdog timer becomes "0" after 2048 counts. A watchdog timer interrupt is a non-maskable interrupt with the highest priority.

An arbitrary value cannot be set in the watchdog timer. A value "FFF_{16}" is automatically set in the watchdog timer when there is a reset, when an **STP** instruction is executed, or when a dummy data is written in the watchdog timer (address 60_{16}). Address 60_{16} is a write only register and its content cannot be read.

In order to stop the watchdog timer (disable its function), a voltage twice the V_{CC} voltage must be applied to the $\overline{RESET}$ pin. During this time, the watchdog timer stops with "FFF_{16}" set. Also, while the $\overline{HOLD}$ pin is "L" level (hold state) in memory expansion or microprocessor mode, the watchdog timer goes into a hold state and clock input to the watchdog timer is prohibited.

In addition to detecting program run-away, the watchdog timer is also used as a return timer from a stop mode (halting of oscillating circuit with the **STP** instruction). When a **STP** instruction is executed, the watchdog timer count source is forced to f_{32} and "FFF_{16}" is set in the watchdog timer. Then when the watchdog timer is started with an external interrupt, a watchdog timer interrupt occurs and an internal clock ø is supplied. This is because some time is required for the oscillator to stabilize. See "Section 4.2 Clock Generation Circuit" for more detail concerning the stop mode.

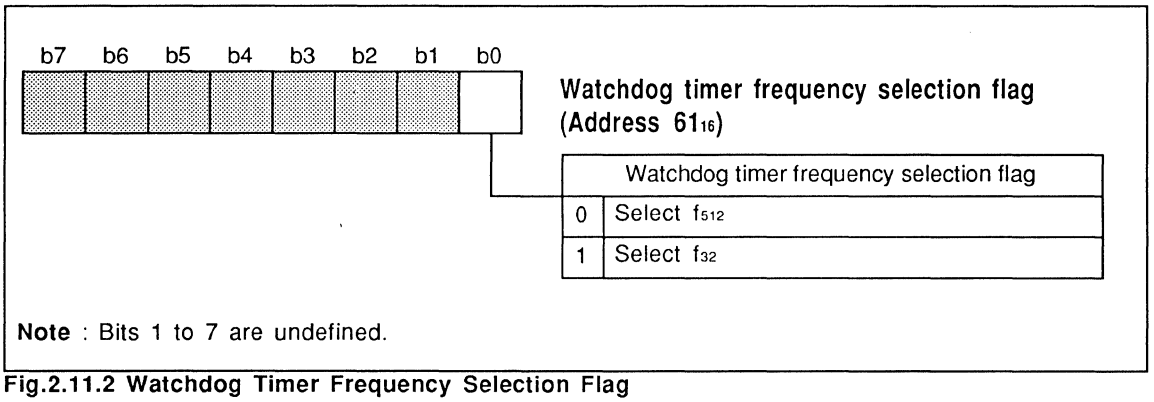


Fig.2.11.2 Watchdog Timer Frequency Selection Flag

2.11.3 Watchdog timer usage

When using the watchdog timer to detect program run-away, the program must write to the watchdog timer before its most significant bit becomes "0". Then if this code is not executed due to program run-away, the most significant bit of the watchdog timer becomes "0" and an interrupt occurs. Thereafter, the control should be passed to the interrupt service routine.

To restart from reset after detecting a program run-away, bit 3 of the processor mode register (software reset bit) must be set to "1" in the watchdog timer interrupt service routine. In this way, a run-way program can be automatically reset and returned to normal routine.

MEMO

CHAPTER 3
RESET

3.1 Reset

3.1.1 Reset operation

The CPU is reset when "L" level is applied to the $\overline{\text{RESET}}$ pin when the supply voltage is 5V±10%. The reset is deactivated and a program starts from address formed by using the contents at bank 0 address $FFFF_{16}$ as the high-order address and address $FFFE_{16}$ as the low-order address, when the $\overline{\text{RESET}}$ pin is returned to "H" level after "L" input is applied for a sufficient duration (approximately 10ms) that is the oscillator requires time to stabilize such as when cancelling stop mode with a **STP** instruction.

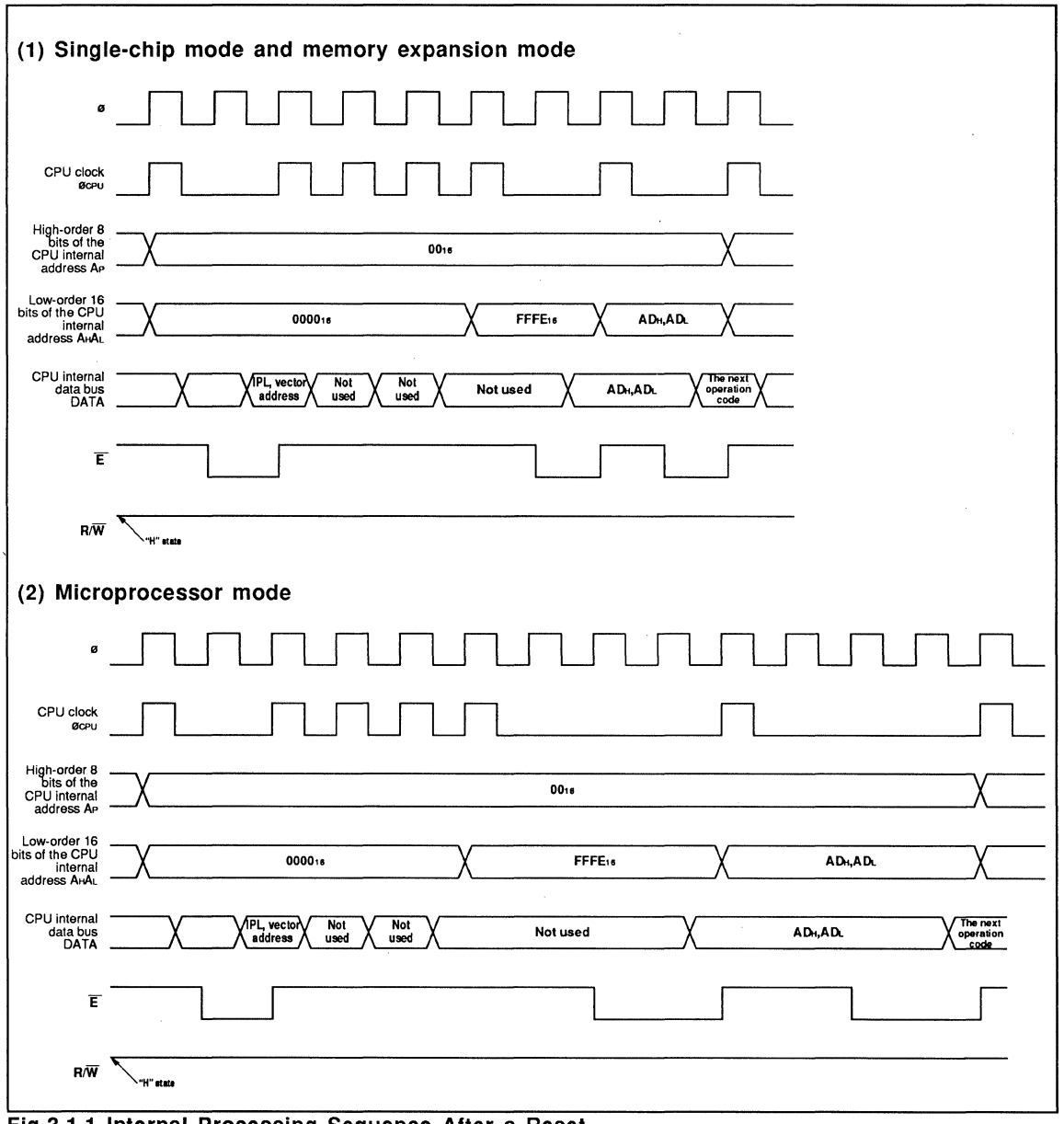


Fig.3.1.1 Internal Processing Sequence After a Reset

3.1.2 Internal status at a reset

Figure 3.1.2 (a) to (c) show the contents of internal registers immediately at a reset.

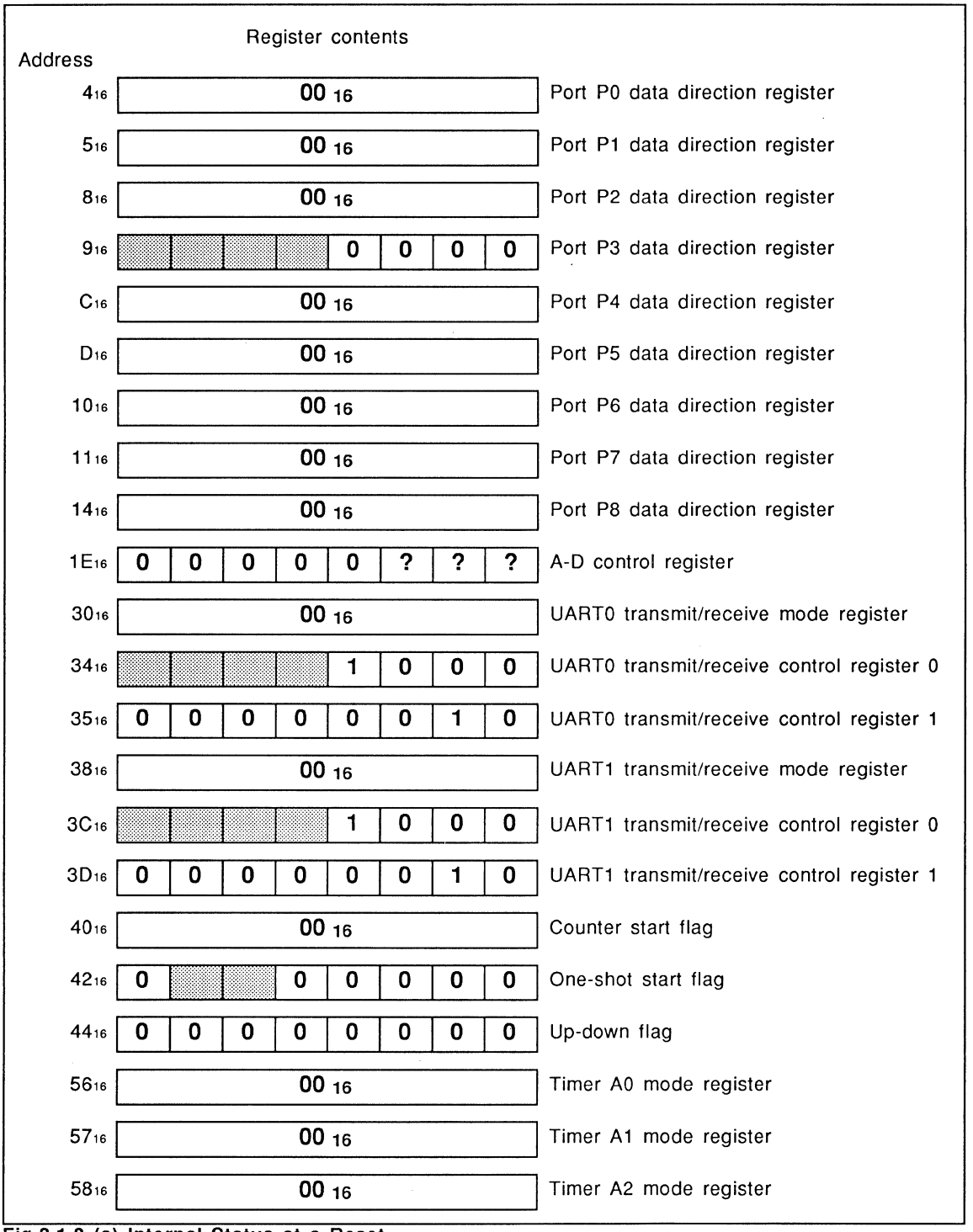


Fig.3.1.2 (a) Internal Status at a Reset

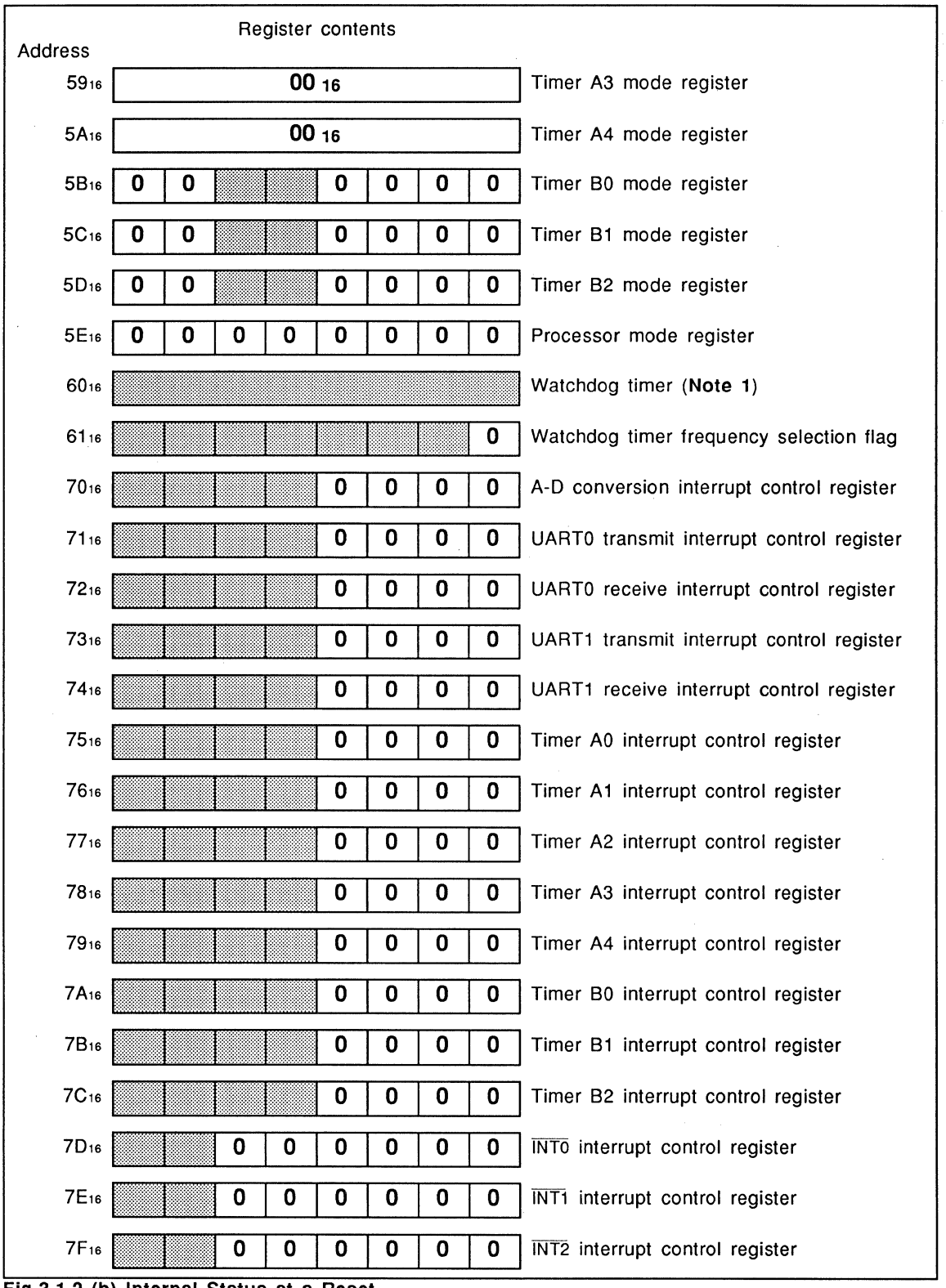
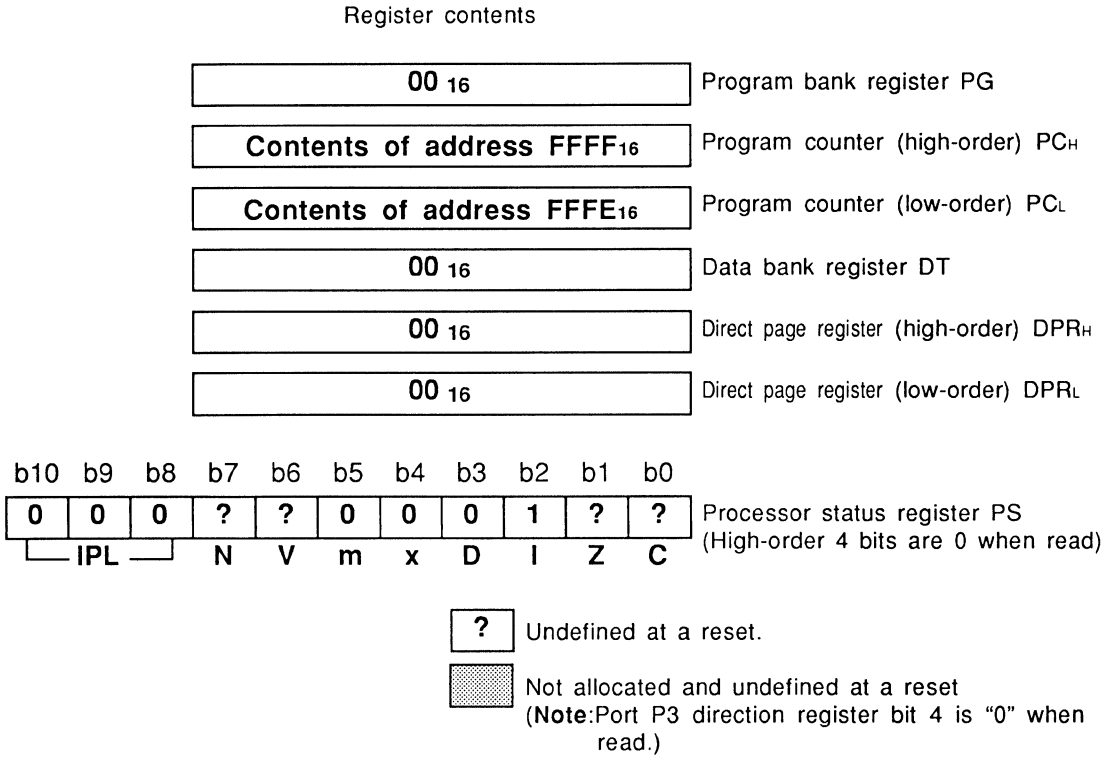


Fig.3.1.2 (b) Internal Status at a Reset



Note 1. Watchdog timer is set to FF_{16} at the reset.
Note 2. The contents of registers and RAM other than those described in Figure 3.1.2 are undefined at a reset is deactivated.

Fig.3.1.2 (c) Internal Status at a Reset

3.2 Reset circuit

The reset circuit must be designed so that the reset input voltage drops below 0.9V when the source voltage reaches 4.5V as shown in Figure 3.2.1.
Figure 3.2.2 shows an example of power-on reset circuit using a system reset IC M51957AL.

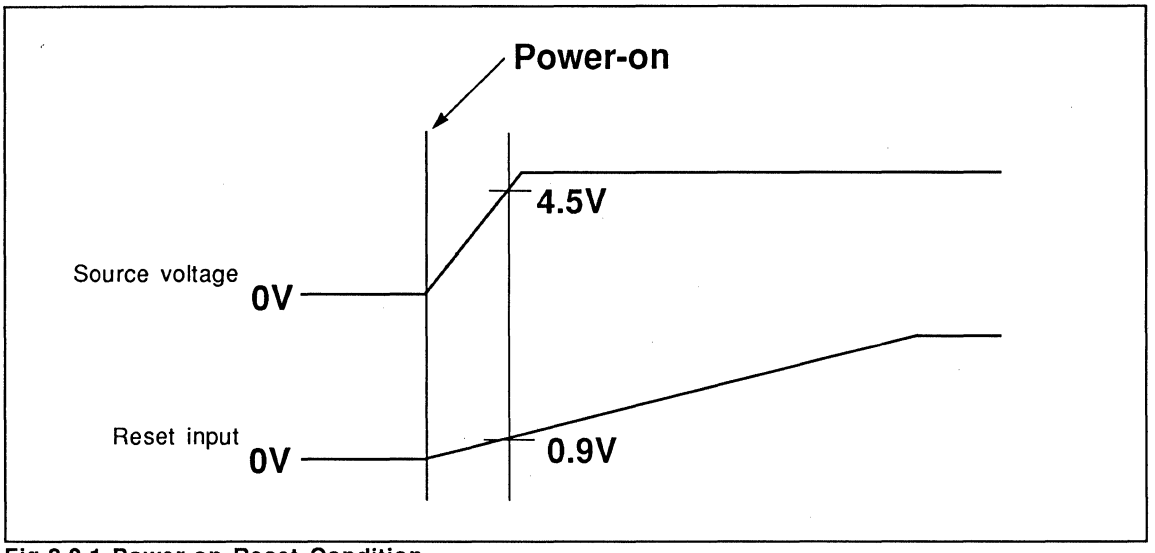


Fig.3.2.1 Power-on Reset Condition

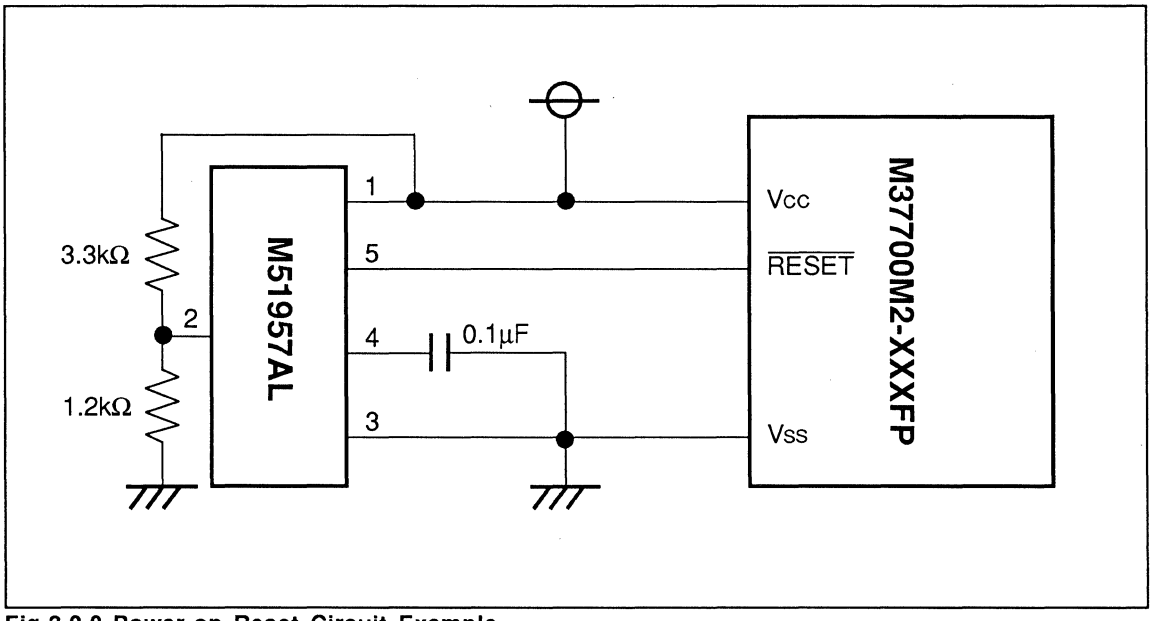
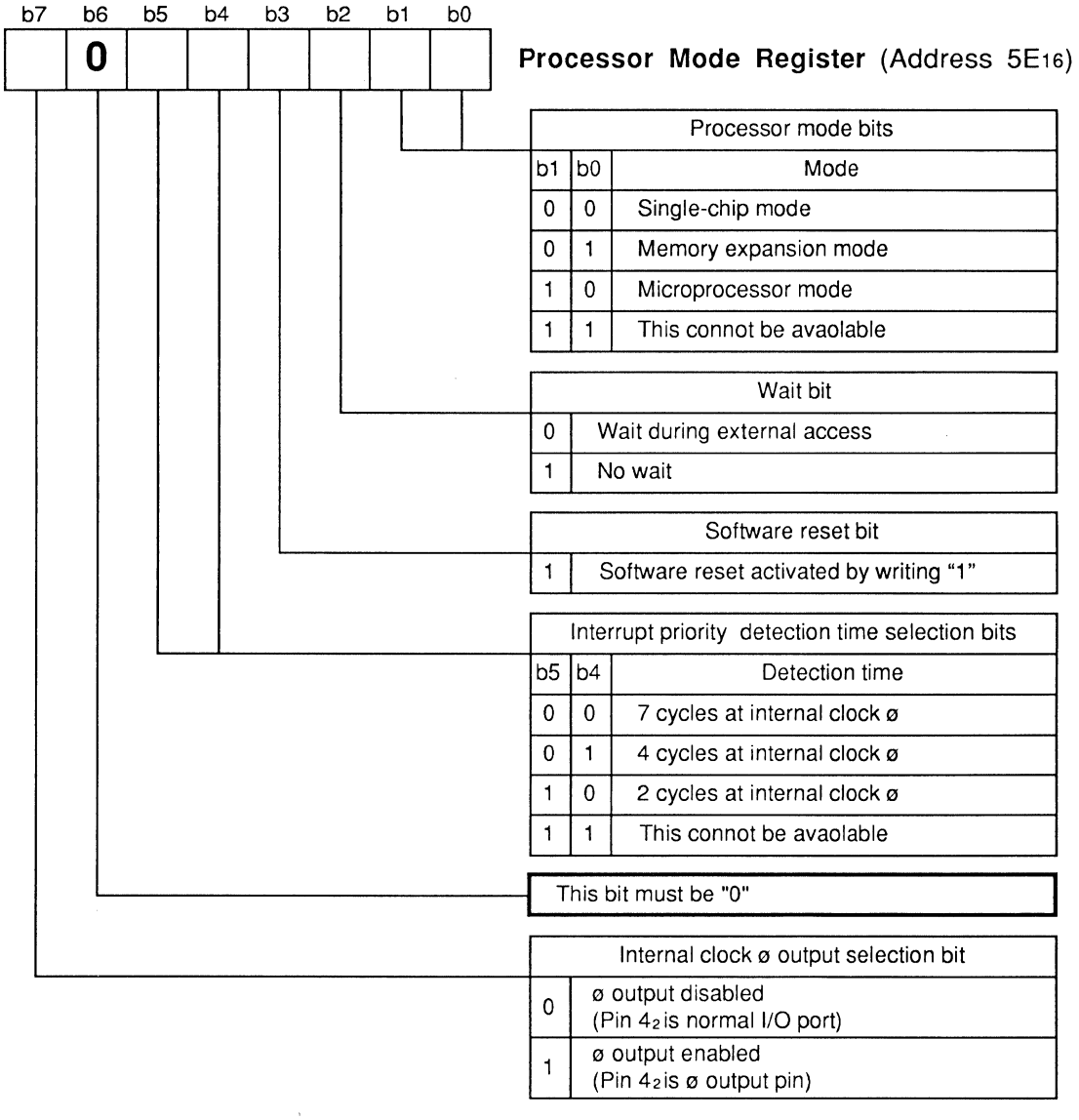


Fig.3.2.2 Power-on Reset Circuit Example

3.3 Software reset

The M37700 can be reset internally with a program. This is done by setting the processor mode register bit 3 to "1". Figure 3.3.1 shows the bit configuration of the processor mode register.



Note : Use the **LDM** or **STA** instruction to write to this register.
(Do not use read modify write type instructions such as **CLB** or **SEB**.)

Fig.3.3.1 Processor Mode Register Bit Structure

Software reset is the same as hardware reset (when the reset pin is pulled to "L" and then restored to "H") except that the contents of the internal RAM are preserved. Therefore, the contents of each register after a software reset is initialized to values shown in Figure 3.1.2.

CHAPTER 4
OSCILLATING CIRCUIT

4.1 Oscillating Circuit

The M37700 is equipped with an oscillating circuit to generate the necessary clock. The frequency input to the clock input pin X_{IN} is divided in half to obtain the internal clock ø. This ø is further divided in half to obtain the bus cycle. Either a ceramic resonator or a crystal oscillator can be connected externally to the internal oscillating circuit.

4.1.1 Circuit using a ceramic oscillator or a crystal oscillator

Figure 4.1.1 shows a circuit example using a ceramic resonator and Figure 4.1.2 shows a circuit example using a crystal resonator. An oscillating circuit is formed by connecting the resonator between X_{IN} and X_{OUT} as shown in the figures. The circuit constants such as Rf, Rd, C_{IN}, and C_{OUT} must be set to the resonator manufacturer's recommended values.

Table 4.1.1 and Table 4.1.2 show the recommended circuit constants for each type of oscillator.

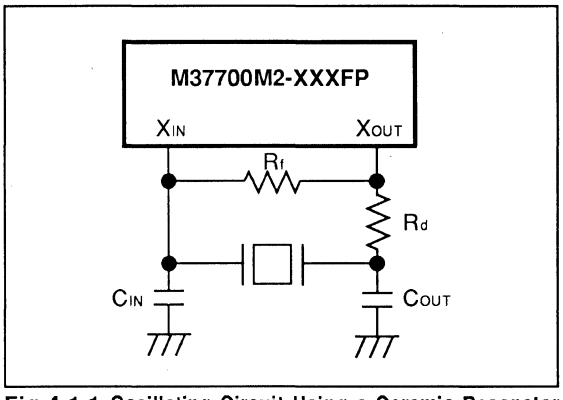


Fig.4.1.1 Oscillating Circuit Using a Ceramic Resonator

4.1.2 External clock input circuit

An external clock signal can be supplied to the internal oscillating circuit. Figure 4.1.3 shows the circuit example for this case. Note that the external clock must be input from pin X_{IN}, and pin X_{OUT} must be left open.

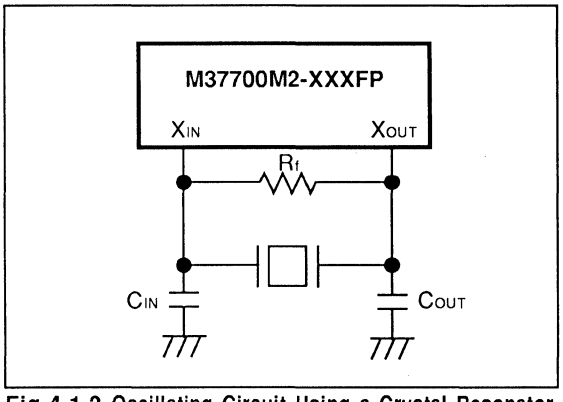


Fig.4.1.2 Oscillating Circuit Using a Crystal Resonator

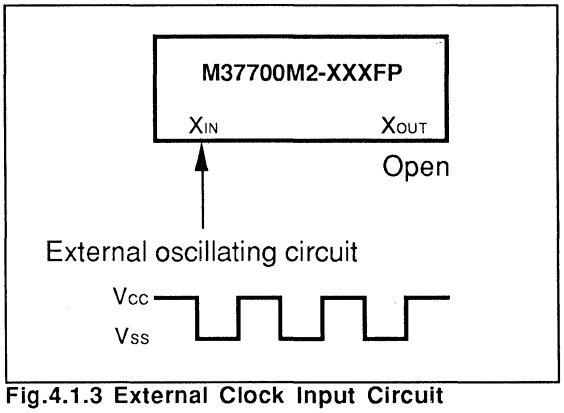


Fig.4.1.3 External Clock Input Circuit

Table 4.1.1 Recommended Circuit Constants when Using a Ceramic Resonator (See Figure 4.1.1)

Manufacturer	Type name	Oscillating frequency (MHz)	Rf (MΩ)	C_{IN} (pF)	C_{OUT} (pF)	Pd (Ω)
MURATA MFG. CO., LTD.	CSA8.00MT	8	1	30	30	220
MURATA MFG. CO., LTD.	CST8.00MT	8	1	30*	30*	220
KYOCERA CORP.	KBR-8.0M	8	1	33	33	470
MURATA MFG. CO., LTD.	CSA16.00MX040	16	1	47	47	—

* Built in the oscillator

Table 4.1.2 Recommended Circuit Constants when Using a Crystal Resonator (See Figure 4.1.2)

Manufacturer	Type name	Oscillating frequency (MHz)	Rf (MΩ)	C_{IN} (pF)	C_{OUT} (pF)
DAIWA SHINKU CORP.	HC-49/u8.000MHz	8	1	39	39
DAIWA SHINKU CORP.	HC-49/u16.000MHz	16	1	39	39

4.2 Clock Generation Circuit

The oscillating circuit consists of an oscillating gate which functions as an amplifier to obtain the necessary gain and an oscillation control flip-flop to control it. Therefore, the oscillation can be started or stopped as necessary. The M37700 is equipped with a clock generation circuit shown in Figure 4.2.1. When an **STP** instruction is executed, the internal clock ø stops at "L" state. At the same time, FFF_{16} is set in the watchdog timer and the watchdog timer input is connected to f_{32}. When the most significant bit of the watchdog timer becomes "0" or when the system is reset, this connection is freed and the watchdog timer is connected to the input determined by the watchdog timer frequency selection flag. Oscillation resumes when an interrupt is received, but the internal clock ø remains at "L" till the most significant bit of the watchdog timer becomes "0". This is to avoid the unstable oscillation period at the rising edge when using a ceramic resonator. In order to enable the **STP** instruction, the **STP** option must be specified when ordering the mask.

When a WIT instruction is executed, the internal clock ø stops at "L" state, but the oscillator does not stop. ø is resumed when an interrupt is received. An instruction is executed immediately because the oscillator is not stopped.

If an instruction using the bus is executed just before execution the **STP** or **WIT** instruction, the program may run-away after recovering from a stop or wait state. Therefore, a **NOP** instruction should be inserted before an **STP** or **WIT** instruction to delay execution. If there is an instruction to change the port output or RAM content just before an **STP** or **WIT** instruction, the port output or the RAM content might not be changed. In this case, insert the number of **NOP** instruction, as shows the following, before the **STP** or **WIT** instruction to adjust the execution timing.

- The data is written into internal RAM or SFR ..... One **NOP** instruction
- The data is written into external memory or I/O without one-shot wait ..... One **NOP** instruction
- The data is written into external memory or I/O with one-shot wait ..... Three **NOP** instructions

The stop or wait state is canceled with an interrupt or with a reset.

- These states are canceled with an interrupt, the following interrupts can be used:
  - Stop state
    - –External interrupt ($\overline{INT0}$, $\overline{INT1}$, and $\overline{INT2}$)
      Clock synchronous serial I/O interrupts (transmission, receive) using external clock
    - –UART interrupts (UART0 receive, UART0 transmission, UART1 receive, UART1 transmission) using external clock
    - –Timer interrupts (however, timer which is used to cancel has to sets the event counter mode before executing **STP** instruction.)
  - Wait state
    - –All hardware interrupts

  However, interrupts must be enabled before executing an **STP** or **WIT** instruction.

- These states are canceled with reset, hardware reset can be only used.
  After cancel, the contents of internal RAM is kept and the contents SFR and CPU registers are shown in Figures 3.1.2(a) to (c).

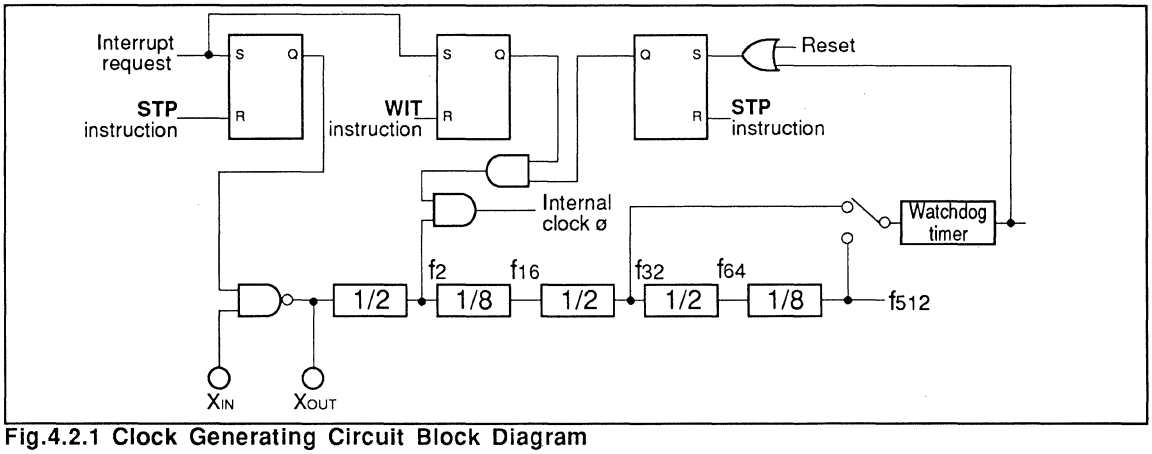


Fig.4.2.1 Clock Generating Circuit Block Diagram

CHAPTER 5
ELECTRICAL CHARACTERISTICS

CHAPTER 5.ELECTRICAL CHARACTERISTICS

5.1 Absolute Maximum Ratings

Absolute Maximum Ratings

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		–0.3~7	V
AVcc	Analog supply voltage		–0.3~7	V
V_I	Input voltage $\overline{RESET}$, CNVss, BYTE		–0.3~12	V
V_I	Input voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$ $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$ $P8_0$~$P8_7$, V_{REF}, X_{IN}		–0.3~Vcc+0.3	V
V_O	Output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$ $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$ $P8_0$~$P8_7$, X_{OUT}, $\overline{E}$		–0.3~Vcc+0.3	V
Pd	Power dissipation	Ta=25°C	300	mW
Topr	Operating temperature		–10~70	°C
Tstg	Storage temperature		–40~125	°C

5.2 Recommended Operating Conditions

Recommended Operating Conditions (V_{CC}=5V±10%, Ta=−10~70°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Nom.	Max.	
V_{CC}	Supply voltage		4.5	5.0	5.5	V
AVcc	Analog supply voltage			Vcc		V
V_{SS}	Supply voltage			0		V
AVss	Analog supply voltage			0		V
V_{IH}	High-level input voltage	$P0_0$~$P0_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$ $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS} BYTE	0.8Vcc		Vcc	V
V_{IH}	High-level input voltage	$P1_0$~$P1_7$, $P2_0$~$P2_7$ (in single-chip mode)	0.8Vcc		Vcc	V
V_{IH}	High-level input voltage	$P1_0$~$P1_7$, $P2_0$~$P2_7$ (in memory expansion mode and microprocessor mode)	0.5Vcc		Vcc	V
V_{IL}	Low-level input voltage	$P0_0$~$P0_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$ $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS} BYTE	0		0.2Vcc	V
V_{IL}	Low-level input voltage	$P1_0$~$P1_7$, $P2_0$~$P2_7$ (in single-chip mode)	0		0.2Vcc	V
V_{IL}	Low-level input voltage	$P1_0$~$P1_7$, $P2_0$~$P2_7$ (in memory expansion mode and microprocessor mode)	0		0.16Vcc	V
$I_{OH(peak)}$	High-level peak output current	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$			−10	mA
$I_{OH(avg)}$	High-level average output current	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$			−5	mA
$I_{OL(peak)}$	Low-level peak output current	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$			10	mA
$I_{OL(avg)}$	Low-level average output current	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$			5	mA
$f(X_{IN})$	External clock frequency input	M37700M2-XXXFP, M37700SFP			8	MHz
		M37700M2AXXXFP, M37700SAFP			16	MHz

Note 1. Average output current is the average value of a 100ms interval.

2. The sum of I_{OL}(peak) for ports P0, P1, P2, P3, and P8 must be 80mA or less.
The sum of I_{OH}(peak) for ports P0, P1, P2, P3, and P8 must be 80mA or less.
The sum of I_{OL}(peak) for ports P4, P5, P6, and P7 must be 80mA or less.
The sum of I_{OH}(peak) for ports P4, P5, P6, and P7 must be 80mA or less.

5.3 M37700M2-XXXFP DC/AC Characteristics

5.3.1 Electrical characteristics

Electrical characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, $f(X_{IN})$=8MHz, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$ $P8_0$~$P8_7$	I_{OH}=–10mA	3			V
V_{OH}	High-level output voltage	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$, $P3_1$, $P3_3$	I_{OH}=–400µA	4.7			V
V_{OH}	High-level output voltage	$P3_2$	I_{OH}=–10mA	3.1			V
			I_{OH}=–400µA	4.8			V
V_{OH}	High-level output voltage	$\overline{E}$	I_{OH}=–10mA	3.4			V
			I_{OH}=–400µA	4.8			V
V_{OL}	Low-level output voltage	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$ $P8_0$~$P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$, $P3_1$, $P3_3$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage	$P3_2$	I_{OL}=10mA			1.9	V
			I_{OL}=2mA			0.43	V
V_{OL}	Low-level output voltage	$\overline{E}$	I_{OL}=10mA			1.6	V
			I_{OL}=2mA			0.4	V
V_{T+}–V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$ $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1		0.4		1	V
V_{T+}–V_{T-}	Hysteresis	$\overline{RESET}$		0.2		0.5	V
V_{T+}–V_{T-}	Hysteresis	X_{IN}		0.1		0.3	V
I_{IH}	High-level input current	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$ X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	µA
I_{IL}	Low-level input current	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$ X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			–5	µA
V_{RAM}	RAM hold voltage		When clock is stopped.	2			V
I_{CC}	Power supply current		In single-chip mode, an output pin is kept open and other pins are connected to V_{SS} during reset. / $f(X_{IN})$=8MHz square waveform		6	12	mA
			Ta=25°C, when clock is stopped.			1	µA
			Ta=70°C, when clock is stopped.			10	µA

5.3.2 A-D converter characteristics

A-D converter characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, $f(X_{IN})$=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	V_{REF}=V_{CC}			8	Bits
—	Absolute accuracy	V_{REF}=V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF}=V_{CC}	2		10	kΩ
t_{CONV}	Conversion time		28.5			µs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

5.3.3 Timing requirements (V_{CC}=5V±10%, V_{SS}=0V, Ta=25°C, $f(X_{IN})$=8MHz, unless otherwise noted)

External clock input

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_c	External clock input cycle time	125			ns
$t_{w(H)}$	External clock input high-level pulse width	50			ns
$t_{w(L)}$	External clock input low-level pulse width	50			ns
t_r	External clock rise time			20	ns
t_f	External clock fall time			20	ns

Single-chip mode

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{su(P0D-E)}$	Port P0 input setup time	200			ns
$t_{su(P1D-E)}$	Port P1 input setup time	200			ns
$t_{su(P2D-E)}$	Port P2 input setup time	200			ns
$t_{su(P3D-E)}$	Port P3 input setup time	200			ns
$t_{su(P4D-E)}$	Port P4 input setup time	200			ns
$t_{su(P5D-E)}$	Port P5 input setup time	200			ns
$t_{su(P6D-E)}$	Port P6 input setup time	200			ns
$t_{su(P7D-E)}$	Port P7 input setup time	200			ns
$t_{su(P8D-E)}$	Port P8 input setup time	200			ns
$t_{h(E-P0D)}$	Port P0 input hold time	0			ns
$t_{h(E-P1D)}$	Port P1 input hold time	0			ns
$t_{h(E-P2D)}$	Port P2 input hold time	0			ns
$t_{h(E-P3D)}$	Port P3 input hold time	0			ns
$t_{h(E-P4D)}$	Port P4 input hold time	0			ns
$t_{h(E-P5D)}$	Port P5 input hold time	0			ns
$t_{h(E-P6D)}$	Port P6 input hold time	0			ns
$t_{h(E-P7D)}$	Port P7 input hold time	0			ns
$t_{h(E-P8D)}$	Port P8 input hold time	0			ns

Memory expansion mode and microprocessor mode

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{su(P0D-E)}$	Port P1 input setup time	60			ns
$t_{su(P2D-E)}$	Port P2 input setup time	60			ns
$t_{su(E-RDY)}$	$\overline{RDY}$ input setup time (when wait bit is "1")	0		60	ns
$t_{su(E-RDY)}$	$\overline{RDY}$ input setup time (when wait bit is "0", and external memory is accessed)	0		300	ns
$t_{h(E-P1D)}$	Port P1 input hold time	0			ns
$t_{h(E-P2D)}$	Port P2 input hold time	0			ns

Timer A input (count input in event counter mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	250			ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125			ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125			ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	2000			ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	1000			ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	1000			ns

Timer A input (external trigger input in one-shot pulse mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	500			ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250			ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250			ns

Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250			ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250			ns

Timer A input (up-down input in event counter mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(UP)}$	TAi_{OUT} input cycle time	2000			ns
$t_{W(UPH)}$	TAi_{OUT} input high-level pulse width	1000			ns
$t_{W(UPL)}$	TAi_{OUT} input low-level pulse width	1000			ns

Timer B input (count input in event counter mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	500			ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	250			ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	250			ns

Timer B input (pulse period measurement mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	2000			ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	1000			ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	1000			ns

Timer B input (pulse width measurement mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TB)}$	TBiIN input cycle time	2000			ns
$t_{W(TBH)}$	TBiIN input high-level pulse width	1000			ns
$t_{W(TBL)}$	TBiIN input low-level pulse width	1000			ns

A-D trigger input

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(AD)}$	$\overline{ADTRG}$ input cycle time (minimum allowable trigger)	2000			ns
$t_{W(ADL)}$	$\overline{ADTRG}$ input low-level pulse width	250			ns

UART clock input

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(CK)}$	CLKi input cycle time	500			ns
$t_{W(CKH)}$	CLKi input high-level pulse width	250			ns
$t_{W(CKL)}$	CLKi input low-level pulse width	250			ns

External interrupt $\overline{INTi}$ input

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(INH)}$	$\overline{INTi}$ input high-level pulse width	250			ns
$t_{W(INL)}$	$\overline{INTi}$ input low-level pulse width	250			ns

5.3.4 Switching characteristics (V_{CC}=5V±10%, V_{SS}=0V, Ta=25°C, f(X_{IN})=8MHz, unless otherwise noted)

Single-chip mode

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time	Figure 5.3.1			200	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time				200	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time				200	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time				200	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time				200	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time				200	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time				200	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time				200	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time				200	ns

Memory expansion mode and microprocessor mode (when wait bit is "1")

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time *	Figure 5.3.1	100			ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")				120	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")				40	ns
$t_{d(P1A-E)}$	Port P1 address output delay time *		100			ns
$t_{d(E-P2Q)}$	Port P2 data output delay time				120	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time				40	ns
$t_{d(P2A-E)}$	Port P2 address output delay time *		100			ns
$t_{d(E-HLDA)}$	$\overline{HLDA}$ output delay time				100	ns
$t_{d(ALE-E)}$	ALE output delay time		−10			ns
$t_{W(ALE)}$	ALE pulse width *		100			ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time *		100			ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time *		100			ns
$t_{h(E-P0A)}$	Port P0 address hold time		20			ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="L")		20			ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		20			ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		20			ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		20			ns
$t_{h(E-P2A)}$	Port P2 address hold time		20			ns
$t_{h(E-P2Q)}$	Port P2 data hold time		20			ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		20			ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		20			ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		20			ns

Memory expansion mode and microprocessor mode (when external memory area is accessed, and wait bit is "0")

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time *	Figure 5.3.1	350			ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")				120	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")				40	ns
$t_{d(P1A-E)}$	Port P1 address output delay time *		350			ns
$t_{d(E-P2Q)}$	Port P2 data output delay time				120	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time				40	ns
$t_{d(P2A-E)}$	Port P2 address output delay time *		350			ns
$t_{d(E-HLDA)}$	$\overline{HLDA}$ output delay time				100	ns
$t_{d(ALE-E)}$	ALE output delay time		−10			ns
$t_{W(ALE)}$	ALE pulse width *		350			ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time *		350			ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time *		350			ns
$t_{h(E-P0A)}$	Port P0 address hold time		20			ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="L")		20			ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		20			ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		20			ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		20			ns
$t_{h(E-P2A)}$	Port P2 address hold time		20			ns
$t_{h(E-P2Q)}$	Port P2 data hold time		20			ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		20			ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		20			ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		20			ns

* The value of port **Pi** address output delay time (**i**=0, 1, or 2), $\overline{BHE}$ output delay time, R/$\overline{W}$ output delay time, and ALE pulse width are dependent on the clock oscillating frequency ($f(X_{IN})$), and these value are defined by the following expressions.

Symbol	Parameter	Wait bit ="1"	Wait bit ="0"
$t_{d(PiA-E)}$	Port Pi address output delay time	$100+(2 \times 10^9/f(X_{IN}))-250ns$	$350+(4 \times 10^9/f(X_{IN}))-500ns$
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		
$t_{W(ALE)}$	ALE pulse width	$(1 \times 10^9/f(X_{IN}))-25ns$	$(3 \times 10^9/f(X_{IN}))-25ns$

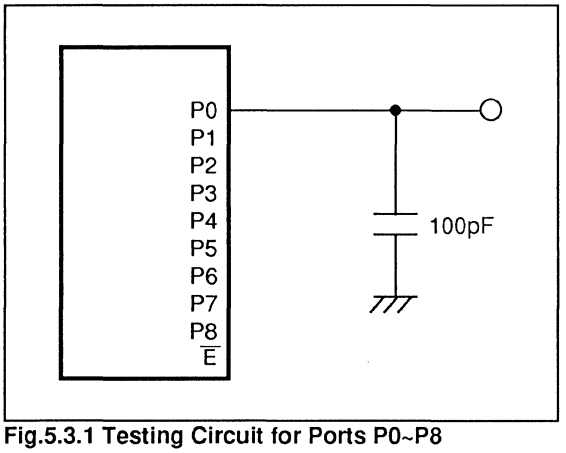


Fig.5.3.1 Testing Circuit for Ports P0~P8

5.4 M37700M2AXXXFP DC/AC Characteristics

5.4.1 Electrical characteristics

Electrical characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$ $P8_0$~$P8_7$	I_{OH}=–10mA	3			V
V_{OH}	High-level output voltage	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$, $P3_1$, $P3_3$	I_{OH}=–400μA	4.7			V
V_{OH}	High-level output voltage	$P3_2$	I_{OH}=–10mA	3.1			V
			I_{OH}=–400μA	4.8			V
V_{OH}	High-level output voltage	$\overline{E}$	I_{OH}=–10mA	3.4			V
			I_{OH}=–400μA	4.8			V
V_{OL}	Low-level output voltage	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$ $P8_0$~$P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$, $P3_1$, $P3_3$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage	$P3_2$	I_{OL}=10mA			1.9	V
			I_{OL}=2mA			0.43	V
V_{OL}	Low-level output voltage	$\overline{E}$	I_{OL}=10mA			1.6	V
			I_{OL}=2mA			0.4	V
V_{T+}–V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$ $\overline{INT0}$~$\overline{INT2}$, $\overline{ADTRG}$, $\overline{CTS0}$, $\overline{CTS1}$, $CLK0$, $CLK1$		0.4		1	V
V_{T+}–V_{T-}	Hysteresis	$\overline{RESET}$		0.2		0.5	V
V_{T+}–V_{T-}	Hysteresis	X_{IN}		0.1		0.3	V
I_{IH}	High-level input current	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$ X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current	$P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$ X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=0V			–5	μA
V_{RAM}	RAM hold voltage		When clock is stopped.	2			V
I_{CC}	Power supply current		In single-chip mode, an output pin is kept open and other pins are connected to Vss during reset. — f(X_{IN})=16MHz square waveform		12	24	mA
			Ta=25°C, when clock is stopped.			1	μA
			Ta=70°C, when clock is stopped.			10	μA

5.4.2 A-D converter characteristics

A-D converter characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	V_{REF}=V_{CC}			8	Bits
—	Absolute accuracy	V_{REF}=V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF}=V_{CC}	2		10	kΩ
t_{CONV}	Conversion time		14.25			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

5.4.3 Timing requirements (V_{CC}=5V±10%, V_{SS}=0V, Ta=25°C, $f(X_{IN})$=16MHz, unless otherwise noted)

External clock input

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_C	External clock input cycle time	62			ns
$t_{W(H)}$	External clock input high-level pulse width	25			ns
$t_{W(L)}$	External clock input low-level pulse width	25			ns
t_r	External clock rise time			10	ns
t_f	External clock fall time			10	ns

Single-chip mode

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{SU(P0D-E)}$	Port P0 input setup time	100			ns
$t_{SU(P1D-E)}$	Port P1 input setup time	100			ns
$t_{SU(P2D-E)}$	Port P2 input setup time	100			ns
$t_{SU(P3D-E)}$	Port P3 input setup time	100			ns
$t_{SU(P4D-E)}$	Port P4 input setup time	100			ns
$t_{SU(P5D-E)}$	Port P5 input setup time	100			ns
$t_{SU(P6D-E)}$	Port P6 input setup time	100			ns
$t_{SU(P7D-E)}$	Port P7 input setup time	100			ns
$t_{SU(P8D-E)}$	Port P8 input setup time	100			ns
$t_{h(E-P0D)}$	Port P0 input hold time	0			ns
$t_{h(E-P1D)}$	Port P1 input hold time	0			ns
$t_{h(E-P2D)}$	Port P2 input hold time	0			ns
$t_{h(E-P3D)}$	Port P3 input hold time	0			ns
$t_{h(E-P4D)}$	Port P4 input hold time	0			ns
$t_{h(E-P5D)}$	Port P5 input hold time	0			ns
$t_{h(E-P6D)}$	Port P6 input hold time	0			ns
$t_{h(E-P7D)}$	Port P7 input hold time	0			ns
$t_{h(E-P8D)}$	Port P8 input hold time	0			ns

Memory expansion mode and microprocessor mode

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{SU(P0D-E)}$	Port P1 input setup time	45			ns
$t_{SU(P2D-E)}$	Port P2 input setup time	45			ns
$t_{SU(E-RDY)}$	$\overline{RDY}$ input setup time (when wait bit is "1")	0		10	ns
$t_{SU(E-RDY)}$	$\overline{RDY}$ input setup time (when wait bit is "0", and external memory is accessed)	0		120	ns
$t_{h(E-P1D)}$	Port P1 input hold time	0			ns
$t_{h(E-P2D)}$	Port P2 input hold time	0			ns

Timer A input (count input in event counter mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	125			ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	62			ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	62			ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	1000			ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	500			ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	500			ns

Timer A input (external trigger input in one-shot pulse mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	250			ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125			ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125			ns

Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125			ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125			ns

Timer A input (up-down input in event counter mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(UP)}$	TAi_{OUT} input cycle time	1000			ns
$t_{W(UPH)}$	TAi_{OUT} input high-level pulse width	500			ns
$t_{W(UPL)}$	TAi_{OUT} input low-level pulse width	500			ns

Timer B input (count input in event counter mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	250			ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	125			ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	125			ns

Timer B input (pulse period measurement mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	1000			ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	500			ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	500			ns

Timer B input (pulse width measurement mode)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	1000			ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	500			ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	500			ns

A-D trigger input

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(AD)}$	$\overline{ADTRG}$ input cycle time (minimum allowable trigger)	1000			ns
$t_{W(ADL)}$	$\overline{ADTRG}$ input low-level pulse width	125			ns

UART clock input

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{C(CK)}$	CLKi input cycle time	250			ns
$t_{W(CKH)}$	CLKi input high-level pulse width	125			ns
$t_{W(CKL)}$	CLKi input low-level pulse width	125			ns

External interrupt $\overline{INTi}$ input

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(INH)}$	$\overline{INTi}$ input high-level pulse width	250			ns
$t_{W(INL)}$	$\overline{INTi}$ input low-level pulse width	250			ns

5.4.4 Switching characteristics (V_{CC}=5V±10%, V_{SS}=0V, Ta=25°C, $f(X_{IN})$=16MHz, unless otherwise noted)

Single-chip mode

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time	Figure 5.4.1			100	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time				100	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time				100	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time				100	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time				100	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time				100	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time				100	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time				100	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time				100	ns

Memory expansion mode and microprocessor mode (when wait bit is "1")

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time *	Figure 5.4.1	30			ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")				80	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")				40	ns
$t_{d(P1A-E)}$	Port P1 address output delay time *		30			ns
$t_{d(E-P2Q)}$	Port P2 data output delay time				80	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time				40	ns
$t_{d(P2A-E)}$	Port P2 address output delay time *		30			ns
$t_{d(E-HLDA)}$	$\overline{HLDA}$ output delay time				50	ns
$t_{d(ALE-E)}$	ALE output delay time		–10			ns
$t_{W(ALE)}$	ALE pulse width *		40			ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time *		30			ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time *		30			ns
$t_{h(E-P0A)}$	Port P0 address hold time		20			ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="L")		20			ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		20			ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		20			ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		20			ns
$t_{h(E-P2A)}$	Port P2 address hold time		20			ns
$t_{h(E-P2Q)}$	Port P2 data hold time		20			ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		20			ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		20			ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		20			ns

Memory expansion mode and microprocessor mode (when external memory area is accessed, and wait bit is "0")

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time *	Figure 5.4.1	155			ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")				80	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")				40	ns
$t_{d(P1A-E)}$	Port P1 address output delay time *		155			ns
$t_{d(E-P2Q)}$	Port P2 data output delay time				80	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time				40	ns
$t_{d(P2A-E)}$	Port P2 address output delay time *		155			ns
$t_{d(E-HLDA)}$	$\overline{HLDA}$ output delay time				50	ns
$t_{d(ALE-E)}$	ALE output delay time		−10			ns
$t_{W(ALE)}$	ALE pulse width *		165			ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time *		155			ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time *		155			ns
$t_{h(E-P0A)}$	Port P0 address hold time		20			ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="L")		20			ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		20			ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		20			ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		20			ns
$t_{h(E-P2A)}$	Port P2 address hold time		20			ns
$t_{h(E-P2Q)}$	Port P2 data hold time		20			ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		20			ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		20			ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		20			ns

* The value of port **Pi** address output delay time (i=0, 1, or 2), $\overline{BHE}$ output delay time, R/$\overline{W}$ output delay time, and ALE pulse width are dependent on the clock oscillating frequency ($f(X_{IN})$), and these value are defined by the following expressions.

Symbol	Parameter	Wait bit ="1"	Wait bit ="0"
$t_{d(PiA-E)}$	Port Pi address output delay time	$30+(2 \times 10^9/f(X_{IN}))-125ns$	$155+(4 \times 10^9/f(X_{IN}))-250ns$
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		
$t_{W(ALE)}$	ALE pulse width	$(1 \times 10^9/f(X_{IN}))-22.5ns$	$(3 \times 10^9/f(X_{IN}))-22.5ns$

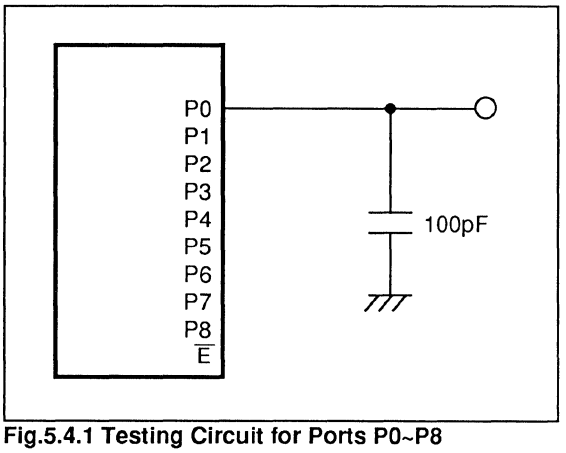
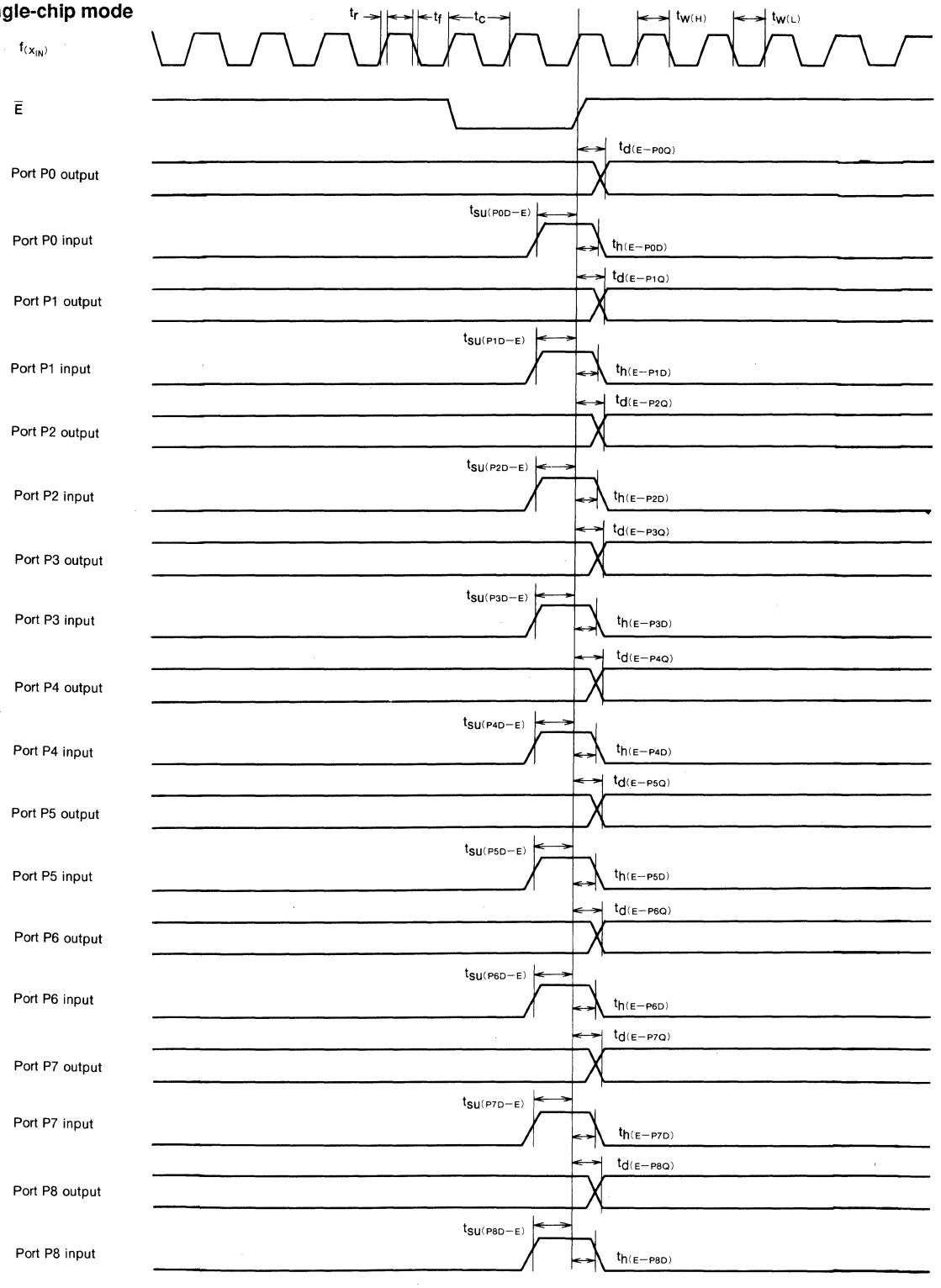
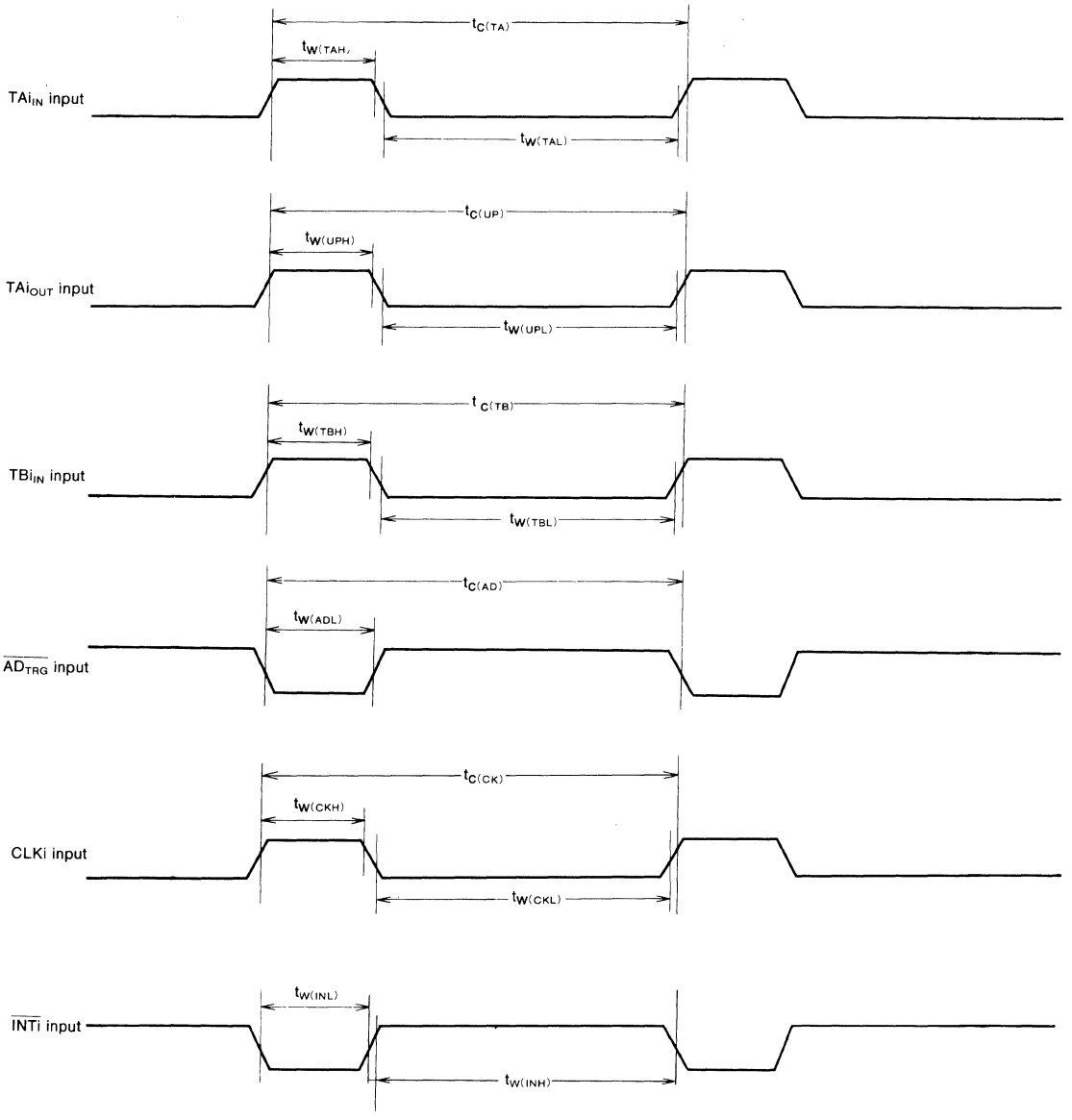


Fig.5.4.1 Testing Circuit for Ports P0~P8

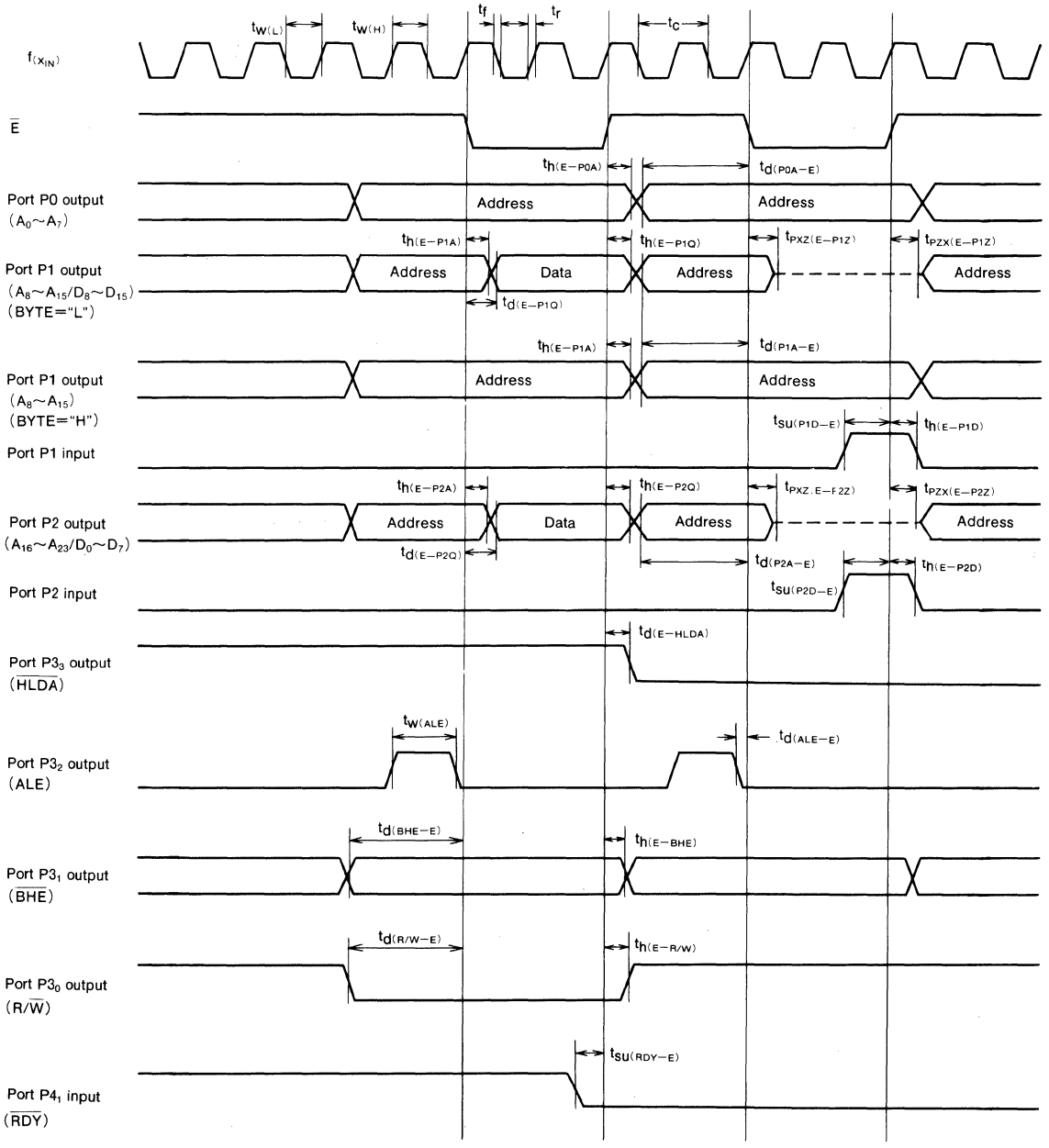
5.5 Timing Diagrams

Single-chip mode




$t_{C(TA)}$
$t_{W(TAH)}$
TAi_{IN} input
$t_{W(TAL)}$
$t_{C(UP)}$
$t_{W(UPH)}$
TAi_{OUT} input
$t_{W(UPL)}$
$t_{C(TB)}$
$t_{W(TBH)}$
TBi_{IN} input
$t_{W(TBL)}$
$t_{C(AD)}$
$t_{W(ADL)}$
$\overline{AD_{TRG}}$ input
$t_{C(CK)}$
$t_{W(CKH)}$
CLKi input
$t_{W(CKL)}$
$t_{W(INL)}$
$\overline{INTi}$ input
$t_{W(INH)}$

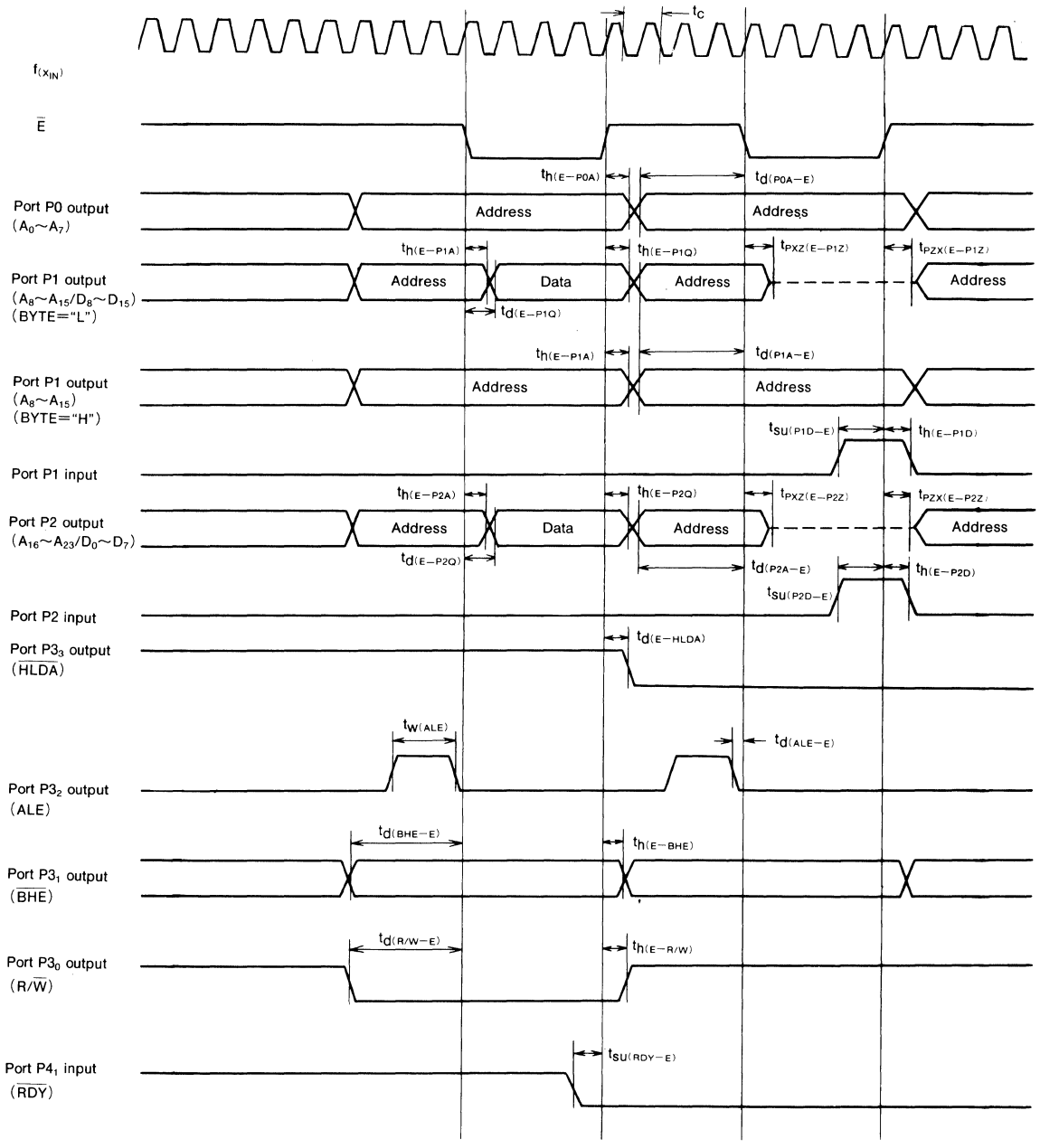
Memory expansion mode and microprocessor mode (when wait bit="1")



Test conditions

- V_{CC}=5V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- Ports P1 and P2 input : V_{IL}=0.8V, V_{IH}=2.5V
- Port $P4_1$ input : V_{IL}=1.0V, V_{IH}=4.0V

Memory expansion mode and microprocessor mode (when wait bit="0", and external memory area is accessed)



Test conditions

- V_{CC}=5V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- Ports P1 and P2 input : V_{IL}=0.8V, V_{IH}=2.5V
- Port $P4_1$ input : V_{IL}=1.0V, V_{IH}=4.0V

Memo

CHAPTER 6
STANDARD CHARACTERISTICS

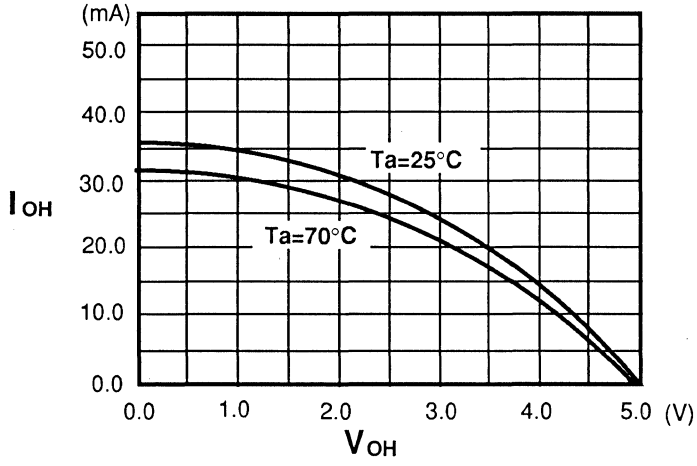
CHAPTER 6.STANDARD CHARACTERISTICS

The data described in this chapter are characteristic examples and are not guaranteed values. Refer to "Chapter 5. Electrical Characteristics" for rated values.

6.1 Standard Port Characteristics

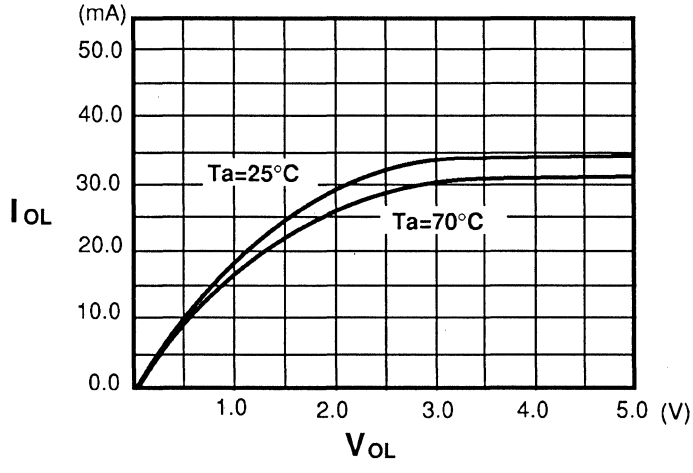
(1)Programmable I/O port (CMOS output) P channel I_{OH}–V_{OH} characteristics

- Supply voltage Vcc=5V



(2)Programmable I/O port (CMOS output) N channel I_{OL}–V_{OL} characteristics

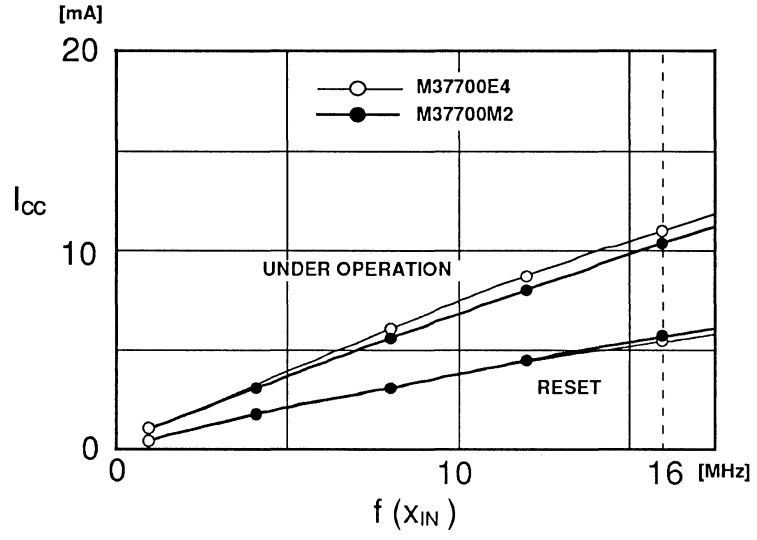
- Supply voltage Vcc=5V



6.2 Icc–f(XIN) Standard Characteristics

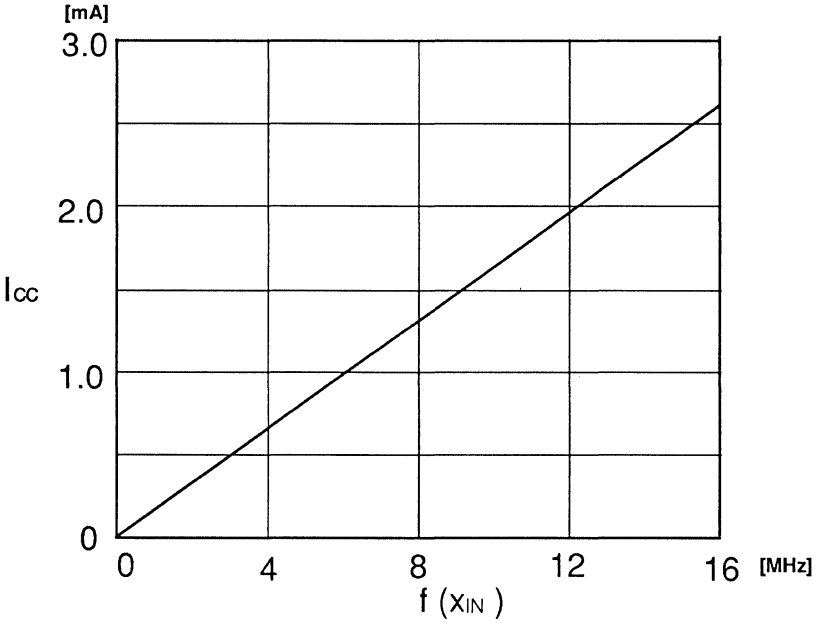
(1)M37700M2 and M37700E4 operating and reset Icc–f(XIN) characteristics

● Measurement condition (Vcc=5V, Ta=25°C, f(XIN): square wave, single-chip mode)



(2)M37700M2 wait Icc–f(XIN) characteristics

● Measurement condition (Vcc=5V, Ta=25°C, f(XIN): square wave, single-chip mode)

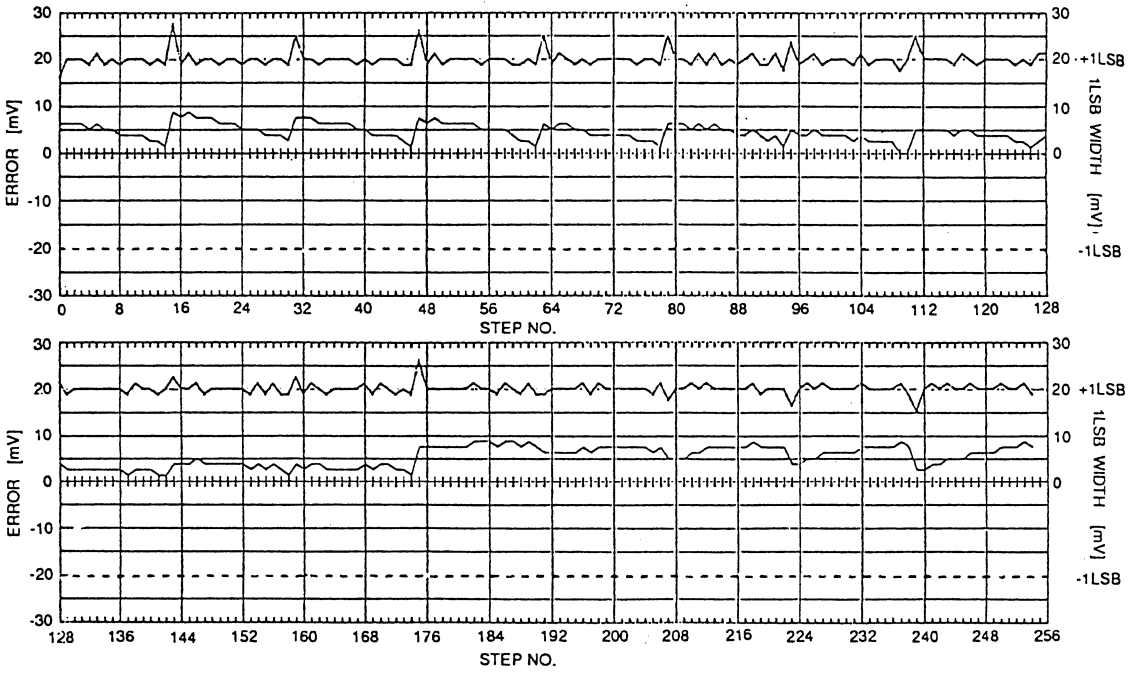


6.3 A-D Converter Standard Characteristics

The lines at the bottom of the graph indicate the absolute precision errors. These are expressed as the deviation from the ideal value when the output code changes. For example, the change in output code from 00_{16} to 01_{16} should occur at ANi=10mV, but the measured value is 6.5mV. Therefore, the measured point of change is 10+6.5=16.5mV.

The lines at the top of the graph indicate the input voltage width for which the output code is constant. For example, the measured input voltage width for which the output code is $0F_{16}$ is 27.5mV. Therefore, the differential non-linear error is 27.5-20=7.5mV (0.375LSB).

- V_{CC} = 5.12[V]
- V_{REF} = 5.12[V]
- X_{IN} = 16[MHz]
- Temp. = 25[deg.]



CHAPTER 7
USAGE PRECAUTIONS

CHAPTER 7.USAGE PRECAUTIONS

7.1 Software

Note the following when programming the M37700 and M37701 series of microprocessors.

7.1.1 Stack pointer (S)

The content of the stack pointer (S) is unpredictable immediately after a reset. Be sure to initialize it before using.

Example)

```
LDX  #27FH
TXS
```

7.1.2 Program bank register (PG) and data bank register (DT)

When using in single-chip mode, do not set values other than "00_{16}" in the program bank register and the data bank register.

7.1.3 Direct page register (DPR)

The execution cycle is reduced by one cycle, by setting the low-order 8-bits of the direct page register (DPR_L) to "00_{16}".

7.1.4 Processor status register (PS)

(1) Decimal mode flag (D flag)

When a decimal arithmetic operation is performed with the D flag set to "1",

- Only the C flag is valid and the Z, N, and V flags are invalid for **ADC** instruction.
- The C and Z flags are valid and the N and V flags are invalid for **SBC** instruction.

Note: Only the **ADC** and **SBC** instructions can be used for decimal arithmetic.

(2) Data length selection flag (m) and index register length selection flag (x)

Using 16-bit immediate data with the m flag set to "1" (data length: 8-bits) or 8-bit immediate data with the m flag set to "0" (data length: 16-bits) will cause programs to run wild. The same is true for the index register length selection flag x. Check the status of these flags when writing programs.

7.1.5 Register save and restore instructions

(1) Saving registers with PSH instruction and restoring registers with PUL instruction

When saving and restoring registers with the **PSH** and **PUL** instructions, the accumulators A and B are affected by the data length selection flag m and the index registers X and Y are affected by the index register length selection flag x (see Figures 7.1.1 and 7.1.2).

(2) Restoring processor mode register and accumulator B with the PUL instruction

When executing the **PUL** instruction with the data length selection flag m set to "0", if the processor status register is included in the register to be restored and its data length selection flag m is set to "1", the high-order 8-bits of the accumulator B may change. In this case, save and restore the processor status register with separate instructions.

Example)

```
PHP
PSH  A,B,X,Y
 :
PUL  A,B,X,Y
PLP
```

(3) PUL instruction

The N and Z flags change when the **PLA** instruction is executed, but the contents of the processor status register do not change if only accumulator A is restored with the **PUL** instruction.

Also, in addressing modes using the direct page register (DPR), the instruction execution cycle is reduced by one cycle if the content of the low-order 8 bits of DPR is "00_{16}".

The subsequent instruction execution cycle may be not reduced when the PUL instruction is used to restore the DPR containing "00_{16}". Therefore, do as follows:

Example)

```
PUL  #X0X1XXXXB    ;Restore registers including DPR
PRD                ;Save DHR
PLD                ;Restore DPR
```

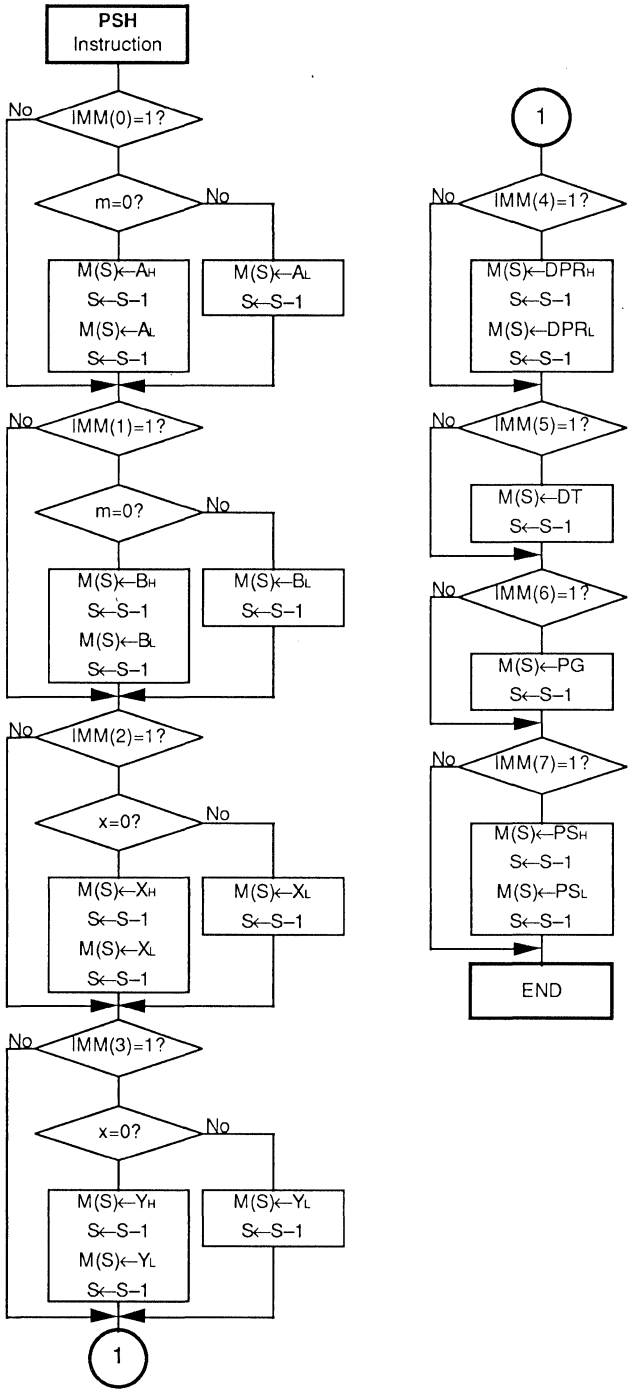


Fig.7.1.1 PSH Instruction Execution Flow

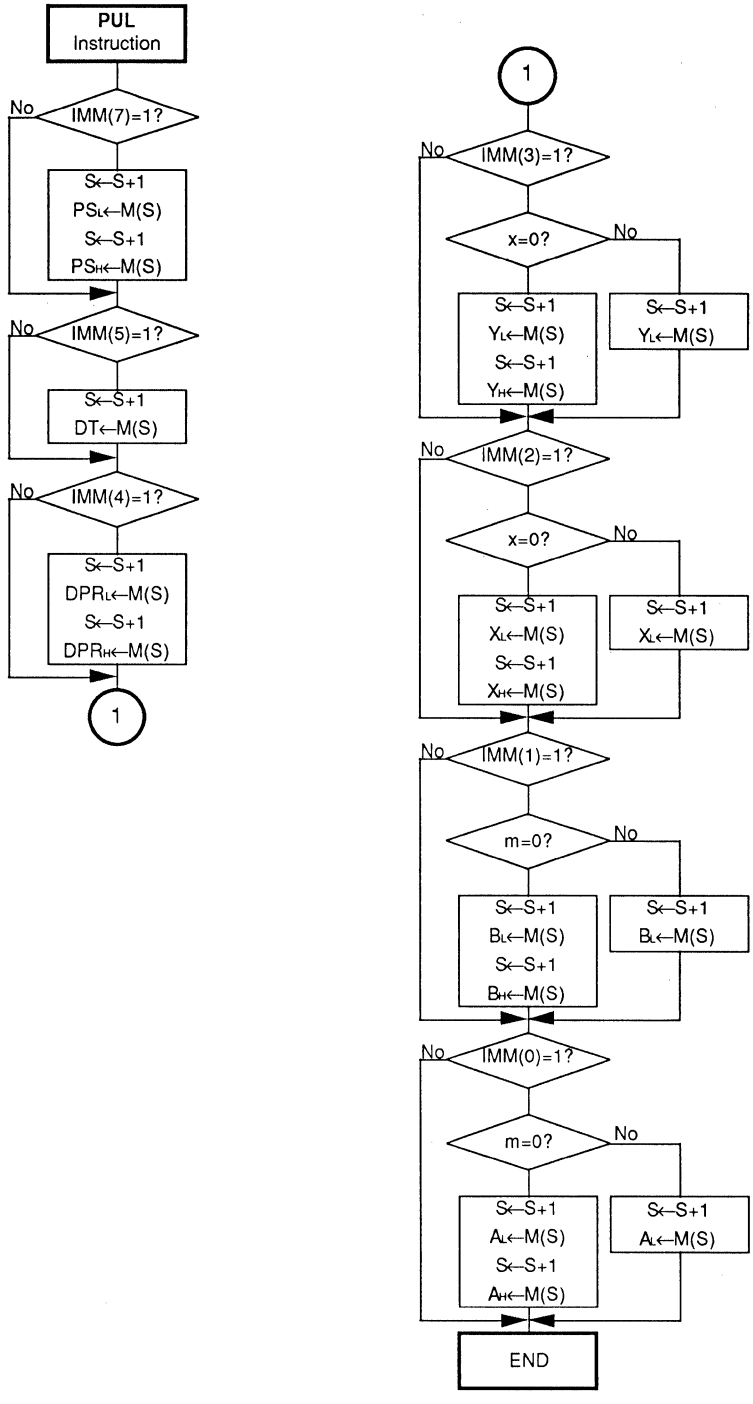


Fig.7.1.2 PUL Instruction Execution Flow

(4) PSH instruction

The program bank register PG can be saved to stack by setting the **PSH** instruction operand bit 6 to 1, but the **PUL** instruction cannot be used to restore PG.

7.1.6 Block transfer instructions (MVN, MVP)

With the block transfer instructions **MVP** and **MVN**, the content of accumulator A indicates the number of bytes to be transferred. For both the **MVN** and **MVP** instructions, the content of accumulator A is affected by flag m. The maximum number of bytes that can be transferred is 65535 bytes when m=0 and 255 bytes when m=1. Furthermore, no transfer is performed when the content of accumulator A is 0. Index registers X and Y indicate the transfer origin and destination addresses respectively and are affected by flag x. The transfer is within a 64K-byte address space when x=0 and within 256 bytes when x=1. When performing block transfer with x=1, the transfer is scrolled in cases such as shown Figure 7.1.3. Note that the data is overwritten if the number of transfer bytes is 256 bytes or more when m=0 and x=1.

Therefore, check the status of flags m and x when using the **MVP** or **MVN** instruction.

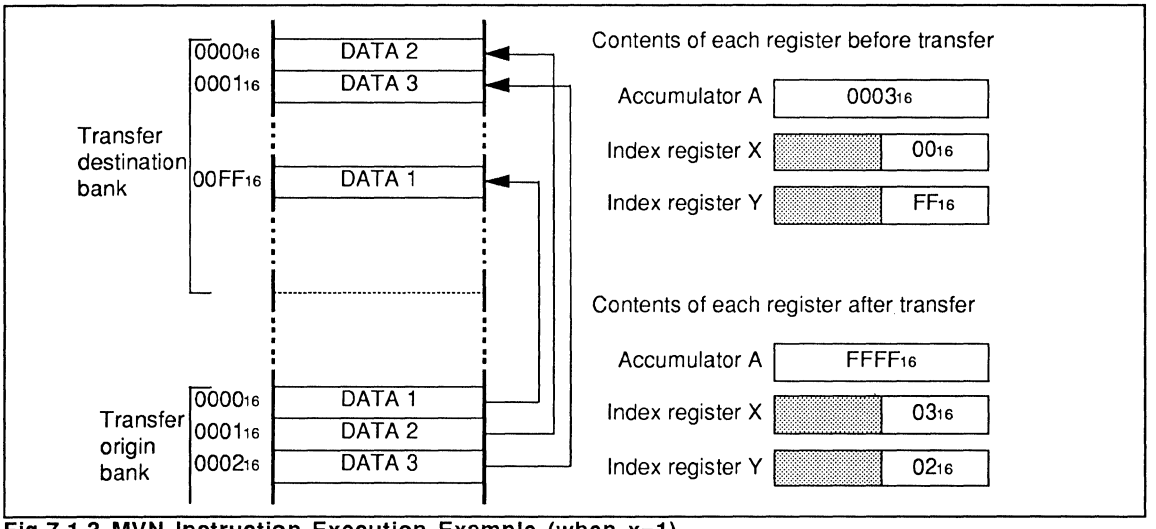


Fig.7.1.3 MVN Instruction Execution Example (when x=1)

7.1.7 BRK instruction

The CPU is unaffected regardless of the content of the second byte of the **BRK** instruction.

7.1.8 BRA instruction

Long relative branch with BRA instruction can only be used within bank 0.

7.1.9 Instruction execution time (instruction execution cycle)

(1) Time required for instruction execution

The MELPS 7700 Series uses a three byte instruction queue buffer and performs instruction prefetch to increase processing speed. Therefore, the number of instruction execution cycles depend on the amount of data in the instruction queue buffer. When programming timers, note that the number of cycles shown in the list of machine instructions are for the shortest case. (See "MELPS 7700 <SOFTWARE> User's Manual")

Also, when creating ROMs, note that except when using 16-bit bus width (BYTE="L") and no wait (no software wait or hardware wait due to $\overline{RDY}$ pin) in memory expansion or microprocessor mode, the execution time required to fetch programs from internal ROM is different from that for external ROM.

(2) 16-bit data access

When accessing 16-bit data, the processing speed can be increased by aligning the data on even number address.

7.2.Hardware

7.2.1 Memory related features

(1) Special function registers

The special function registers located at addresses 0_{16} to $7F_{16}$ are used to control internal devices such as timers and serial I/O. The bits of these registers are classified into read only, write only, and read/write bits. An attempt to write to a read only bit is ignored and the result of reading a write only bit is unpredictable (there are exceptions).

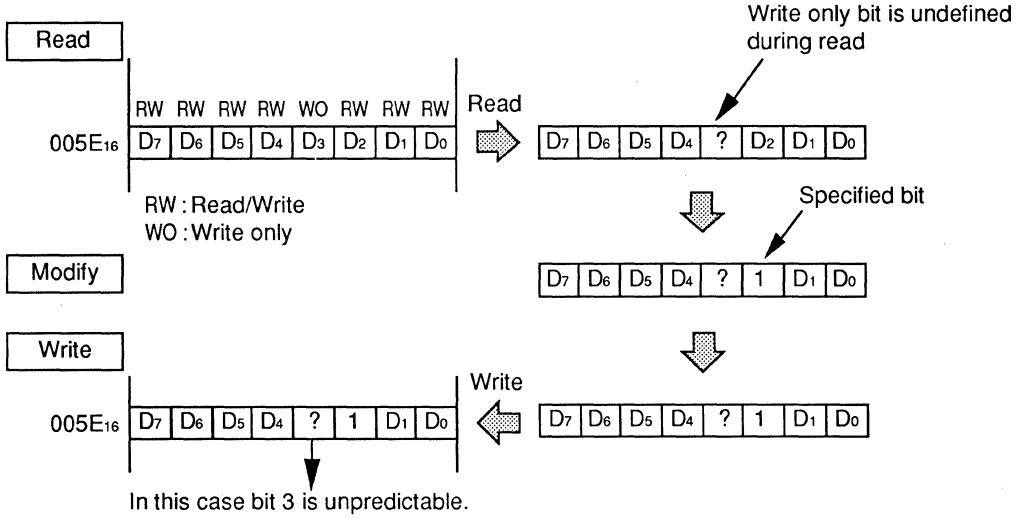
Read-modify-write type instructions such as **CLB** or **SEB** must not be used when modifying the content of a register containing write only bits or when one of the unspecified bits (see Figure 7.2.1) returns an unpredictable result when read. In these cases, use the **LDM** or **STA** instruction.

Note : Do not use a **CLB** or **SEB** instruction for the following registers:

- ●One-shot start flag (Address 42_{16})
- ●Processor mode register (Address $5E_{16}$)

Example) SEB instruction

Mnemonic: SEB #04H,005EH (when m=1)



Note : **ASL**, **CLB**, **DEC**, **INC**, **LSR**, **ROL**, **ROR**, and **SEB** are read-modify-write type instructions.

Fig.7.2.1 Read-modify-write Instruction Execution Sequence

(2) Wait bit

The wait bit is cleared to "0" at reset and program execution starts in one-shot wait mode ($\overline{E}$ output pulse width is doubled during external area access). Single-chip mode is unaffected by this bit, but in memory expansion and microprocessor modes, the wait bit must be switched from a program when externally expanded memory or I/O satisfies the timing specification without wait.

7.2.2 Input/Output pin related features

(1) Double function port

When a double function pin that acts as an input pin to internal device and as a programmable I/O pin is used as input pin to internal device, the data direction register of the corresponding port must be set to input mode before selecting the function.

For ports that are shared as internal device output pin, the pin is forced as output when the function is selected regardless of the content of the corresponding data direction register.

(2) Memory expansion mode and microprocessor mode

In memory expansion and microprocessor mode, the data direction registers for ports $P4_0$ and $P4_1$ must be set to input mode. (All programmable I/O pins are set to input mode at reset.)

7.2.3 Interrupts

The priority detection time selection bits (bits 5 and 4) in the processor mode register are both set to "0" at reset to select the longest interval. The shortest priority detection time can be used for the M37700 family. Therefore, if execution is to be performed immediately after accepting an interrupt, bit 5 should be set to "1" and bit 4 should be set to "0". Furthermore, the interrupt request bit of the M37700 and M37701 can be set and cleared by software.

7.2.4 Timers related features

(1) Reading a timer that is operating

To read a timer that is operating, set flag m to "0" and read all 16 bits of the timer register at once.

(2) One-shot start flag

Bit 7 of the one-shot start flag (42_{16}) must be set to "0" regardless of whether the timer is used or not (this bit is set to "0" at reset).

(3) Up-Down flag

When the two-phase pulse signal processing function (valid only in event counter mode) is not used, bits 7 to 5 of the up-down flag (44_{16}) must be set to "0".

(4) Writing to timer register

When a value is written in the timer while it is operating, the count start flag is cleared to "0" and the timer stops (except in PWM mode). To resume count, set the count start flag of the corresponding timer to "1".

(5) Timer interrupt request timing for timer A and timer B

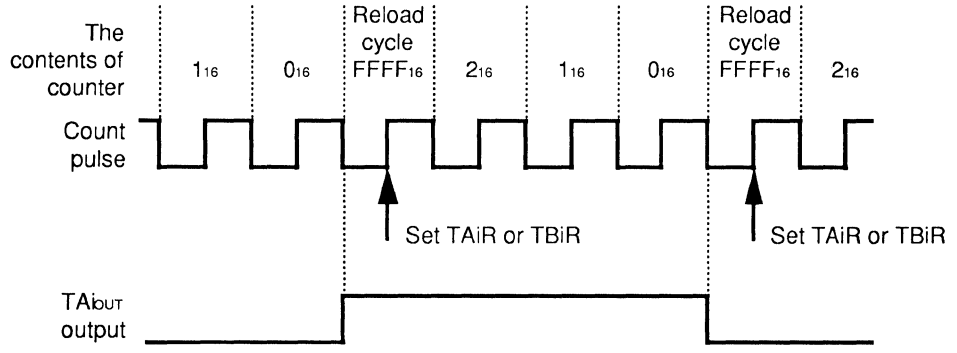
●Timer mode and event counter mode for timer A and timer B

The interrupt request flag is set at the timing shown in Figure 7.2.2.

●Timer A one-shot pulse mode and PWM mode

The interrupt request flag is set as soon as the pulse output from pin TAi_{OUT} falls.

Example) When 0003_{16} is set in the timer register



TAiR: Timer Ai interrupt request flag
TBiR: Timer Bi interrupt request flag

Note: When the timer register is read during a reload cycle, $FFFF_{16}$ is returned during a down count and 0_{16} is returned during an up count.

Fig.7.2.2 Interrupt Request Generation Timing in Timer Mode and Event Counter Mode

(6) Changing counter direction in timer A event counter mode

In event counter mode, the input signal to the TAiOUT pin should be internally synchronized with the event input (input signal to TAiIN pin) so that the counter direction can be changed externally with the input signal from the TAiOUT pin.

- When the falling edge is selected, the input level of the TAiOUT pin is captured while the TAiIN pin input signal is "L" and that level becomes effective when the input signal becomes "H".
- When the rising edge is selected, the input level of the TAiOUT pin is captured while the TAiIN pin input signal is "H" and that level becomes effective when the input signal becomes "L".

Therefore, in case such as shown in Figure 7.2.3, there is a displacement in the counter values (the same is true when the counter direction is changed with software). Thus the recommended relationship between up/down switching input and event input is as follows:

- When the falling edge is selected, switch direction while the TAiIN pin input signal is at "L".
- When the rising edge is selected, switch direction while the TAiIN pin input signal is at "H".

Figure 7.2.4 shows the recommended waveform when the rising edge is selected.
Furthermore, the count start flag should be set after the up/down input level has been established (see Figure 7.2.5).

Example) Event phase:Falling edge
Up/down switching trigger:TAiOUT pin input signal

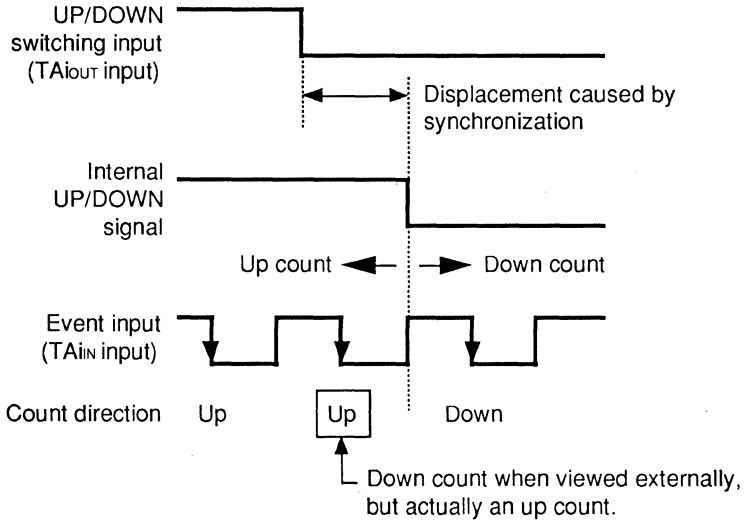


Fig.7.2.3 Displacement Caused by Up/Down Switching

Example) Event phase:Rising edge
Up/down switching trigger:TAiOUT pin input signal

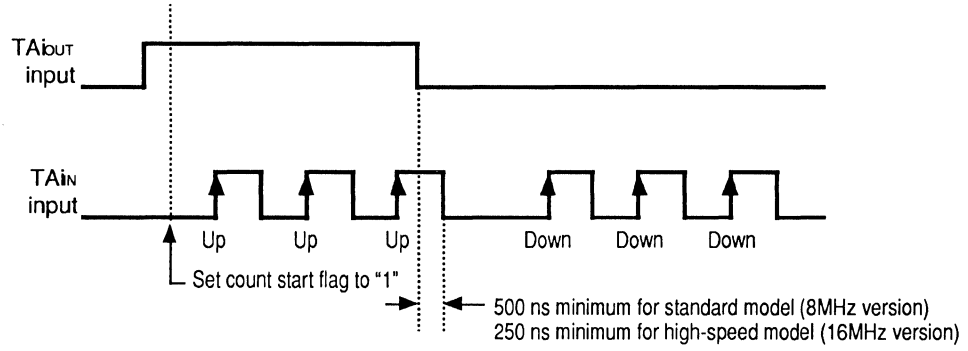
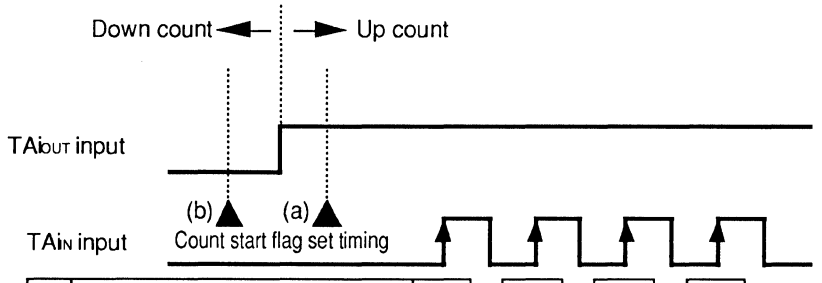


Fig.7.2.4 Recommended Waveform for Up/Down Switching Input and Event Input

<When the rising edge is effective>



(a)	Counter operation when the count start flag is set after the UP/DOWN level is established.	Up	Up	Up	Up
(b)	Counter operation when the count start flag is set before the UP/DOWN level is established.	Down	Up	Up	Up

In case (b), the first effective edge is used as the down count.

Fig.7.2.5 Setting the Count Start Flag

(7) Two-phase pulse signal processing function in timer A event counter mode

When the two-phase pulse signal processing function is selected in event counter mode and the input pulse changes direction at the timing shown in Figure 7.2.6, an error occurs in the counter value.

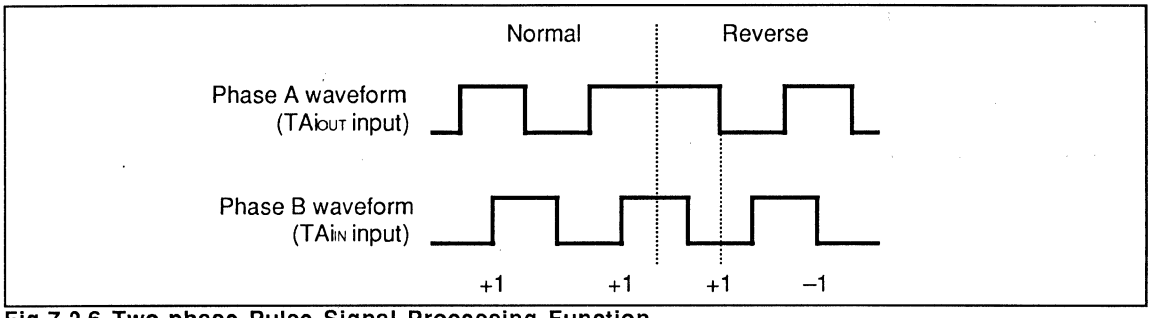


Fig.7.2.6 Two-phase Pulse Signal Processing Function

Therefore, do as follows in a system that requires the two-phase signal processing of input pulse changing direction.

Disable timer Ai two-phase pulse signal processing function, and use the normal event counter mode. Select the falling edge as the effective edge and select an external up/down switching trigger. Externally connect the circuit shown in Figure 7.2.7 and perform the count shown in Figure 7.2.8 using this circuit. Set the "H" and "L" width of phase A and phase B to at least 1 µs.

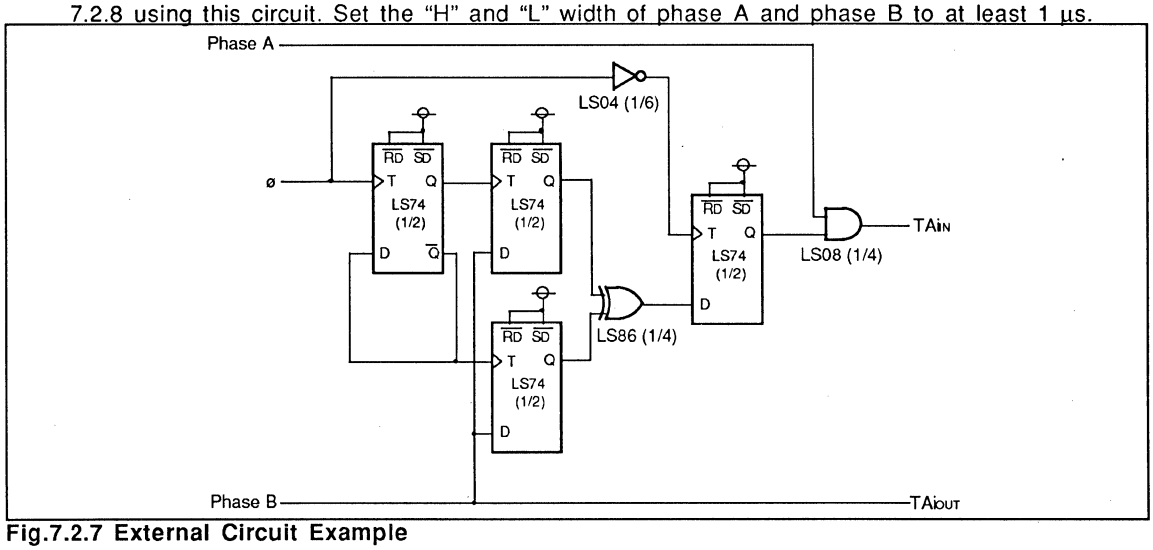


Fig.7.2.7 External Circuit Example

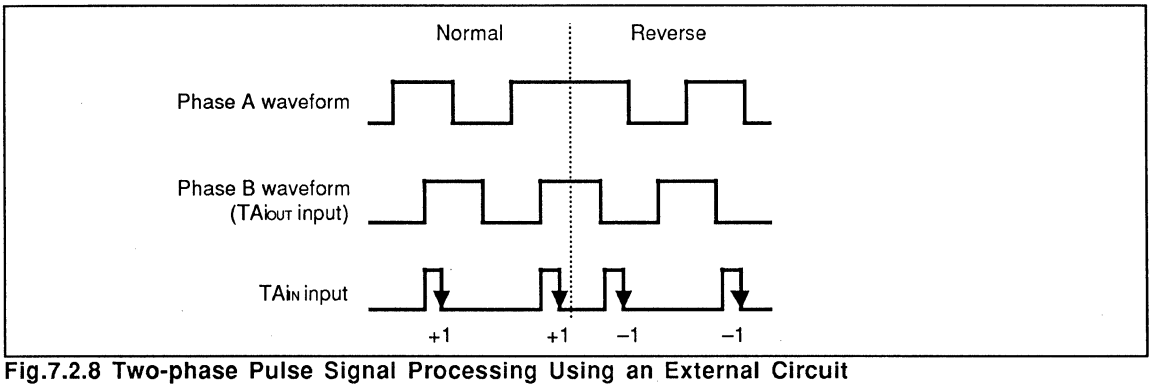


Fig.7.2.8 Two-phase Pulse Signal Processing Using an External Circuit

(8) Timer A one-shot pulse mode

When "00_{16}" is set in low-order 8 bits of the timer Ai register in one-shot pulse mode, pulse output is performed normally at the first start trigger and the pulse width is incorrect for the subsequent triggers. In this case, reset the value immediately after one-shot pulse output, set the count start flag to "1" (count enabled), and wait for the next trigger.

Furthermore, if the timer Ai mode register bit 2 is set to "0" during timer A one-shot pulse output mode, the external output is disabled and the corresponding port can be used as a programmable I/O pin.

(9) Start trigger for Timer A one-shot pulse mode and PWM mode

Note that the start trigger is synchronized to the timer count pulse and if a low frequency count pulse is selected, it may take some time for a pulse to be generated after a trigger is issued.

(10) Updating output pulse width in timer A PWM mode

When changing the content of timer Ai register (changing output pulse width) during PWM output interval, it should be performed while the PWM output is "L". If the content of the timer Ai register is changed while the PWM output is "H", the "L" width becomes the inverse of the updated value as shown in Figure 7.2.9 and the pulse cycle will be different from the rest.

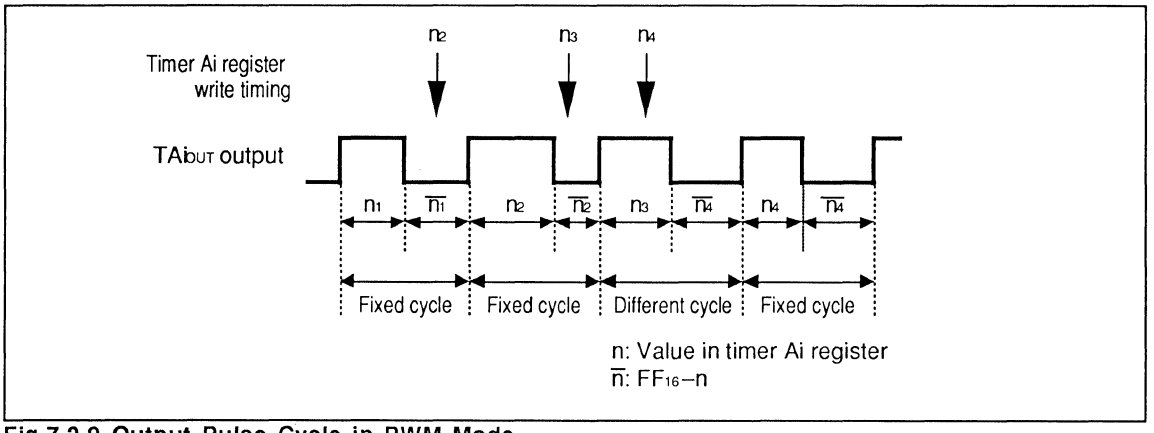


Fig.7.2.9 Output Pulse Cycle in PWM Mode

(11) Pulse output start in timer A PWM mode

The interval between a trigger and the actual PWM output differs between 16-bit PWM mode and 8-bit PWM mode.

●In 16-bit PWM mode

PWM output starts immediately after a trigger as shown in Figure 7.2.10. However, the start trigger is internally synchronized and depending on the count source, it may take some time for the pulse output to start (see section 7.2.4 (9)).

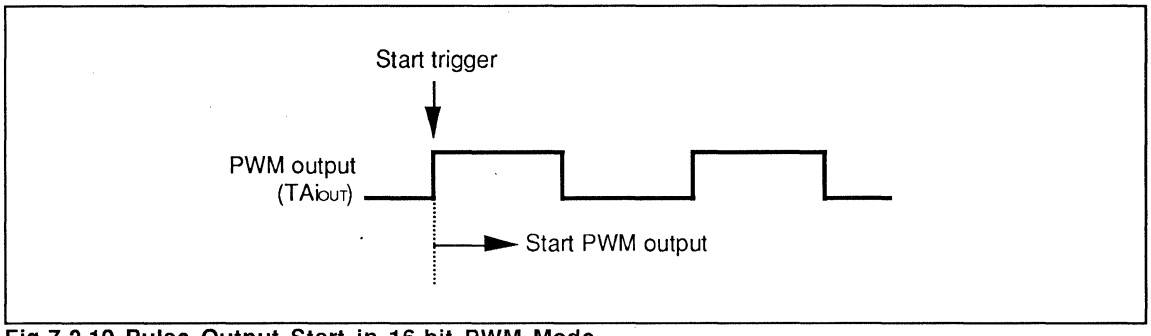


Fig.7.2.10 Pulse Output Start in 16-bit PWM Mode

●In 8-bit PWM mode

PWM output starts after "L" level is output for n to n+1 cycles (proportional to the prescaler value) as shown in Figure 7.2.11.

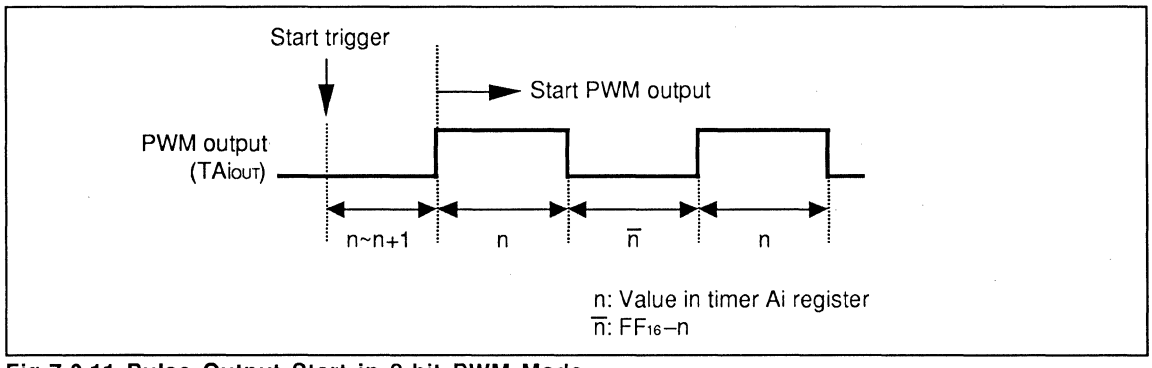


Fig.7.2.11 Pulse Output Start in 8-bit PWM Mode

7.2.5 Serial I/O

(1) Clock synchronous serial I/O

In this mode, the transmitter must be operating even in receive only mode because the receive clock for serial data input is generated by the transmitter.

(2) UART mode

Error detection must be performed before reading the receive buffer register because the error flags in the transmission control register 1 are initialized when the receive buffer register is read.

7.2.6 A-D conversion

If the conversion speed is of priority, set A-D control register bit 7 (A-D conversion frequency selection flag) to "1" from a program because this bit is initialized to "0" at reset and the slowest clock is selected.

7.2.7 Watchdog Timer

When the watchdog timer is disabled by applying 2×Vcc to the $\overline{\text{RESET}}$ pin, a watchdog timer interrupt may occur if the rise of the 10V (2×Vcc) applied to the $\overline{\text{RESET}}$ pin is later than the rise of the 5V power supply voltage. In this case, do as follows:

a) Program to set software reset bit to "1" in the watchdog timer interrupt routine.

b)Set the content of the watchdog timer interrupt vector equal to the reset vector.

7.2.8 Reset Related Features

If the oscillator is stable, the microcomputer can be reset by applying "L" level for a minimum of 2μs to the $\overline{\text{RESET}}$ pin. However, for power-on reset or reset while executing an **STP** instruction, the "L" level must be sufficiently long (approximately 10ms).

7.2.9 Microcomputer Status During Stop, Wait, One-shot Wait, Ready, and Hold

Table 7.2.1 shows the microcomputer status during stop, wait, one-shot wait, ready, and hold.

Table 7.2.1 Microcomputer Status During Stop, Wait, One-shot Wait, Ready, and Hold

Status / Parameter	Enabling condition	Oscillator (Note 1)	ø output	E output	Port status	Watchdog timer status	Status reset
STP instruction (stop mode)	Specify "STP instruction enabled" on the mask ROM confirmation form.	Halted	Halt at "L"	Halt at "H" or "L"	Retain bus and port status when the **STP** instruction is executed (Note 2)	Halt (set "FFF_{16}" in watchdog timer and select count source f_{32})	Reset or external interrupt ($\overline{\text{INT}}$ or serial I/O using external clock, timer A and B with event counter mode)
WIT instruction (wait mode)	Enabled in all modes	Operating	Operating	Halt at "H" or "L"	Retain bus and port status when the **WIT** instruction is executed (Note 2)	Operating	Reset or hardware interrupt
Wait bit (one-shot wait mode)	Access external area with processor mode register bit 2 set to "0"	Operating	Operating	"H" or "L" pulse width is doubled during external area access	———	Operating	Set processor mode register bit 2 to "1"
$\overline{\text{RDY}}$ input (ready status)	During memory expansion mode or microprocessor mode	Operating	Operating	Halt at "H" or "L"	Retain bus and port status when "L" level is applied	Operating	When $\overline{\text{RDY}}$ input returns to "H"
$\overline{\text{HOLD}}$ input (hold status)	During memory expansion mode or microprocessor mode	Operating	Operating	Halt at "H"	Ports P0, P1, P2, $P3_0$, and $P3_1$ are floating. Ports $P3_2$ and $P3_3$ halt at "L". Ports $P4_3$ to $P4_7$, P5, P6, P7, and P8 retain port status when "L" is applied.	Halted	When $\overline{\text{HOLD}}$ input returns to "H"

Note 1 : Timer A, timer B, serial I/O, and A-D converter can be used when oscillating.

Note 2 : If there is an instruction to change the port output or RAM content just before an **STP** or **WIT** instruction, the port output or the RAM content might not be changed. In this case, insert the number of **NOP** instruction, as shows the following, before the **STP** or **WIT** instruction to adjust the execution timing.

- The data is written into internal RAM or SFR ..... One **NOP** instruction
- The data is written into external memory or I/O without one-shot wait ..... One **NOP** instruction
- The data is written into external memory or I/O with one-shot wait ..... Three **NOP** instructions

MEMO

CHAPTER 8
TREATMENT OF INTERNAL PROM TYPE

CHAPTER 8.TREATMENT OF INTERNAL PROM TYPE

8.1 Description

The M37700E2-XXXFP/M37700E4-XXXFP (one-time PROM) and M37700E2FS/M37700E4FS (EPROM) are microcomputers with a built-in programmable ROM (PROM). Each has a high-speed version using a 16MHz external clock. By using a program writing adapter (note 1), normal PROM writers can be used to write programs in the PROM of these microprocessors. Therefore, these microprocessors are suitable for small volume/large variety productions. The one-time PROM type enables programs to be written once. The EPROM type enables programs to be erased by exposing the erase window on top of the package to an ultra-violet light. Therefore, it is suited for program development and prototype use.
Table 8.1.1 shows the PROM types of the M37700 family. A corresponding PROM type is available for each mask ROM type with the same amount of memory. These products have identical functions with only differences among them being the ROM type and supply voltage (EPROM types have different package). The EPROM types use a ceramic LCC package, but the foot pattern can be made equal to an 80-pin plastic molded QFP type by using a special IC socket.

Table 8.1.1 M37700 Family Built-in PROM Types

Built-in PROM Type		External clock	Memory Size (Bytes)		Corresponding mask
One-time PROM type	EPROM type	input frequency	PROM	RAM	ROM type
M37700E2-XXXFP	M37700E2FS	8MHz	16K	512	M37700M2-XXXFP
M37700E2AXXXFP	M37700E2AFS	16MHz			M37700M2AXXXFP
M37700E4-XXXFP	M37700E4FS	8MHz	32K	2K	M37700M4-XXXFP
M37700E4AXXXFP	M37700E4AFS	16MHz			M37700M4AXXXFP

Table 8.1.2 Difference between Built-in PROM Type and Mask ROM Type

Parameter	PROM type	Mask ROM type
ROM type	One-time PROM or EPROM	Mask ROM
Supply voltage V_{CC}	5V±5%	5V±10%

Note 1. The following write adapters are available for the built-in PROM type.

Type	Package	Program writing adapter
One-time PROM type	80-pin plastic molded QFP (80P6)	PCA4707
EPROM type	80-pin ceramic LCC (80D0)	PCA4708

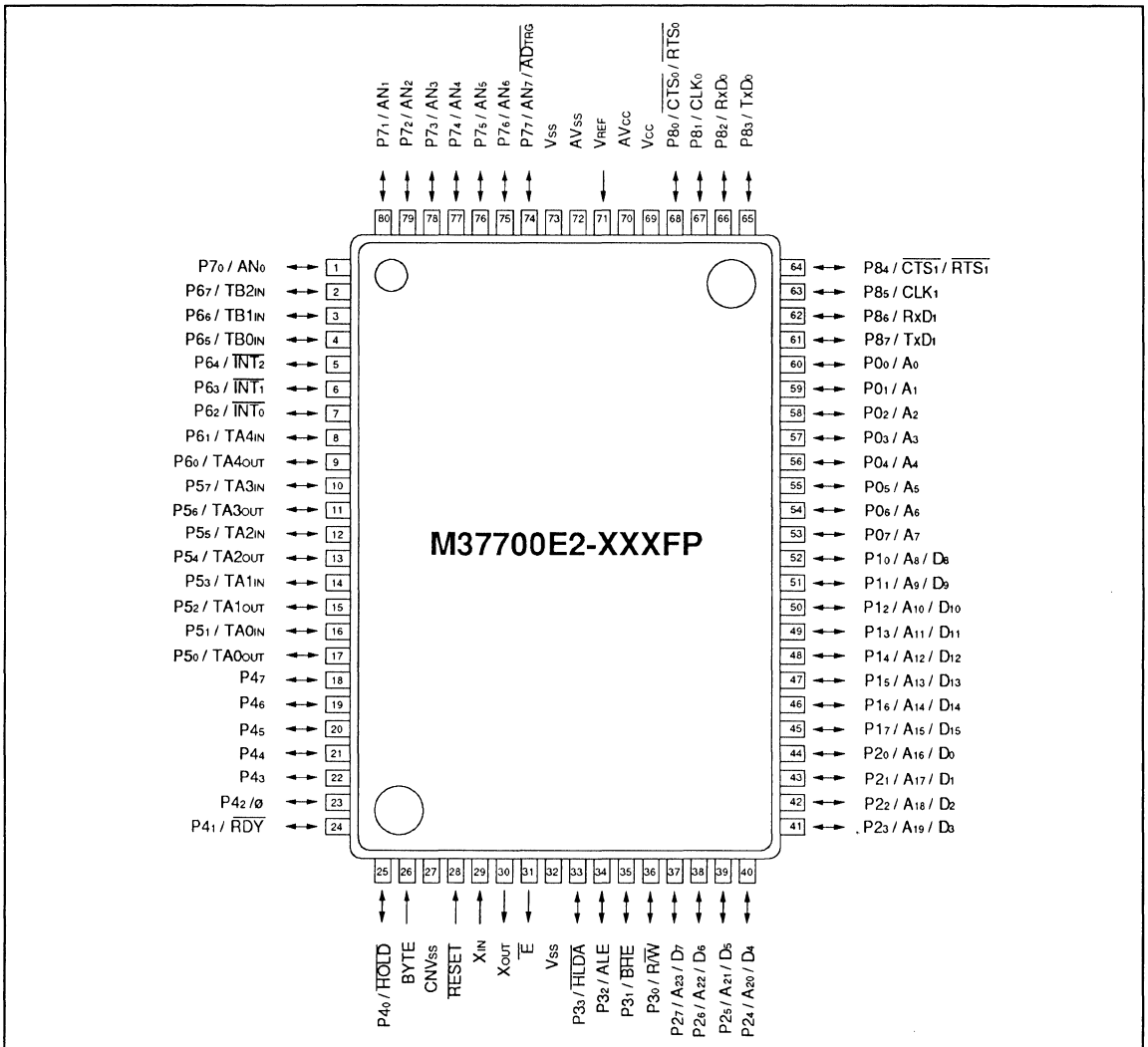
8.2 Functional Description

Figure 8.2.1 shows the pin connection diagram of the built-in PROM type. The pin arrangement is identical to the mask ROM type. The built-in PROM types have a normal operating mode which provides the same functions as the mask ROM type and an EPROM mode used to write to the PROM. The pin functions depend on the mode. In normal operation mode, the pin functions are equivalent to the corresponding mask ROM type. Table 8.2.1 shows the function of each pin in EPROM mode.

Table 8.2.1 Pin Description in EPROM Mode

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5V±5% to V_{CC} and 0V to V_{SS}.
CNV$_{SS}$	V_{PP} input	Input	Connect to V_{PP} when programming or verifying.
BYTE	V_{PP} input	Input	Connect to V_{PP} when programming or verifying.
$\overline{RESET}$	Reset input	Input	Connect to V_{SS}.
X_{IN}	Clock input	Input	Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Open.
AV$_{CC}$, AV$_{SS}$	Analog power supply input		Externally connect AV$_{CC}$ to V_{CC} and AV$_{SS}$ to V_{SS}.

Pin	Name	Input/Output	Functions
V_{REF}	Reference voltage input	Input	Connect to V_{SS}.
$P0_0$~$P0_7$	Address input	Input	The low-order 8-bit (A_7~A_0) address input pins.
$P1_0$~$P1_7$	Address input	Input	$P1_0$~$P1_6$ are high-order 7-bit address input pins. Connect $P1_7$ to V_{CC}.
$P2_0$~$P2_7$	Data input/output	I/O	8-bit data (D_0~D_7) input/output pins.
$P3_0$~$P3_3$	Input port P3	Input	Connect to V_{SS}.
$P4_0$~$P4_7$	Input port P4	Input	Connect to V_{SS}.
$P5_0$~$P5_7$	Control input	Input	$P5_1$ and $P5_2$ function as $\overline{OE}$ and $\overline{CE}$ input. Connect $P5_0$, $P5_3$, $P5_4$, and $P5_5$ to V_{CC}, and $P5_6$ and $P5_7$ to V_{SS}.
$P6_0$~$P6_7$	Input port P6	Input	Connect to V_{SS}.
$P7_0$~$P7_7$	Input port P7	Input	Connect to V_{SS}.
$P8_0$~$P8_7$	Input port P8	Input	Connect to V_{SS}.



Outline 80P6 (One-time PROM type)
80D0 (EPROM type)

Fig.8.2.1 Pin Configuration

8.3 EPROM Mode

The EPROM mode is entered by pulling the $\overline{RESET}$ pin "L". In EPROM mode, ports P0, P1, P2, $P5_1$, $P5_2$ and pins CNV_{SS} and BYTE become EPROM pins (M5M27C256K equivalent) and read/write to built-in PROM can be performed in the same manner as for M5M27C256K. However, there is no device identification code. Therefore, program conditions must be set carefully. X_{IN} and X_{OUT} pins must be connected to a clock (ceramic resonator or an external input).

Table 8.3.1 shows the pin assignments in EPROM mode and Figure 8.3.1 shows the pin connections in EPROM mode.

The program area should specify the following:

Addresses 4000_{16}~$7FFF_{16}$ for M37700E2-XXXFP, M37700E2AXXXFP, M37700E2FS, M37700E2AFS
Addresses 0000_{16}~$7FFF_{16}$ for M37700E4-XXXFP, M37700E4AXXXFP, M37700E4FS, M37700E4AFS

Table 8.3.1 Pin Assignments in EPROM Mode

	Built-in PROM type	M5M27C256K
V_{CC}	V_{CC}	V_{CC}
V_{PP}	CNV_{SS}, BYTE	V_{PP}
V_{SS}	V_{SS}	V_{SS}
Address input	Ports P0, $P1_0$~$P1_6$	A_0~A_{14}
Data I/O	Port P2	D_0~D_7
$\overline{CE}$	$P5_2$	$\overline{CE}$
$\overline{OE}$	$P5_1$	$\overline{OE}$

(1) Read

To read the EPROM, set the $\overline{CE}$ and $\overline{OE}$ pins to "L" level and input the address of the data (A_0~A_{14}) to be read. The data will be output to the I/O pins D_0~D_7. The data I/O pins will be floating when either the $\overline{CE}$ or $\overline{OE}$ pin is at "H".

(2) Write

To write to the EPROM, set the $\overline{OE}$ pin to "H" level. The CPU enters the program mode when V_{PP} is applied to the V_{PP} pin. Set the address to be written to with pins A_0~A_{14} and input the data to be written through pins D_0~D_7. The data is written when the $\overline{CE}$ pin is pulled to "L" level.

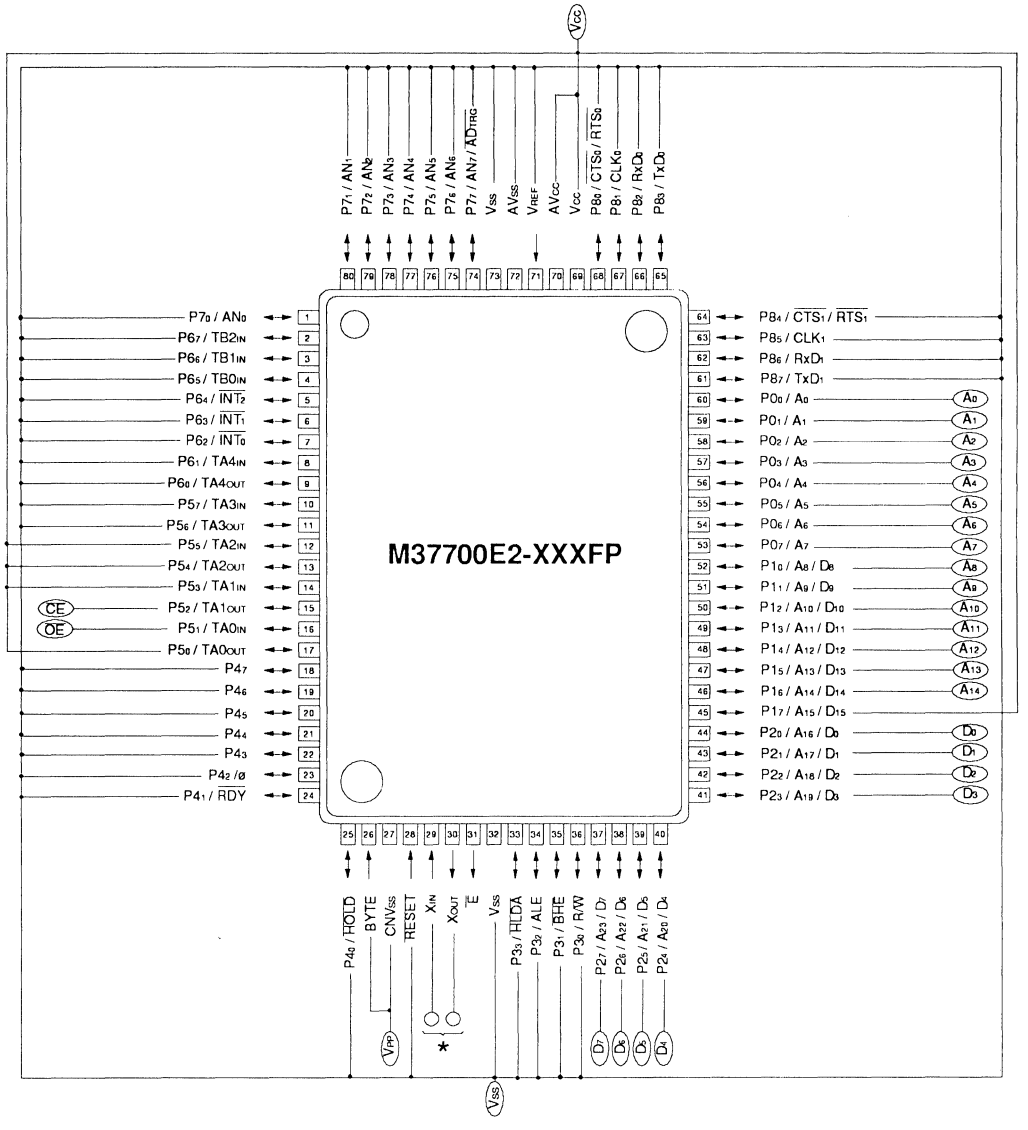
(3) Erase (EPROM type only)

The program is erased by exposing the glass window on top to an ultra-violet light having a wave length of 2537 Angstrom. The light must be at least 15W·S/cm^2.

Table 8.3.2 Input/Output Signals in Each Mode

Mode \ Pin name	$\overline{CE}$	$\overline{OE}$	V_{PP}	V_{CC}	Data I/O
Read-out	V_{IL}	V_{IL}	5V	5V	Output
Output	V_{IL}	V_{IH}	5V	5V	Floating
disable	V_{IH}	X	5V	5V	Floating
Programming	V_{IL}	V_{IH}	12.5V	6V	Input
Programming verify	V_{IH}	V_{IL}	12.5V	6V	Output
Program disable	V_{IH}	V_{IH}	12.5V	6V	Floating

Note: An X indicates either V_{IL} or V_{IH}.



Outline 80P6

* : Connect to ceramic oscillating circuit.

◯ : Same function as EPROM (M5M27C256K).

Fig.8.3.1 Pin connections in EPROM Mode

8.4 Fast Programming Algorithm

To program the built-in PROM type using a fast programming algorithm, first set V_{CC}=6V, V_{PP}=12.5V, and address to 0_{16}. Then apply a 1ms write pulse, check that the data can be read, and if it cannot be read, repeat the procedure until the data can be read. Record the number of pulses applied (N) before the data was read and then write the data again, further applying three times the number of pulses (3 x N ms). When this series of write operation is complete, increment the address and repeat the above procedure until the last address is reached.
Finally, after writing to all addresses, read with V_{CC}=V_{PP}=5V (or V_{CC}=V_{PP}=5.25V).

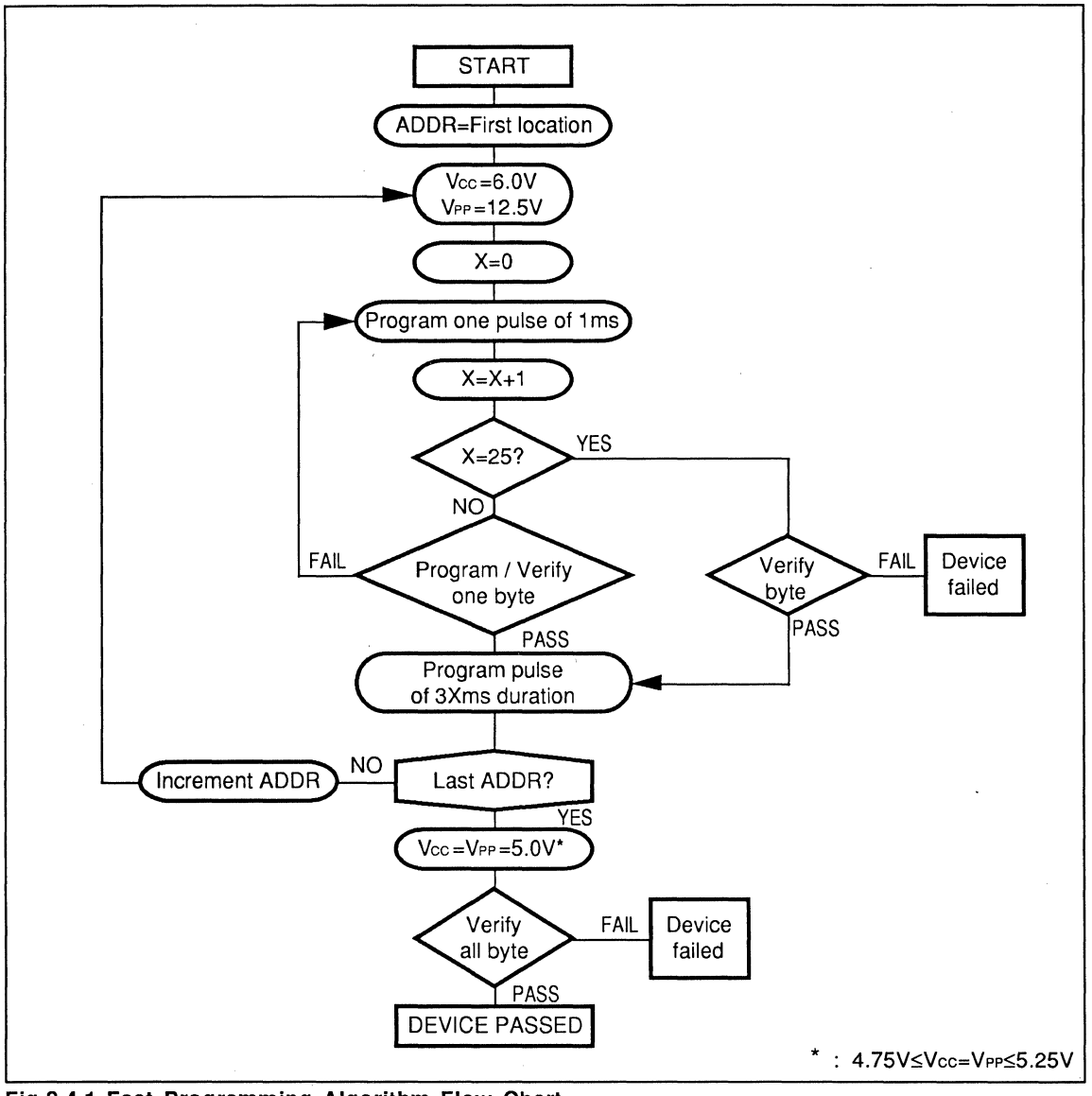


Fig.8.4.1 Fast Programming Algorithm Flow Chart

Fast Program Operation

Table 8.4.1 AC Electrical Characteristics (Ta=25±5°C, V_{CC}=6V±0.25V, V_{PP}=12.5±0.3V, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{AS}	Address setup time	2			µs
t_{OES}	$\overline{OE}$ setup time	2			µs
t_{OS}	Data setup time	2			µs
t_{AH}	Address hold time	0			µs
t_{OH}	Data hold time	2			µs
t_{OFP}	Output enable to output float delay	0		130	ns
t_{VCS}	V_{CC} setup time	2			µs
t_{VPS}	V_{PP} setup time	2			µs
t_{FPW}	$\overline{CE}$ initial program pulse width	0.95	1	1.05	ms
t_{OPW}	$\overline{CE}$ over program pulse width	2.85		78.75	ms
t_{OE}	Data valid from $\overline{OE}$			150	ns

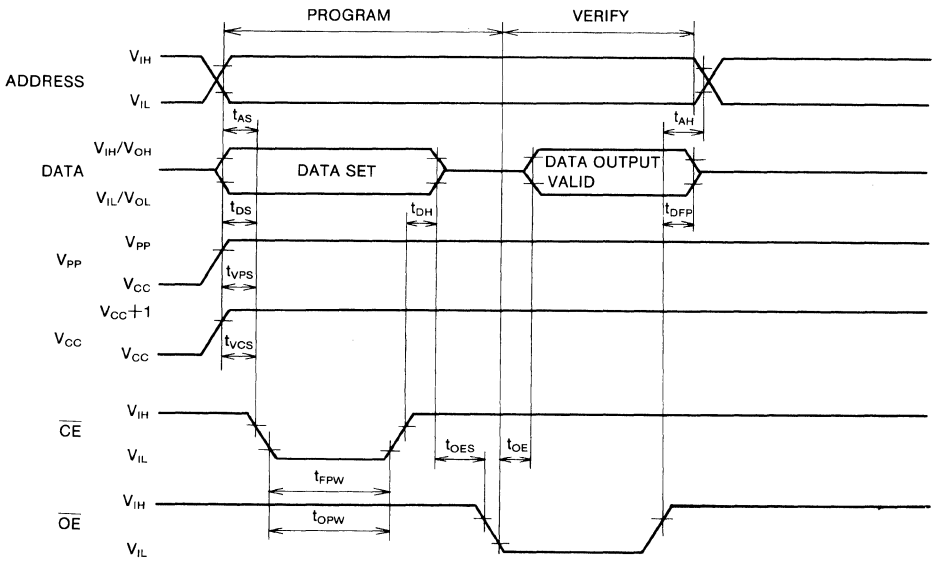


Fig.8.4.2 Fast Programming Timing Diagram

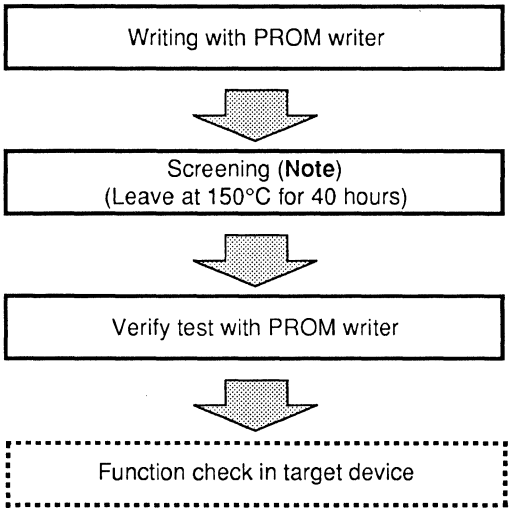
8.5 Notes

8.5.1 Notes on all built-in PROM types

High voltage is required to write to the built-in PROM. However, be careful not to apply excessive voltage. Be especially careful during power-on.

8.5.2 Notes on one-time PROM type

User programmable one-time PROM types (M37700E2FP, M37700E2AFP, M37700E4FP, and M37700E4AFP) are not given write test and screening after assembly.
To improve their reliability after writing, we recommend that they are written and tested as shown in the flow diagram below before use.



Note : The screening time must be less than 100 hours, because the temperature during screening (150°C) is higher than the recommended maximum storage temperature (125°C).

Fig.8.5.1 Recommend Flow Diagram

8.5.3 Notes on EPROM type

- Cover the transparent glass window during read mode because exposing to sun light or fluorescent lamp can cause the information to be erased.
- A shield to cover the transparent window is available from Mitsubishi. Be careful that the shield does not touch the microcomputer lead pins.
- Clean the transparent glass when erasing. If the window is unclean, erasing may be incomplete.
- The EPROM type is enclosed in an 80-pin ceramic LCC package. However, a special IC socket can be used to make the foot pattern similar to the 80-pin QFP type so that it can be used as a prototype for the mask ROM type.

CHAPTER 9
M37701M2-XXXSP

9.1 Description

The M37701M2-XXXFP is a 16-bit single-chip microcomputer designed with high-performance CMOS silicon gate technology. It is housed in a 64-pin shrink plastic molded DIP.

This single-chip microcomputer has a large 16M-byte addressable space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controllers that require high-speed processing of large data.

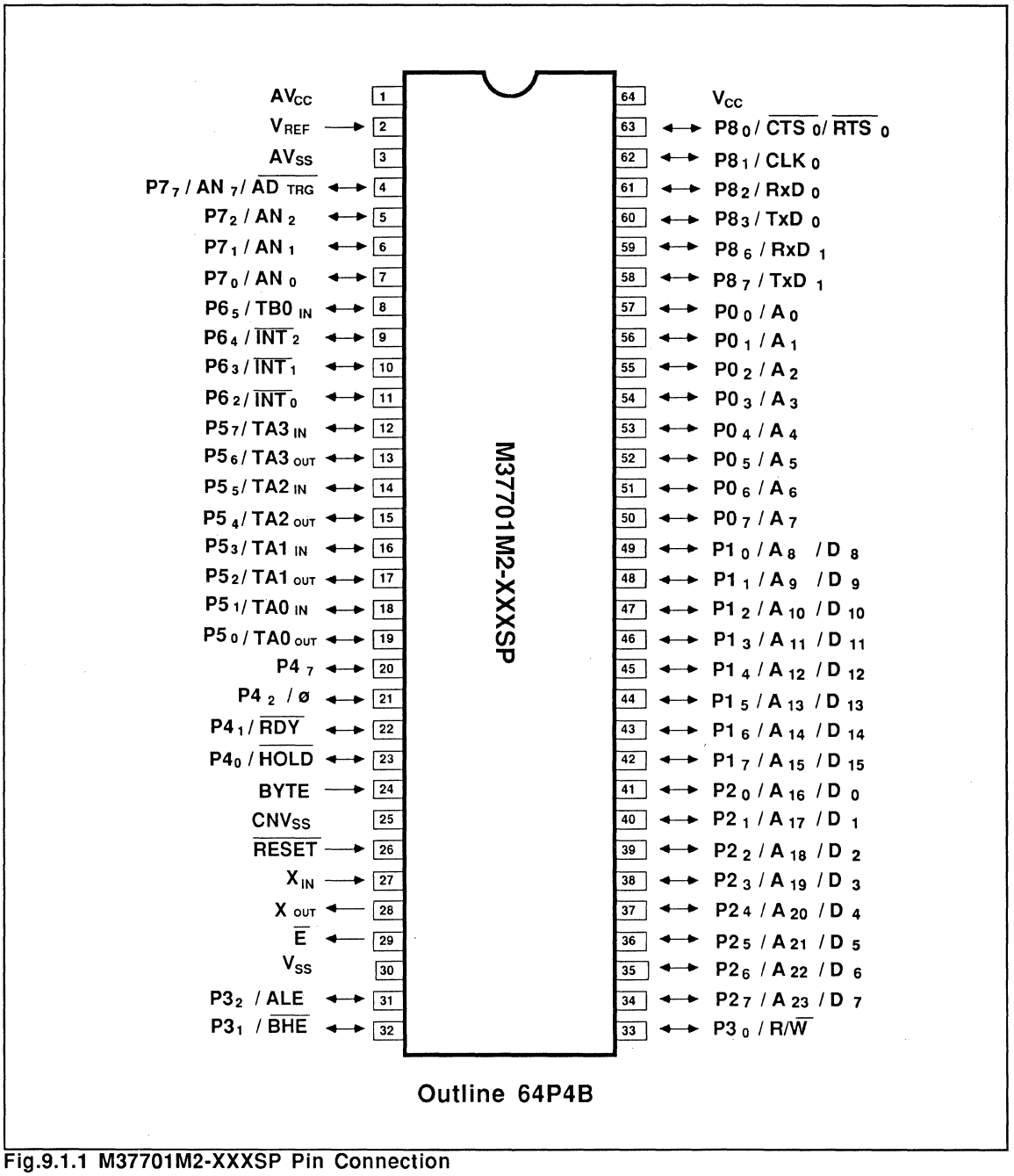


Fig.9.1.1 M37701M2-XXXSP Pin Connection

9.2 M37701 Family

The M37701M2-XXXSP of products is equal to the M37700M2-XXXFP enclosed in a 64-pin shrink plastic molded DIP. The M37701 family includes the types shown in table 9.2.1. All of these types are pin compatible with each other. Only the memory type, size, and operating clock are different. The user can select the element best suited for his use.

Table 9.2.1 M37701 Family

Type name	ROM size (bytes)	RAM size (bytes)	Clock frequency (MHz)
M37701M2-XXXSP	16K (Mask ROM)	512	8
M37701M2AXXXSP	16K (Mask ROM)	512	16
M37701SSP	—	512	8
M37701SASP	—	512	16
M37701E2-XXXSP	16K (One-time PROM)	512	8
M37701E2AXXXSP	16K (One-time PROM)	512	16
M37701M4-XXXSP	32K (Mask ROM)	2048	8
M37701M4AXXXSP	32K (Mask ROM)	2048	16
M37701S4SP	—	2048	8
M37701S4ASP	—	2048	16
M37701E4-XXXSP	32K (One-time PROM)	2048	8
M37701E4AXXXSP	32K (One-time PROM)	2048	16

9.3 M37701M2-XXXSP Characteristics

- Number of basic instructions 103
- Memory size ROM ..... 16K bytes
  RAM ..... 512 bytes
- Instruction execution time (shortest instruction at 8MHz) 500ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8MHz) ..... 30mW (Typ.)
- Interrupts ..... 19 sources, 7 levels
- Multi-function16-bit timers ..... 4+1+3
- Serial I/O 2
- 8-bit A-D converter ..... 4 channel input
- Watchdog timer
- Programmable I/O (ports P0, P1, P2, P3, P4, P5, P6, P7, and P8) 53

9.4 M37701M2-XXXSP Performance Overview

Table 9.4.1 shows the performance overview of the M37701M2-XXXSP.

Table 9.4.1 M37701M2-XXXSP Performance Overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time	M37701M2-XXXSP	500ns (shortest instruction at 8MHz frequency)
	M37701M2AXXXSP	250ns (shortest instruction at 16MHz frequency)
Clock frequency	M37701M2-XXXSP	8MHz (maximum)
	M37701M2AXXXSP	16MHz (maximum)
Memory size	ROM	16384 bytes
	RAM	512 bytes
Input/Output ports	Ports P0, P1, P2, P5	8 bits x 4
	Port P8	6 bits x 1
	Ports P4, P6, P7	4 bits x 3
	Port P3	3 bits x 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16 bits x 5(4 with I/O functions and 1 internal timer)
	TB0, TB1, TB2	16 bits x 3(1 with I/O functions and 2 internal timers)
Serial I/O		Clock asynchronous serial I/O x 2 (UART0 can also be used as clock synchronous)
A-D converter		8 bits x 1 (4 channels)
Watchdog timer		12 bits x 1
Interrupts		3 external, 16 internal (priority levels 0 to 7 can be set for each interrupt with software)
Clock generating circuit		Built-in (externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5V±10%
Power dissipation		30mW (at external 8MHz frequency)
Input/Output characteristics	Input/Output voltage	5V
	Output current	5mA
Memory expansion		Maximum 16M bytes
Operating temperature range		–10 to 70°C
Device structure		CMOS high-performance silicon gate process
Package		64-pin shrink plastic molded DIP

9.5 Differences Between M37701M2-XXXSP and M37700M2-XXXFP

Table 9.5.1 shows the differences between M37701M2-XXXSP and M37700M2-XXXFP.

Table 9.5.1 Differences between M37701M2-XXXSP and M37700M2-XXXFP

Functions			M37701M2-XXXSP	M37700M2-XXXFP
I/O ports			53 (in single-chip mode)	68 (in single-chip mode)
	Port P0		8 bits	8 bits
	Port P1		8 bits	8 bits
	Port P2		8 bits	8 bits
	Port P3		3 bits ($P3_3$/$\overline{HLDA}$ unavailable)	4 bits
	Port P4		4 bits ($P4_3$~$P4_6$ unavailable)	8 bits
	Port P5		8 bits	8 bits
	Port P6		4 bits ($P6_0$, $P6_1$, $P6_6$, and $P6_7$ unavailable)	8 bits
	Port P7		4 bits ($P7_3$~$P7_6$ unavailable)	8 bits
	Port P8		6 bits ($P8_4$ and $P8_5$ unavailable)	8 bits
Timers			16 bits × 8	16 bits × 8
	Timer A	TA0	Timer I/O pins available	Timer I/O pins available
		TA1	Input=TAi_{IN}, output=TAi_{OUT} (i=0~3)	Input=TAi_{IN}, output=TAi_{OUT} (i=0~4)
		TA2		
		TA3		
		TA4	Internal timer ($TA4_{IN}$ and $TA4_{OUT}$ unavailable)	
	Timer B	TB0	Timer input pin ($TB0_{IN}$) available	Timer input pin (TBi_{IN}) available
		TB1	Internal timer ($TB1_{IN}$ and $TB2_{IN}$	(i=0~2)
		TB2	unavailable)	
Serial I/O			2	2
	UART0		Clock asynchronous/synchronous serial I/O	Clock asynchronous/synchronous serial I/O
	UART1		Clock asynchronous serial I/O	Clock asynchronous/synchronous serial I/O
A-D converter			One 8-bit resolution 4-channel analog input pin AN_0, AN_1, AN_2, AN_7 (AN_3~AN_6 unavailable) **Note** : Pin AN_7 is in common with external trigger pin.	One 8-bit resolution 8-channel analog input pin AN_0, AN_1, AN_2, AN_3 AN_4, AN_5, AN_6, AN_7 **Note** : Pin AN_7 is in common with external trigger pin.
Package			64-pin shrink plastic molded DIP (64P4B)	80-pin plastic molded QFP (80P6)

9.6 M37701 Functional Description

The internal circuit of the M37701M2-XXXSP is identical to that of the M37700M2-XXXFP including the control registers and memory allocation in SFR area. However, since the M37701M2-XXXSP has only 64 pins, some I/O pins are not provided externally. Therefore, there are some restrictions in the I/O ports and built-in peripheral device functions. Otherwise it can be used in the same manner as the M37700M2-XXXFP. The precautions when using each function are as follows.

9.6.1 A-D converter

The analog selection bit in the A-D control register must be "000", "001", "010", or "111" because analog inputs are AN_0~AN_2 and AN_7 (four channels).

The A-D conversion time for sweep mode is the same as that of the M37700 family. The contents of A-D registers 3~6 are unpredictable in sweep mode.

9.6.2 Timers

Timers TA4, TB1, and TB2 are internal timers without I/O. Do not select the timer I/O functions for the I/O pins in the timer Ai (i=4) or Bi (i=1, 2) mode register.

9.6.3 Serial I/O

UART1 can only be used in UART mode. It cannot be used in clock synchronous serial I/O mode. Therefore, the serial I/O mode selection bit in the UART1 transmission mode register must not be set to "001".

The $\overline{CTS}/\overline{RTS}$ selection bit in the UART1 transmission control register 0 must be set to "1" (this bit is set to "0" at reset).

9.6.4 Ports

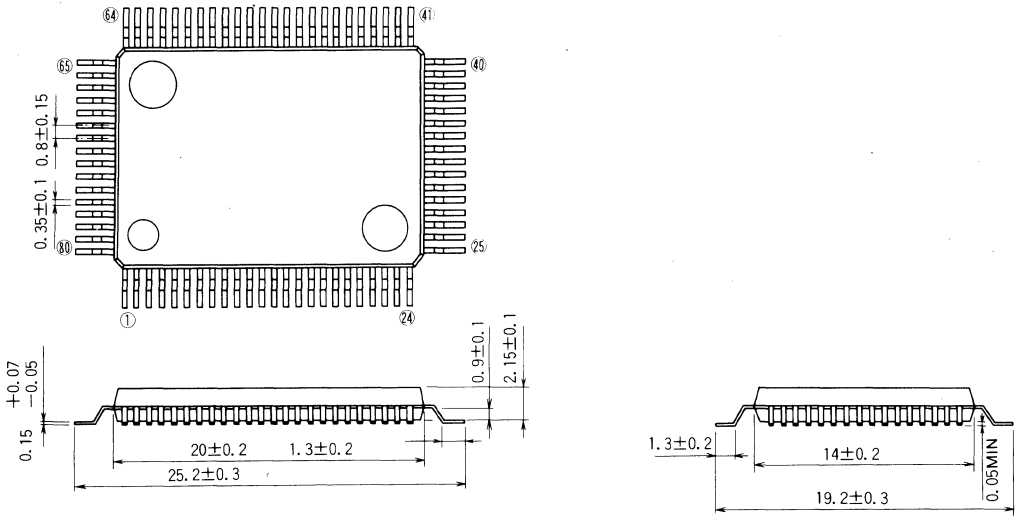
The port registers and data direction registers for ports P4, P6, P7, and P8 contain 8 bits. However, the bits in the data direction register with no corresponding pins must be set to "1" (output) and in the port register must be set to "0" (the data direction register is set to "0" at reset). This is also required when using special functions such as timer I/O.

APPENDIX 1
OUTLINE DRAWING

APPENDIX 1.OUTLINE DRAWING

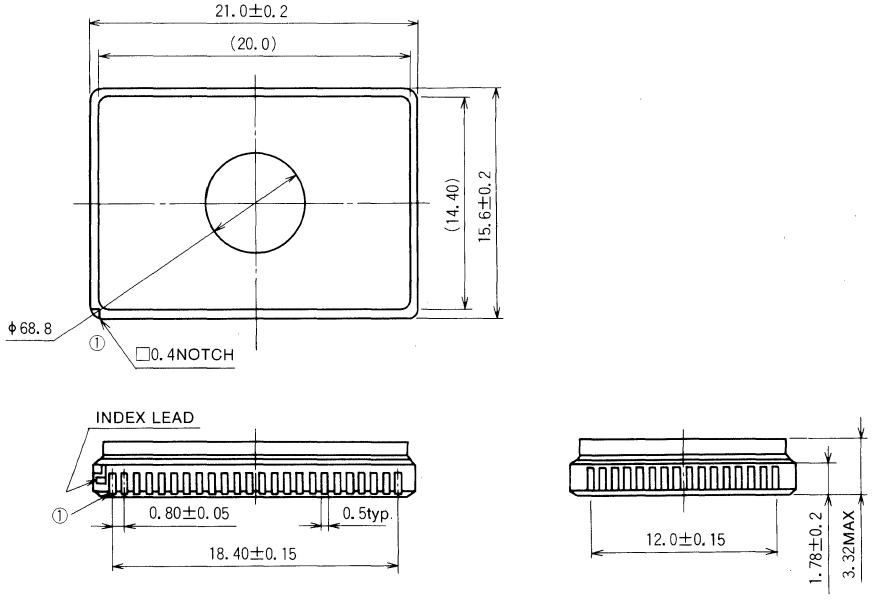
1.80-pin plastic molded QFP (Type name:80P6)

Dimension in mm



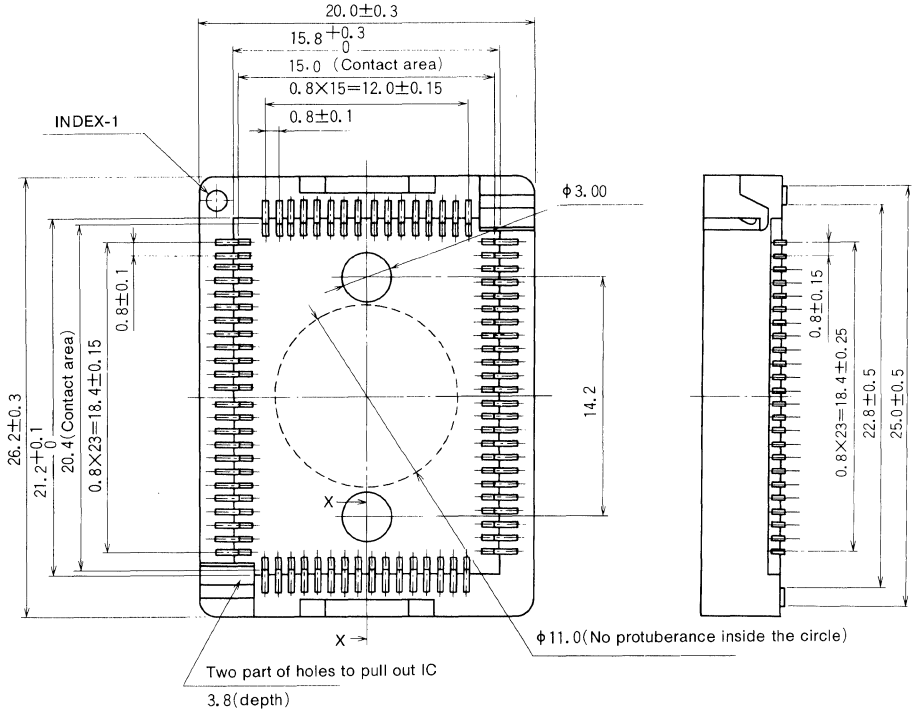
2.80-pin ceramic LCC (Type name:80D0)

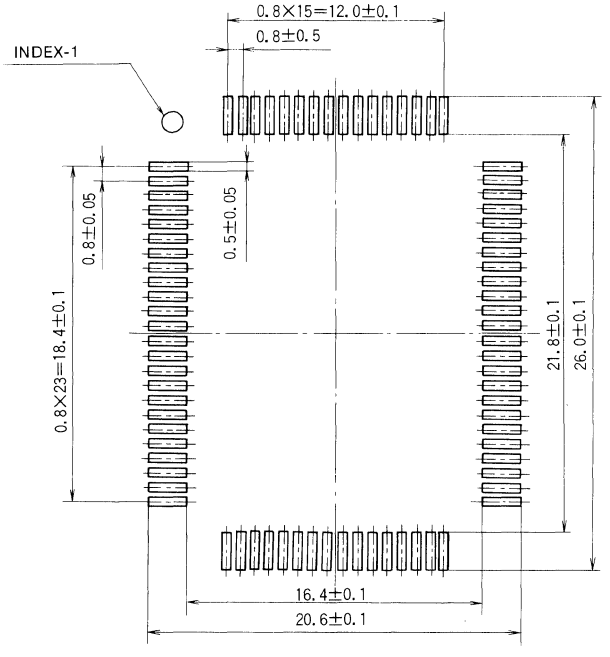
Dimension in mm



3.Outline of EPROM Type IC Socket (YAMAICHI ELECTRIC MFG. CO., LTD, Type name : IC61-0804-034)

Dimension in mm



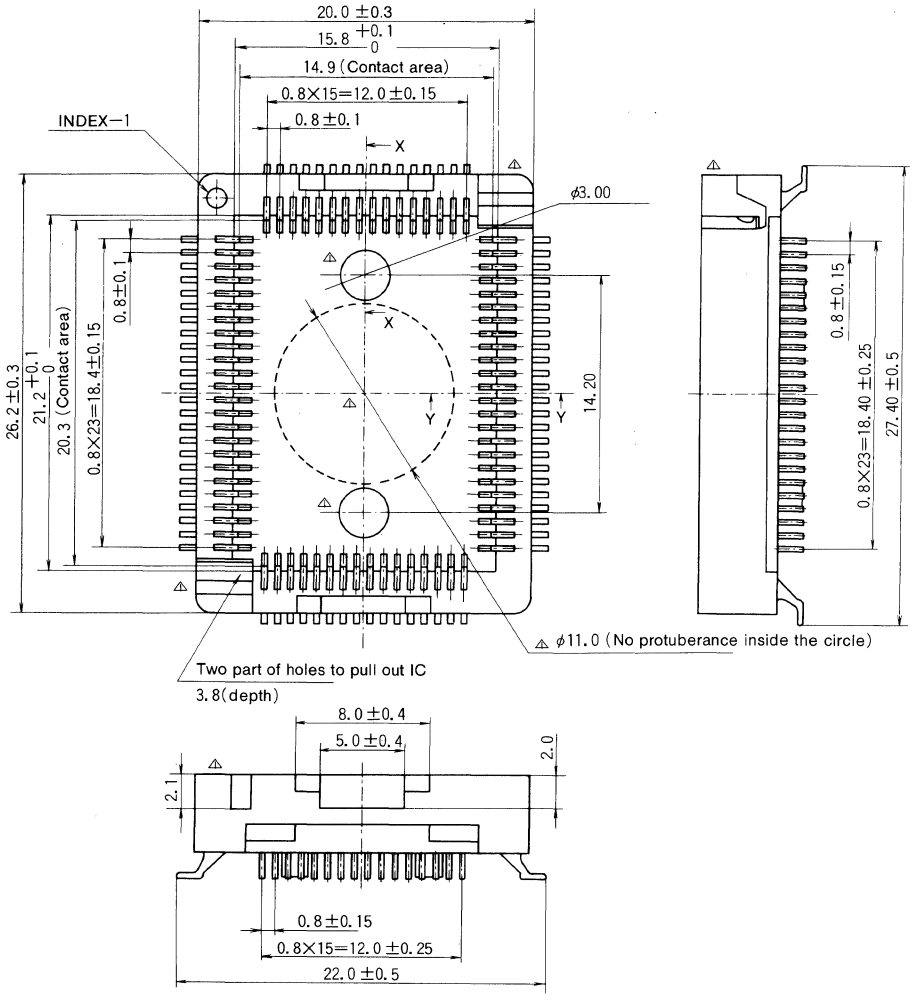


Sample printed circuit foot pattern dimensions

APPENDIX 1.OUTLINE DRAWING

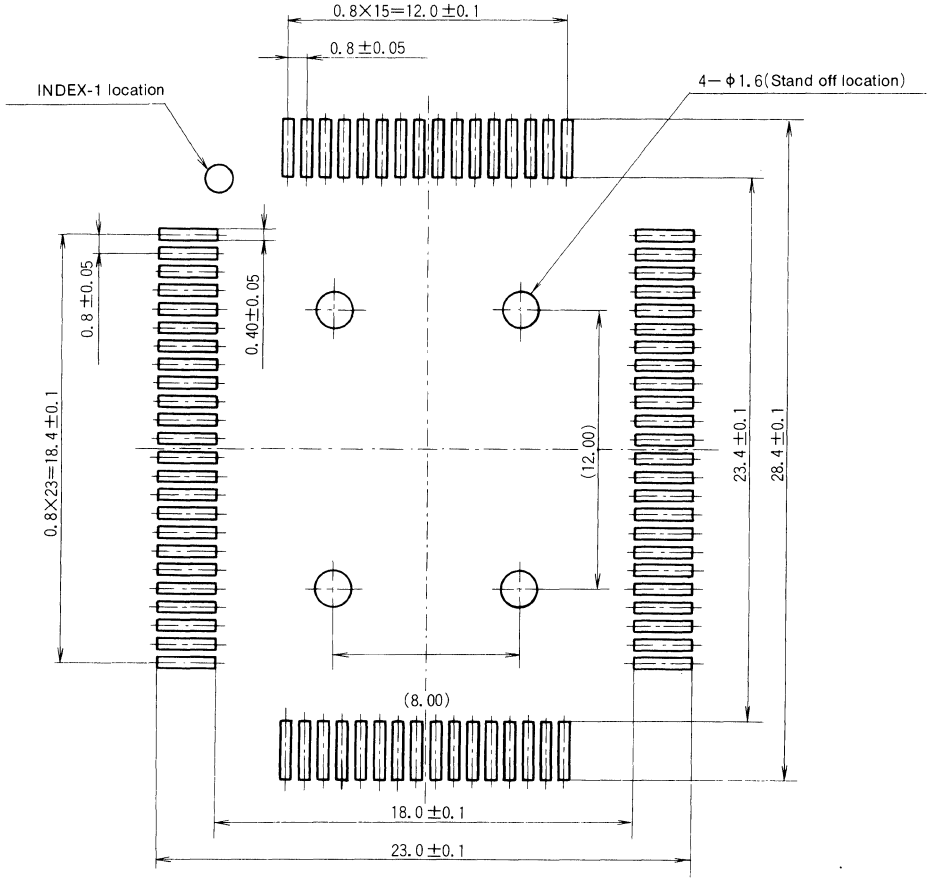
4–1.Outline of EPROM Type IC Socket (MITSUBISHI ELECTRIC CORPORATION, 80LCC-046)

Dimension in mm



4–2.Outline of EPROM Type IC Socket (MITSUBISHI ELECTRIC CORPORATION, 80LCC-046)

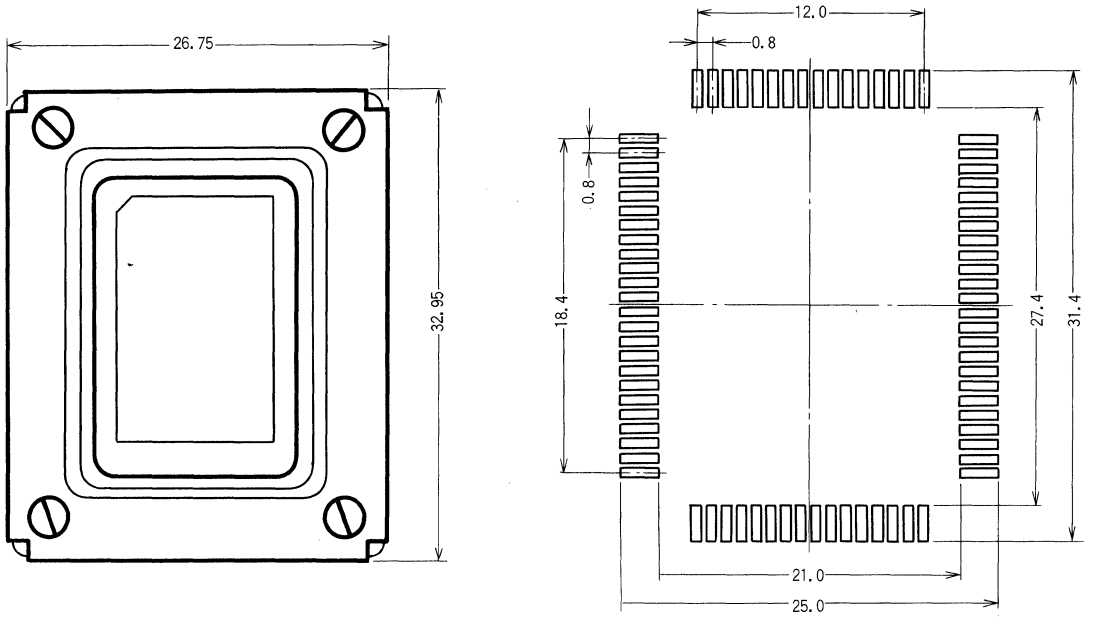
Dimension in mm



Sample printed circuit foot pattern dimensions

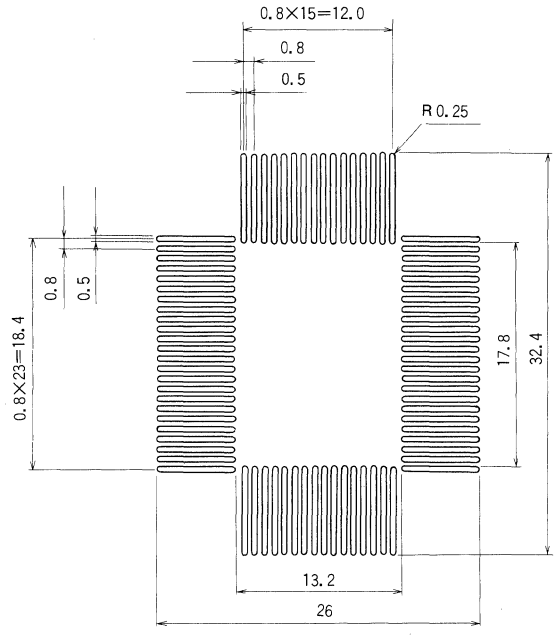
5.Outline of One-time PROM Type IC Socket (YAMAICHI ELECTRIC MFG. CO., LTD, Type name : IC138-080-003-S5)

Dimension in mm



6.Common Printed Circuit Foot Pattern for 80P6, 80P6N, LCC IC Sockets

Dimension in mm



APPENDIX 2
SETTING OF UNUSED PINS

APPENDIX 2.SETTING OF UNUSED PINS

1.Setting of unused pins in single-chip mode

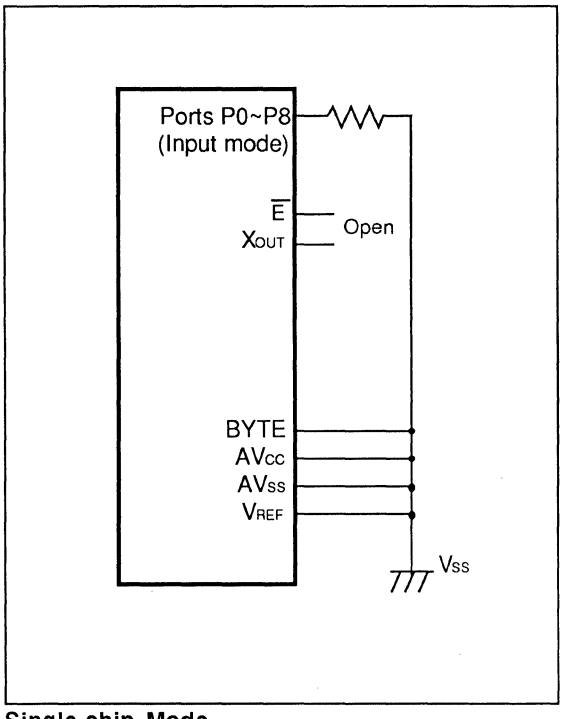
Pin	Setting
Ports P0 to P8	Set to input mode and connect to Vss through a resistor (pull-down)
$\overline{E}$, X_{OUT} (Note 1)	Open
AV_{CC}	Connect to V_{CC} or V_{SS}
AV_{SS}, V_{REF}, BYTE	Connect to V_{SS}

Note 1: When external clock is input to X_{IN}.

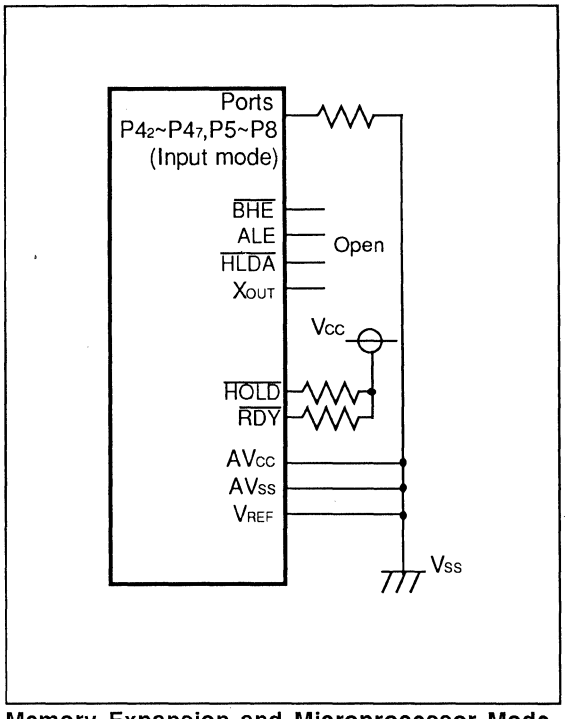
2.Setting of unused pins in memory expansion and microprocessor mode

Pin	Setting
Ports $P4_2$~$P4_7$, P5~P8	Set to input mode and connect to Vss through a resistor (pull-down)
$\overline{BHE}$ (Note 2), ALE (Note 3), $\overline{HLDA}$, X_{OUT} (Note 4)	Open
$\overline{HOLD}$, $\overline{RDY}$	Connect to V_{CC} through a resistor (pull-up)
AV_{CC}	Connect to V_{CC} or V_{SS}
AV_{SS}, V_{REF}	Connect to V_{SS}

Note 2: When BYTE="H".
Note 3: When BYTE="H" and address space is 64K bytes.
Note 4: When external clock is input to X_{IN}.



Single-chip Mode



Memory Expansion and Microprocessor Mode

APPENDIX 3
M37700 FAMILY PERFORMANCE OVERVIEW

APPENDIX 3. M37700 FAMILY PERFORMANCE OVERVIEW

1.M37700 family common performance

The following table shows the performance common to the M37700 family.

Parameters			Functions
Programmable I/O ports			68 in single-chip mode (Note)
Interrupt sources			3 external types, 16 internal types
Built-in devices	Timers	Timer A	16-bit timer x 5
		Timer B	16-bit timer x 3
	Serial I/O		(UART or clock synchronous) x 2
	A-D converter		8-bit A-D converter (8 channel)
	Watchdog timer		12 bit x 1
Clock generating circuit			Built-in (externally connected to a ceramic resonator or quartz crystal resonator)
Power dissipation			30mW (at external 8MHz frequency)
Input/output voltage			5V
Output current			5mA
Memory expansion			16M bytes maximum
Operating temperature range			–10 to 70°C
Device structure			CMOS high-performance silicon gate process

Note: 38 during memory expansion mode and microprocessor mode and for external ROM version.

2.Difference in external clock input frequency by type

All types of the M37700 family are available in 8MHz external clock input version and 16MHz version (high-speed).

External clock input	Mask ROM version	External ROM version	One-time PROM version	EPROM version
8MHz (max.)	M37700M2-XXXFP	M37700SFP	M37700E2-XXXFP	M37700E2FS
	M37700M4-XXXFP	M37700S4FP	M37700E4-XXXFP	M37700E4FS
16MHz (max.)	M37700M2AXXXFP	M37700SAFP	M37700E2AXXXFP	M37700E2AFS
	M37700M4AXXXFP	M37700S4AFP	M37700E4AXXXFP	M37700E4AFS

The shortest instruction execution time .....500ns at $f(X_{IN})$ input frequency 8MHz
250ns at $f(X_{IN})$ input frequency 16MHz (High-speed version)

3. Memory characteristics and size by type

The following table shows the difference in memory characteristics and size by type.

Memory size (Byte)	Mask ROM	One-time PROM	EPROM
ROM...16K	M37700M2-XXXFP	M37700E2-XXXFP	M37700E2FS
RAM...512	M37700M2AXXXFP	M37700E2AXXXFP	M37700E2AFS
ROM...External	M37700SFP	—	—
RAM...512	M37700SAFP	—	—
ROM...32K	M37700M4-XXXFP	M37700E4-XXXFP	M37700E4FS
RAM...2K	M37700M4AXXXFP	M37700E4AXXXFP	M37700E4AFS
ROM...External	M37700S4FP	—	—
RAM...2K	M37700S4AFP	—	—

Note that the operating voltage differs as follows according to the built-in ROM type.

ROM Type	Operating Voltage
Mask ROM type External ROM type	5V±10%
One-time PROM type EPROM type	5V±5%

4.External ROM types

The external ROM types have the same RAM size, function, and electrical characteristics as the following mask ROM size in microprocessor mode.

Type name	Mask ROM type	RAM size	Operating clock
M37700SFP	M37700M2-XXXFP	512 bytes	8MHz
M37700SAFP	M37700M2AXXXFP	512 bytes	16MHz
M37700S4FP	M37700M4-XXXFP	2048 bytes	8MHz
M37700S4AFP	M37700M4AXXXFP	2048 bytes	16MHz

5.Package

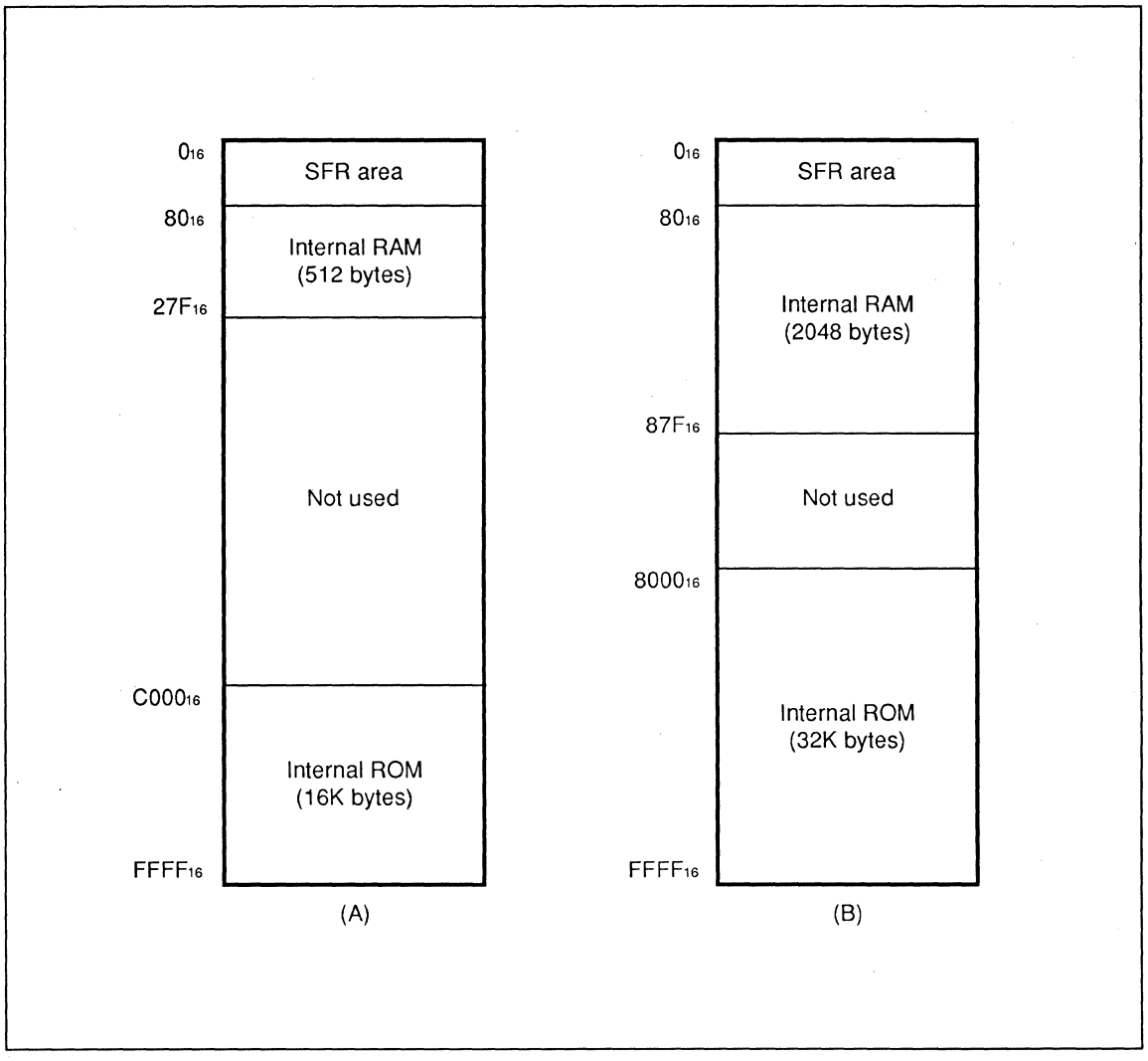
All types except the EPROM type are packaged in an 80-pin plastic molded QFP (type name: 80P6). The EPROM version is packaged in a ceramic LCC (type name: 80D0).

MEMO

APPENDIX 4
M37700 FAMILY MEMORY MAP

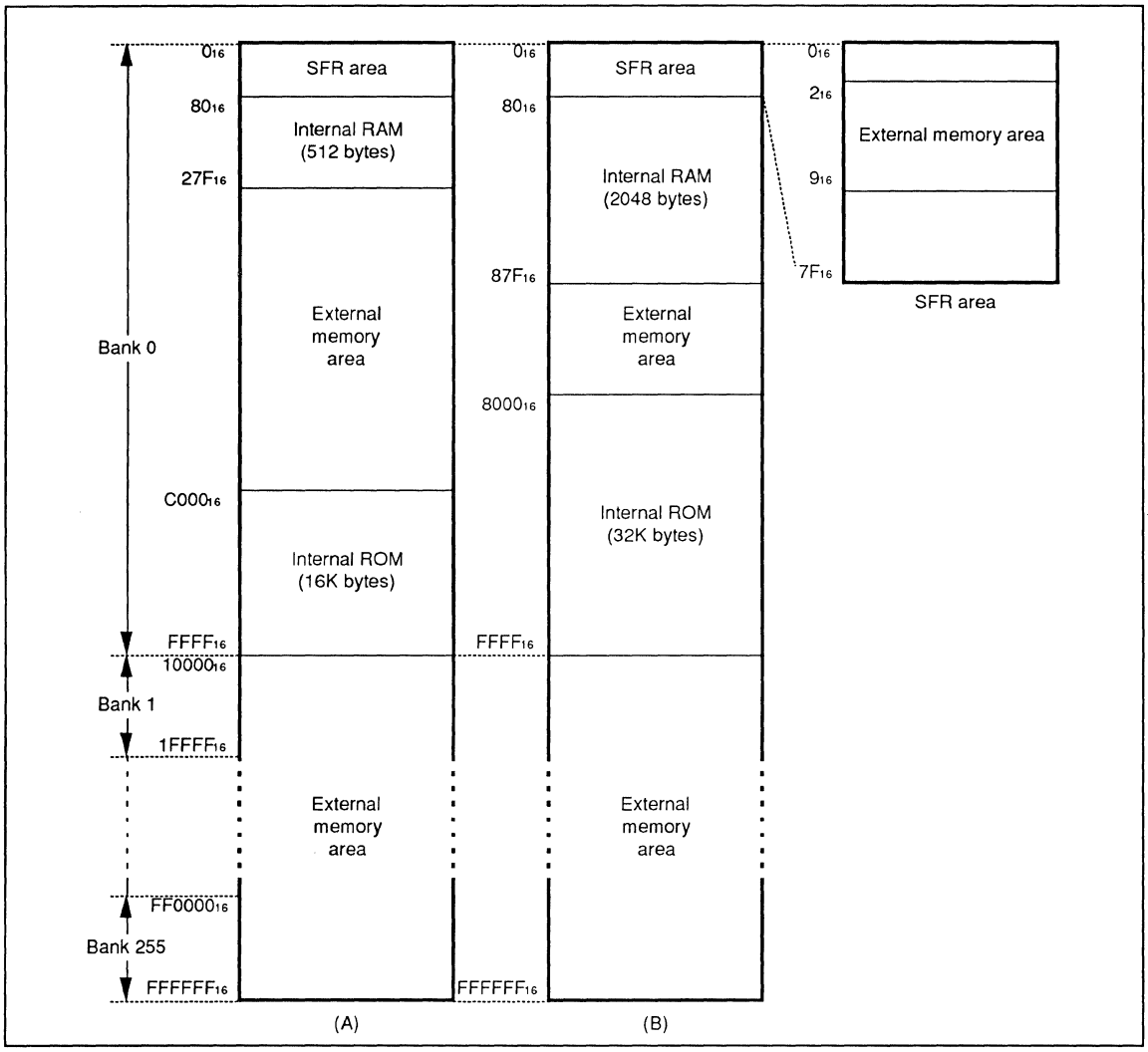
APPENDIX 4. M37700 FAMILY MEMORY MAP

1.Memory map in single-chip mode



No.	Type name	Internal ROM type
(A)	M37700M2-XXXFP/M37700M2AXXXFP	Mask ROM
	M37700E2-XXXFP/M37700E2AXXXFP	One-time PROM
	M37700E2FS/M37700E2AFS	EPROM
(B)	M37700M4-XXXFP/M37700M4AXXXFP	Mask ROM
	M37700E4-XXXFP/M37700E4AXXXFP	One-time PROM
	M37700E4FS/M37700E4AFS	EPROM

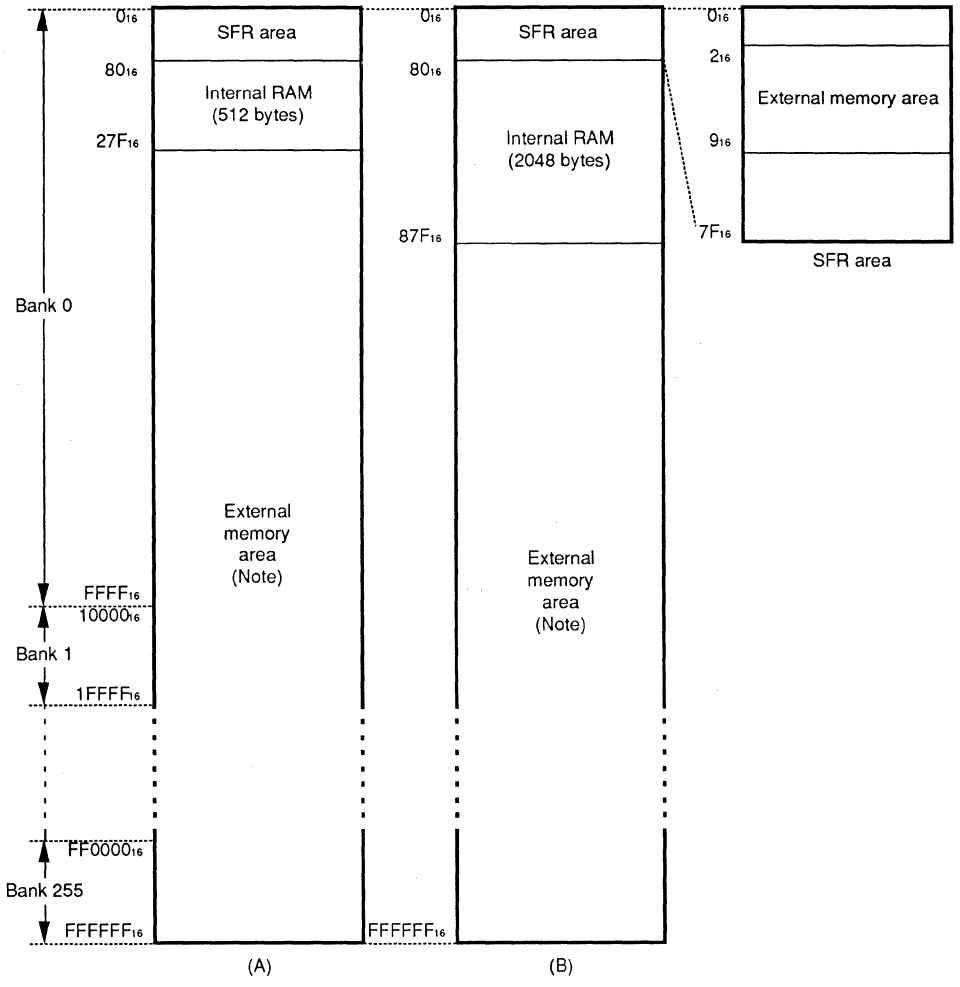
2.Memory map in memory expansion mode



No.	Type name	Internal ROM type
(A)	M37700M2-XXXFP/M37700M2AXXXFP	Mask ROM
	M37700E2-XXXFP/M37700E2AXXXFP	One-time PROM
	M37700E2FS/M37700E2AFS	EPROM
(B)	M37700M4-XXXFP/M37700M4AXXXFP	Mask ROM
	M37700E4-XXXFP/M37700E4AXXXFP	One-time PROM
	M37700E4FS/M37700E4AFS	EPROM

APPENDIX 4. M37700 FAMILY MEMORY MAP

3.Memory map in microprocessor mode (and external ROM type)



Note: Address FFD6$_{16}$ to FFFF$_{16}$ in bank 0 is the interrupt vector area and must be assigned to ROM.

No.	Type name
(A)	M37700SFP/M37700SAFP M37700M2-XXXFP/M37700M2AXXXFP (microprocessor mode) M37700E2-XXXFP/M37700E2AXXXFP (microprocessor mode) M37700E2FS/M37700E2AFS (microprocessor mode)
(B)	M37700S4FP/M37700S4AFP M37700M4-XXXFP/M37700M4AXXXFP (microprocessor mode) M37700E4-XXXFP/M37700E4AXXXFP (microprocessor mode) M37700E4FS/M37700E4AFS (microprocessor mode)

APPENDIX 5
SFR AREA MEMORY MAP

APPENDIX 5.SFR AREA MEMORY MAP

Address (Hexadecimal notation)		Access
000000₁₆		
000001₁₆		
000002₁₆	Port P0 register	RW
000003₁₆	Port P1 register	RW
000004₁₆	Port P0 data direction register	RW
000005₁₆	Port P1 data direction register	RW
000006₁₆	Port P2 register	RW
000007₁₆	Port P3 register	→
000008₁₆	Port P2 data direction register	RW
000009₁₆	Port P3 data direction register	→
00000A₁₆	Port P4 register	RW
00000B₁₆	Port P5 register	RW
00000C₁₆	Port P4 data direction register	RW
00000D₁₆	Port P5 data direction register	RW
00000E₁₆	Port P6 register	RW
00000F₁₆	Port P7 register	RW
000010₁₆	Port P6 data direction register	RW
000011₁₆	Port P7 data direction register	RW
000012₁₆	Port P8 register	RW
000013₁₆		
000014₁₆	Port P8 data direction register	RW
000015₁₆		
00001D₁₆		
00001E₁₆	A-D control register	RW
00001F₁₆		
000020₁₆	A-D register 0	RO
000021₁₆		
000022₁₆	A-D register 1	RO
000023₁₆		
000024₁₆	A-D register 2	RO
000025₁₆		
000026₁₆	A-D register 3	RO
000027₁₆		
000028₁₆	A-D register 4	RO
000029₁₆		
00002A₁₆	A-D register 5	RO
00002B₁₆		
00002C₁₆	A-D register 6	RO
00002D₁₆		
00002E₁₆	A-D register 7	RO
00002F₁₆		

RO: Read only
WO: Write only
RW: Read/Write
?: Definite when read

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

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

The high-order 4 bits of the port 3 register and port P3 data direction register are write prohibited and always return "0" when read.
The **CLB** and **SEB** instructions can be used.

Address (Hexadecimal notation)			Access
000030₁₆	UART 0 transmit/receive mode register		RW
000031₁₆	UART 0 baud rate generator		WO
000032₁₆ 000033₁₆	UART 0 transmission buffer register	L H	WO
000034₁₆	UART 0 transmit/receive control register 0		→
000035₁₆	UART 0 transmit/receive control register 1		→
000036₁₆ 000037₁₆	UART 0 receive buffer register	L H	RO →
000038₁₆	UART 1 transmit/receive mode register		RW
000039₁₆	UART 1 baud rate generator		WO
00003A₁₆ 00003B₁₆	UART 1 transmission buffer register	L H	WO
00003C₁₆	UART 1 transmit/receive control register 0		→
00003D₁₆	UART 1 transmit/receive control register 1		→
00003E₁₆ 00003F₁₆	UART 1 receive buffer register	L H	RO →
000040₁₆	Count start flag		RW
000041₁₆			
000042₁₆	One-shot start flag		WO
000043₁₆			
000044₁₆	Up-down flag		→
000045₁₆			
000046₁₆ 000047₁₆	Timer A0 register	L H	RW
000048₁₆ 000049₁₆	Timer A1 register	L H	RW
00004A₁₆ 00004B₁₆	Timer A2 register	L H	RW
00004C₁₆ 00004D₁₆	Timer A3 register	L H	RW
00004E₁₆ 00004F₁₆	Timer A4 register	L H	RW
000050₁₆ 000051₁₆	Timer B0 register	L H	RW
000052₁₆ 000053₁₆	Timer B1 register	L H	RW
000054₁₆ 000055₁₆	Timer B2 register	L H	RW

The high-order 4 bits of the UARTi transmit/receive control register 0 are write prohibited and are unpredictable when read.

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	RO	RW	RW	RW
RO	RO	RO	RO	RO	RW	RO	RW

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

The UARTi receive buffer register is write prohibited and the high-order 7 bits are always "0" when read.

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	RO	RW	RW	RW
RO	RO	RO	RO	RO	RW	RO	RW

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

b7	b6	b5	b4	b3	b2	b1	b0
WO	WO	WO	RW	RW	RW	RW	RW

The high-order 3 bits of the up-down flag are write only. The **CLB** and **SEB** instructions can be used.

RO: Read only
WO: Write only
RW: Read/Write
?: Definite when read

Address (Hexadecimal notation)		Access
000056₁₆	Timer A0 mode register	RW
000057₁₆	Timer A1 mode register	RW
000058₁₆	Timer A2 mode register	RW
000059₁₆	Timer A3 mode register	RW
00005A₁₆	Timer A4 mode register	RW
00005B₁₆	Timer B0 mode register	RW
00005C₁₆	Timer B1 mode register	RW
00005D₁₆	Timer B2 mode register	RW
00005E₁₆	Processor mode register	→
00005F₁₆		
000060₁₆	Watchdog timer	WO
000061₁₆	Watchdog timer frequency selection flag	RW
000062₁₆		
00006F₁₆		
000070₁₆	A-D conversion interrupt control register	RW
000071₁₆	UART 0 transmission interrupt control register	RW
000072₁₆	UART 0 receive interrupt control register	RW
000073₁₆	UART 1 transmission interrupt control register	RW
000074₁₆	UART 1 receive interrupt control register	RW
000075₁₆	Timer A0 interrupt control register	RW
000076₁₆	Timer A1 interrupt control register	RW
000077₁₆	Timer A2 interrupt control register	RW
000078₁₆	Timer A3 interrupt control register	RW
000079₁₆	Timer A4 interrupt control register	RW
00007A₁₆	Timer B0 interrupt control register	RW
00007B₁₆	Timer B1 interrupt control register	RW
00007C₁₆	Timer B2 interrupt control register	RW
00007D₁₆	$\overline{\text{INT0}}$ interrupt control register	RW
00007E₁₆	$\overline{\text{INT1}}$ interrupt control register	RW
00007F₁₆	$\overline{\text{INT2}}$ interrupt control register	RW
000080₁₆	Internal RAM	

RO: Read only
WO: Write only
RW: Read/Write
?: Definite when read

b7	b6	b5	b4	b3	b2	b1	b0
RW	RW	RW	RW	WO	RW	RW	RW

Bit 3 of the processor mode register is write only. The **CLB** and **SEB** instructions cannot be used to change the content of this register. Use **LDM** or **STA** instructions instead.

APPENDIX 6
CONTROL REGISTERS

APPENDIX 6.CONTROL REGISTERS

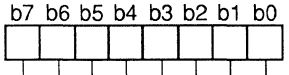
The bit configurations of each control register allocated in the SFR area are shown on the following pages. Each table shows the bit names, functions, content when reset is deactivated, and bit attributes.

* Bit attributes: Each bit in the control register is either read only, write only, or read/write. The following abbreviations are used to indicate the attribute.

  R: read
  W: write
  ○: Allowed
  ×: Not allowed

  If an attempt is made to write to a read only bit, the data is not written. If a write only bit is read, the result is definite.

1.Port Pi data direction registers (i=0~8)

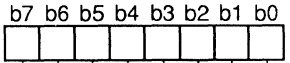


Port P_i data direction register
(Addresses 04_{16},05_{16},08_{16},09_{16},$0C_{16}$,$0D_{16}$,10_{16},11_{16},14_{16})

Bit	Bit name	Functions	At reset	R	W
0	Port P_{i0} data direction register	0:Input mode 1:Output mode	0	○	○
1	Port P_{i1} data direction register		0	○	○
2	Port P_{i2} data direction register		0	○	○
3	Port P_{i3} data direction register		0	○	○
4	Port P_{i4} data direction register		0	○	○
5	Port P_{i5} data direction register		0	○	○
6	Port P_{i6} data direction register		0	○	○
7	Port P_{i7} data direction register		0	○	○

Note: The high-order 4 bits of port P3 data direction register are write prohibited and these bits will always return "0" when read.

2.A-D control register

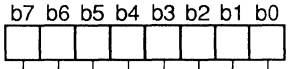


A-D control register (Address $1E_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Analog input selection bits	b2b1b0 0 0 0 :Select AN_0 0 0 1 :Select AN_1 0 1 0 :Select AN_2 0 1 1 :Select AN_3 1 0 0 :Select AN_4 1 0 1 :Select AN_5 1 1 0 :Select AN_6 1 1 1 :Select AN_7 (Note)	0	○	○
1					
2					
3	A-D mode selection bits	b4b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweet mode	0	○	○
4			0	○	○
5	Trigger selection bit	0 : Software trigger (internal trigger) 1 : $\overline{ADTRG}$ input trigger (external trigger)	0	○	○
6	A-D conversion start flag	0 : Stop A-D conversion 1 : Start A-D conversion	0	○	○
7	A-D conversion frequency (ϕ_{AD}) selection flag	0 : Select $f(X_{IN})/8$ 1 : Select $f(X_{IN})/4$	0	○	○

Note: Pin AN_7 cannot be used as analog voltage input pin when an external trigger is selected.

3.UARTi transmit/receive mode registers (i=0, 1)

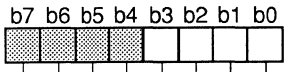


UART0 transmit/receive mode register (Address 30_{16})
UART1 transmit/receive mode register (Address 38_{16})

Bit	Bit name	Functions	At reset	R	W
0	Serial I/O mode selection bits	b2b1b0 0 0 0 :Serial I/O prohibited 0 0 1 :Clock synchronous serial I/O 0 1 0 :This cannot be available 0 1 1 :This cannot be available 1 0 0 :7-bit UART 1 0 1 :8-bit UART 1 1 0 :9-bit UART 1 1 1 :This cannot be available	0	○	○
1					
2					
3	Internal/external clock selection bits	0 : Internal clock 1 : External clock	0	○	○
4	Stop bit length selection bit (in UART mode)	0 : One stop bit 1 : Two stop bits	0	○	○
5	Odd/even parity selection bit (in UART mode)	0 : Odd parity 1 : Even parity	0	○	○
6	Parity enable bit (in UART mode)	0 : Parity disabled 1 : Parity enabled	0	○	○
7	Sleep function selection bit (in UART mode)	0 : Sleep function disabled 1 : Sleep function enabled	0	○	○

Note: Bits 4 to 6 are ignored in clock synchronous mode.
Bit 7 must be "0" when using clock synchronous mode.

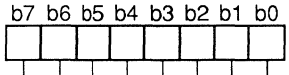
4.UARTi transmit/receive control register 0 (i=0, 1)



UART0 transmit/receive control register 0 (Address 34_{16})
UART1 transmit/receive control register 0 (Address $3C_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	BRG count source selection bits	b1b0 0 0 :Select $f_{(XIN)}/2$ (f_2) 0 1 :Select $f_{(XIN)}/16$ (f_{16}) 1 0 :Select $f_{(XIN)}/64$ (f_{64}) 1 1 :Select $f_{(XIN)}/512$ (f_{512})	0	○	○
1			0	○	○
2	$\overline{CTS}/\overline{RTS}$ selection bit	0 : Select $\overline{CTS}$ 1 : Select $\overline{RTS}$	0	○	○
3	Transmission register empty flag	0 : Data in transmission register (transmitting) 1 : No data in transmission register (transmit complete)	1	○	×
4	These bits cannot be written to because no memory is allocated. The result of reading these bits is indefinite.		Undefined	×	×
5			Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

5.UARTi transmit/receive control register 1 (i=0, 1)

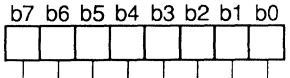


UART0 transmit/receive control register 1 (Address 35_{16})
UART1 transmit/receive control register 1 (Address $3D_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Transmission enable flag	0 : Transmission disable 1 : Transmission enable	0	○	○
1	Transmission buffer empty flag	0 : Data in transmission buffer register 1 : No data in transmission buffer register	1	○	×
2	Receive enable flag	0 : Receive disable 1 : Receive enable	0	○	○
3	Receive completion flag	0 : No data in receive buffer register 1 : Data in receive buffer register	0	○	×
4	Overrun error flag	0 : No overrun error 1 : Overrun error occurred	0	○	×
5	Framing error flag (in UART mode)	0 : No framing error 1 : Framing error occurred	0	○	×
6	Parity error flag (in UART mode)	0 : No parity error 1 : Parity error occurred	0	○	×
7	Error sum flag (in UART mode)	0 : No error 1 : Error occurred	0	○	×

Note: Bits 5 to 7 are ignored in clock synchronous mode.
Each error flag is cleared to "0" when the receive buffer register is read.

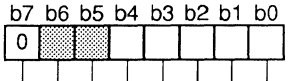
6.Count start flag



Count start flag (Address 40_{16})

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 count start flag	0:Count stop 1:Count start	0	○	○
1	Timer A1 count start flag		0	○	○
2	Timer A2 count start flag		0	○	○
3	Timer A3 count start flag		0	○	○
4	Timer A4 count start flag		0	○	○
5	Timer A5 count start flag		0	○	○
6	Timer A6 count start flag		0	○	○
7	Timer A7 count start flag		0	○	○

7.One-shot start flag

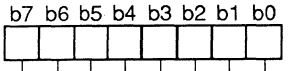


One-shot start flag (Address 42_{16})

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 one-shot start flag	1:One-shot start	0	×	○
1	Timer A1 one-shot start flag		0	×	○
2	Timer A2 one-shot start flag		0	×	○
3	Timer A3 one-shot start flag		0	×	○
4	Timer A4 one-shot start flag		0	×	○
5	These bits cannot be written to because no memory is allocated. The result of reading these bits is unpredictable.		Undefined	×	×
6			Undefined	×	×
7	This bit must be set to "0".		0	×	○

Note: Do not use ready-modify-write type instructions such as **CLB** and **SEB** for these registers. Use **LDM** or **STA** instructions to write to these registers.

8.Up-down flag



Up-down flag (Address 44_{16})

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 up-down flag	0:Down count 1:Up count	0	○	○
1	Timer A1 up-down flag		0	○	○
2	Timer A2 up-down flag		0	○	○
3	Timer A3 up-down flag		0	○	○
4	Timer A4 up-down flag		0	○	○
5	Timer A2 two-phase pulse signal processing selection bit	0:Two-phase pulse signal processing disable 1:Two-phase pulse signal processing enable	0	×	○
6	Timer A3 two-phase pulse signal processing selection bit		0	×	○
7	Timer A4 two-phase pulse signal processing selection bit		0	×	○

Note: The high-order 3 bits of this register are write only, but read-modify-write type instructions such as **CLB** and **SEB** can be used.

9.Timer Ai mode registers (i=0~4)

b7 b6 b5 b4 b3 b2 b1 b0

Timer Ai mode register (Addresses 56_{16}~$5A_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Timer Ai mode selection bits	b1b0 0 0 :Timer mode 0 1 :Event counter mode 1 0 :One-shot pulse mode 1 1 :PWM mode	0	○	○
1			0	○	○
2	The meaning of these bits depend on the timer operation mode.		0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6	Count source selection bits	b7b6 0 0 :Select $f_{(XIN)}/2$ (f_2) 0 1 :Select $f_{(XIN)}/16$(f_{16}) 1 0 :Select $f_{(XIN)}/64$ (f_{64}) 1 1 :Select $f_{(XIN)}/512$ (f_{512})	0	○	○
7			0	○	○

(1)Timer mode

b7 b6 b5 b4 b3 b2 b1 b0

0 0

Timer Ai mode register <Timer mode>

Bit	Bit name	Functions
0	Timer Ai mode selection bits	b1b0 0 0 :Timer mode
1		
2	Pulse output enable bit	0:Disable 1:Enable
3	Gate function selection bit	b4b3 0 X :Gate function disabled 1 0 :Timer operates while TAi_{IN} input is "L" 1 1 :Timer operates while TAi_{IN} input is "H"
4		
5	This bit must be set to "0"	
6	Count source selection bits	b7b6 0 0 :Select $f_{(XIN)}/2$ (f_2) 0 1 :Select $f_{(XIN)}/16$(f_{16}) 1 0 :Select $f_{(XIN)}/64$ (f_{64}) 1 1 :Select $f_{(XIN)}/512$ (f_{512})
7		

(2)Event counter mode

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

Timer Ai mode register <Event counter mode>

Bit	Bit name	Functions
0	Timer Ai mode selection bits	b1b0 0 1 :Event counter mode
1		
2	Pulse output enable bit	0:Disable 1:Enable
3	Count polarity selection bit	0:Count at the falling edge of the input signal 1:Count at the rising edge of the input signal
4	Up-down switching factor selection bit	0:Contents of the up-down flag 1:Input signal of the TAiOUT pin
5	This bit must be set to "0".	
6	These bits are ignored (may be "0" or "1").	
7		

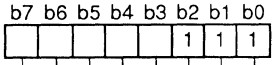
(3)One-shot pulse mode

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

Timer Ai mode register <One-shot pulse mode>

Bit	Bit name	Functions
0	Timer Ai mode selection bits	b1b0 1 0 :One-shot pulse mode
1		
2	This bit must be set to "1".	
3	Trigger selection bits	b4b3 0 X :Internal trigger (software trigger) 1 0 :Trigger at the falling edge of TAiIN input 1 1 :Trigger at the rising edge of TAiIN input
4		
5	This bit must be set to "0".	
6	Count source selection bits	b7b6 0 0 :Select $f_{(XIN)}/2$ (f_2) 0 1 :Select $f_{(XIN)}/16$ (f_{16}) 1 0 :Select $f_{(XIN)}/64$ (f_{64}) 1 1 :Select $f_{(XIN)}/512$ (f_{512})
7		

(4)PWM mode



Timer Ai mode register <PWM mode>

Bit	Bit name	Functions
0 1	Timer Ai mode selection bits	b1b0 1 1 :PWM mode
2	This bit must be set to "1".	
3 4	Trigger selection bits	b4b3 0 X :Internal trigger (software trigger) 1 0 :Trigger at the falling edge of TA_{iIN} input 1 1 :Trigger at the rising edge of TA_{iIN} input
5	16/8-bit pulse modulation width selection bit	0:16-bit width PWM 1:8-bit width PWM
6 7	Count source selection bits	b7b6 0 0 :Select $f_{(XIN)}/2$ (f_2) 0 1 :Select $f_{(XIN)}/16$(f_{16}) 1 0 :Select $f_{(XIN)}/64$ (f_{64}) 1 1 :Select $f_{(XIN)}/512$ (f_{512})

10.Timer Bi mode registers (i=0~2)

b7 b6 b5 b4 b3 b2 b1 b0

Timer Bi mode register (Addresses $5B_{16}$~$5D_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Timer Bi mode selection bits	b1b0 0 0 :Timer mode 0 1 :Event counter mode 1 0 :Pulse cycle/pulse width measurement mode 1 1 :This cannot be available	0	○	○
1			0	○	○
2	The meaning of these bits depend on the timer operation mode.		0	○	○
3			0	○	○
4	These bits cannot be written to because no memory is allocated. The result of reading these bits is unpredictable.		Undefined	×	×
5			Undefined	×	×
6	Count source selection bits	b7b6 0 0 :Select f(XIN)/2 (f_2) 0 1 :Select f(XIN)/16(f_{16}) 1 0 :Select f(XIN)/64 (f_{64}) 1 1 :Select f(XIN)/512 (f_{512})	0	○	○
7			0	○	○

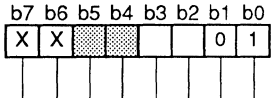
(1)Timer mode]

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

Timer Bi mode register <Timer mode>

Bit	Bit name	Functions
0	Timer Bi mode selection bits	b1b0 0 0 :Timer mode
1		
2	These bits are ignored (may be "0" or "1").	
3		
4	These bits cannot be written to because no memory is allocated. The result of reading these bits is unpredictable.	
5		
6	Count source selection bits	b7b6 0 0 :Select f(XIN)/2 (f_2) 0 1 :Select f(XIN)/16(f_{16}) 1 0 :Select f(XIN)/64 (f_{64}) 1 1 :Select f(XIN)/512 (f_{512})
7		

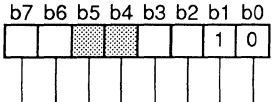
(2)Event counter mode



Timer Bi mode register <Event counter mode>

Bit	Bit name	Functions
0	Timer Bi mode selection bits	b1b0 0 1 :Event counter mode
1		
2	Count polarity selection bits	b3b2 0 0 :Count at the falling edge of the input signal 0 1 :Count at the rising edge of the input signal 1 0 :Count at both edge of the input signal 1 1 :This cannot be available
3		
4	These bits cannot be written to because no memory is allocated. The result of reading these bits is unpredictable.	
5		
6	These bits are ignored (may be "0" or "1").	
7		

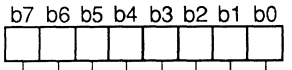
(3)Pulse cycle/pulse width measurement mode



Timer Bi mode register <Pulse cycle/pulse width measurement mode>

Bit	Bit name	Functions
0	Timer Bi mode selection bits	b1b0 1 0 :Pulse cycle/pulse width measurement mode
1		
2	Count polarity selection bits	b3b2 0 0 :Pulse cycle measurement mode (from falling edge to the next falling edge) 0 1 :Pulse cycle measurement mode (from rising edge to the next rising edge) 1 0 :Pulse width measurement mode 1 1 :This cannot be available
3		
4	These bits cannot be written to because no memory is allocated. The result of reading these bits is unpredictable.	
5		
6	Count source selection bits	b7b6 0 0 :Select $f_{(XIN)}/2$ (f_2) 0 1 :Select $f_{(XIN)}/16$(f_{16}) 1 0 :Select $f_{(XIN)}/64$ (f_{64}) 1 1 :Select $f_{(XIN)}/512$ (f_{512})
7		

11.Processor mode register

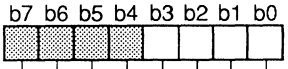


Processor mode register (Address $5E_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Processor mode bits	b1b0 0 0 :Single-chip mode 0 1 :Memory expansion mode 1 0 :Microprocessor mode 1 1 :This cannot be available	0	○	○
1			0	○	○
2	Wait bit	0:Wait during external access 1:No wait	0	○	○
3	Software reset bit	Software reset activated by writing "1".	0	×	○
4	Interrupt priority detection time selection bits	b5b4 0 0 :Select f(XIN)/14 0 1 :Select f(XIN)/8 1 0 :Select f(XIN)/4 1 1 :This cannot be available	0	○	○
5			0	○	○
6	This bit must be set to "0".		0	○	○
7	Internal clock ø output selection bit	0:ø output disabled (P42 is normal I/O port) 1:ø output enable (P42 is ø output pin)	0	○	○

Note: Bit 3 is write only.
Do not use read-modify-write type instructions such as **CLB** or **SEB** for this register. Use **LDM** or **STA** instructions to write to this register.

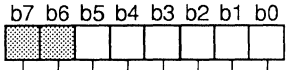
12.A-D conversion, UART 0/1 transmission, UART 0/1 receive, timers A0~A4, timers B0~B2,interrupt control registers



A-D conversion, UART0/1 transmission, UART0/1 receive, timers A0~A4, timers B0~B2 interrupt control registers (Addresses 70_{16}~$7C_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Interrupt priority level selection bits	b2b1b0 0 0 0 :Level 0 (Interrupt disabled) 0 0 1 :Level 1 0 1 0 :Level 2 0 1 1 :Level 3 1 0 0 :Level 4 1 0 1 :Level 5 1 1 0 :Level 6 1 1 1 :Level 7	0	○	○
1			0	○	○
2			0	○	○
3	Interrupt request bit	0:No interrupt request 1:Interrupt request	0	○	○
4	These bits cannot be written to because no memory is allocated. The result of reading these bits is unpredictable.		Undefined	×	×
5			Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

13.$\overline{INT0}$~$\overline{INT2}$ interrupt control registers



$\overline{INT0}$~$\overline{INT2}$ interrupt control registers (Addresses $7D_{16}$~$7F_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Interrupt priority level selection bits	b2b1b0 0 0 0 :Level 0 (Interrupt disabled) 0 0 1 :Level 1 0 1 0 :Level 2 0 1 1 :Level 3 1 0 0 :Level 4 1 0 1 :Level 5 1 1 0 :Level 6 1 1 1 :Level 7	0	○	○
1			0	○	○
2			0	○	○
3	Interrupt request bit	0:No interrupt request 1:Interrupt request	0	○	○
4	Polarity selection bit	0:Set request bit at "H" level for level sense and the falling edge for edge sense 1:Set request bit at "L" level for level sense and the rising edge for edge sense	0	○	○
5	Level/edge sense selection bit	0:Edge sense 1:Level sense	0	○	○
6	These bits cannot be written to because no memory is allocated. The result of reading these bits is unpredictable.		Undefined	×	×
7			Undefined	×	×

MEMO

APPENDIX 7
INSTRUCTION CODE TABLE

APPENDIX 7.INSTRUCTION CODE TABLE

INSTRUCTION CODE TABLE-1

D_7~D_4 \ D_3~D_0	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 A,(DIR,X)		ORA A,SR	SEB DIR,b	ORA A,DIR	ASL DIR	ORA A,L(DIR)	PHP	ORA A,IMM	ASL A	PHD	SEB ABS,b	ORA A,ABS	ASL ABS	ORA A,ABL
0001	1	BPL	ORA A,(DIR),Y	ORA A,(DIR)	ORA A,(SR),Y	CLB DIR,b	ORA A,DIR,X	ASL DIR,X	ORA A,L(DIR),Y	CLC	ORA A,ABS,Y	DEC A	TAS	CLB ABS,b	ORA A,ABS,X	ASL ABS,X	ORA A,ABL,X
0010	2	JSR ABS	AND A,(DIR,X)	JSR ABL	AND A,SR	BBS DIR,b,R	AND A,DIR	ROL DIR	AND A,L(DIR)	PLP	AND A,IMM	ROL A	PLD	BBS ABS,b,R	AND A,ABS	ROL ABS	AND A,ABL
0011	3	BMI	AND A,(DIR),Y	AND A,(DIR)	AND A,(SR),Y	BBC DIR,b,R	AND A,DIR,X	ROL DIR,X	AND A,L(DIR),Y	SEC	AND A,ABS,Y	INC A	TSA	BBC ABS,b,R	AND A,ABS,X	ROL ABS,X	AND A,ABL,X
0100	4	RTI	EOR A,(DIR,X)	Note 1	EOR A,SR	MVP	EOR A,DIR	LSR DIR	EOR A,L(DIR)	PHA	EOR A,IMM	LSR A	PHG	JMP ABS	EOR A,ABS	LSR ABS	EOR A,ABL
0101	5	BVC	EOR A,(DIR),Y	EOR A,(DIR)	EOR A,(SR),Y	MVN	EOR A,DIR,X	LSR DIR,X	EOR A,L(DIR),Y	CLI	EOR A,ABS,Y	PHY	TAD	JMP ABL	EOR A,ABS,X	LSR ABS,X	EOR A,ABL,X
0110	6	RTS	ADC A,(DIR,X)	PER	ADC A,SR	LDM DIR	ADC A,DIR	ROR DIR	ADC A,L(DIR)	PLA	ADC A,IMM	ROR A	RTL	JMP (ABS)	ADC A,ABS	ROR ABS	ADC A,ABL
0111	7	BVS	ADC A,(DIR),Y	ADC A,(DIR)	ADC A,(SR),Y	LDM DIR,X	ADC A,DIR,X	ROR DIR,X	ADC A,L(DIR),Y	SEI	ADC A,ABS,Y	PLY	TDA	JMP (ABS,X)	ADC A,ABS,X	ROR ABS,X	ADC A,ABL,X
1000	8	BRA REL	STA A,(DIR,X)	BRA REL	STA A,SR	STY DIR	STA A,DIR	STX DIR	STA A,L(DIR)	DEY	Note 2	TXA	PHT	STY ABS	STA A,ABS	STX ABS	STA A,ABL
1001	9	BCC	STA A,(DIR),Y	STA A,(DIR)	STA A,(SR),Y	STY DIR,X	STA A,DIR,X	STX DIR,Y	STA A,L(DIR),Y	TYA	STA A,ABS,Y	TXS	TXY	LDM ABS	STA A,ABS,X	LDM ABS,X	STA A,ABL,X
1010	A	LDY IMM	LDA A,(DIR,X)	LDX IMM	LDA A,SR	LDY DIR	LDA A,DIR	LDX DIR	LDA A,L(DIR)	TAY	LDA A,IMM	TAX	PLT	LDY ABS	LDA A,ABS	LDX ABS	LDA A,ABL
1011	B	BCS	LDA A,(DIR),Y	LDA A,(DIR)	LDA A,(SR),Y	LDY DIR,X	LDA A,DIR,X	LDX DIR,Y	LDA A,L(DIR),Y	CLV	LDA A,ABS,Y	TSX	TYX	LDY ABS,X	LDA A,ABS,X	LDX ABS,Y	LDA A,ABL,X
1100	C	CPY IMM	CMP A,(DIR,X)	CLP IMM	CMP A,SR	CPY DIR	CMP A,DIR	DEC DIR	CMP A,L(DIR)	INY	CMP A,IMM	DEX	WIT	CPY ABS	CMP A,ABS	DEC ABS	CMP A,ABL
1101	D	BNE	CMP A,(DIR),Y	CMP A,(DIR)	CMP A,(SR),Y	PEI	CMP A,DIR,X	DEC DIR,X	CMP A,L(DIR),Y	CLM	CMP A,ABS,Y	PHX	STP	JMP L(ABS)	CMP A,ABS,X	DEC ABS,X	CMP A,ABL,X
1110	E	CPX IMM	SBC A,(DIR,X)	SEP IMM	SBC A,SR	CPX DIR	SBC A,DIR	INC DIR	SBC A,L(DIR)	INX	SBC A,IMM	NOP	PSH	CPX ABS	SBC A,ABS	INC ABS	SBC A,ABL
1111	F	BEQ	SBC A,(DIR),Y	SBC A,(DIR)	SBC A,(SR),Y	PEA	SBC A,DIR,X	INC DIR,X	SBC A,L(DIR),Y	SEM	SBC A,ABS,Y	PLX	PUL	JSR (ABS,X)	SBC A,ABS,X	INC ABS,X	SBC A,ABL,X

Note 1 : 42_{16} specifies the contents of the INSTRUCTION CODE TABLE-2.
About the second word's codes, refer to the INSTRUCTION CODE TABLE-2.

Note 2 : 89_{16} specifies the contents of the INSTRUCTION CODE TABLE-3.
About the third word's codes, refer to the INSTRUCTION CODE TABLE-2.

INSTRUCTION CODE TABLE-2 (The first word's code of each instruction is 42_{16})

$D_7 \sim D_4$ \ $D_3 \sim D_0$	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		ORA B,(DIR,X)		ORA B,SR		ORA B,DIR		ORA B,L(DIR)		ORA B,IMM	ASL B			ORA B,ABS		ORA B,ABL
0001	1		ORA B,(DIR),Y	ORA B,(DIR)	ORA B,(SR),Y		ORA B,DIR,X		ORA B,L(DIR),Y		ORA B,ABS,Y	DEC B	TBS		ORA B,ABS,X		ORA B,ABL,X
0010	2		AND B,(DIR,X)		AND B,SR		AND B,DIR		AND B,L(DIR)		AND B,IMM	ROL B			AND B,ABS		AND B,ABL
0011	3		AND B,(DIR),Y	AND B,(DIR)	AND B,(SR),Y		AND B,DIR,X		AND B,L(DIR),Y		AND B,ABS,Y	INC B	TSB		AND B,ABS,X		AND B,ABL,X
0100	4		EOR B,(DIR,X)		EOR B,SR		EOR B,DIR		EOR B,L(DIR)	PHB	EOR B,IMM	LSR B			EOR B,ABS		EOR B,ABL
0101	5		EOR B,(DIR),Y	EOR B,(DIR)	EOR B,(SR),Y		EOR B,DIR,X		EOR B,L(DIR),Y		EOR B,ABS,Y		TBD		EOR B,ABS,X		EOR B,ABL,X
0110	6		ADC B,(DIR,X)		ADC B,SR		ADC B,DIR		ADC B,L(DIR)	PLB	ADC B,IMM	ROR B			ADC B,ABS		ADC B,ABL
0111	7		ADC B,(DIR),Y	ADC B,(DIR)	ADC B,(SR),Y		ADC B,DIR,X		ADC B,L(DIR),Y		ADC B,ABS,Y		TDB		ADC B,ABS,X		ADC B,ABL,X
1000	8		STA B,(DIR,X)		STA B,SR		STA B,DIR		STA B,L(DIR)			TXB			STA B,ABS		STA B,ABL
1001	9		STA B,(DIR),Y	STA B,(DIR)	STA B,(SR),Y		STA B,DIR,X		STA B,L(DIR),Y	TYB	STA B,ABS,Y				STA B,ABS,X		STA B,ABL,X
1010	A		LDA B,(DIR,X)		LDA B,SR		LDA B,DIR		LDA B,L(DIR)	TBY	LDA B,IMM	TBX			LDA B,ABS		LDA B,ABL
1011	B		LDA B,(DIR),Y	LDA B,(DIR)	LDA B,(SR),Y		LDA B,DIR,X		LDA B,L(DIR),Y		LDA B,ABS,Y				LDA B,ABS,X		LDA B,ABL,X
1100	C		CMP B,(DIR,X)		CMP B,SR		CMP B,DIR		CMP B,L(DIR)		CMP B,IMM				CMP B,ABS		CMP B,ABL
1101	D		CMP B,(DIR),Y	CMP B,(DIR)	CMP B,(SR),Y		CMP B,DIR,X		CMP B,L(DIR),Y		CMP B,ABS,Y				CMP B,ABS,X		CMP B,ABL,X
1110	E		SBC B,(DIR,X)		SBC B,SR		SBC B,DIR		SBC B,L(DIR)		SBC B,IMM				SBC B,ABS		SBC B,ABL
1111	F		SBC B,(DIR),Y	SBC B,(DIR)	SBC B,(SR),Y		SBC B,DIR,X		SBC B,L(DIR),Y		SBC B,ABS,Y				SBC B,ABS,X		SBC B,ABL,X

APPENDIX 7.INSTRUCTION CODE TABLE

INSTRUCTION CODE TABLE–3 (The first word's code of each instruction is 89_{16})

D_7~D_4 \ D_3~D_0	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		MPY (DIR,X)		MPY SR		MPY DIR		MPY L(DIR)		MPY IMM				MPY ABS		MPY ABL
0001	1		MPY (DIR),Y	MPY (DIR)	MPY (SR),Y		MPY DIR,X		MPY L(DIR),Y		MPY ABS,Y				MPY ABS,X		MPY ABL,X
0010	2		DIV (DIR,X)		DIV SR		DIV DIR		DIV L(DIR)	XAB	DIV IMM				DIV ABS		DIV ABL
0011	3		DIV (DIR),Y	DIV (DIR)	DIV (SR),Y		DIV DIR,X		DIV L(DIR),Y		DIV ABS,Y				DIV ABS,X		DIV ABL,X
0100	4										RLA IMM						
0101	5																
0110	6																
0111	7																
1000	8																
1001	9																
1010	A																
1011	B																
1100	C			LDT IMM													
1101	D																
1110	E																
1111	F																

APPENDIX 8
MACHINE INSTRUCTIONS

APPENDIX 8.MACHINE INSTRUCTIONS

SYMBOLS

The following notations are used for the following descriptions.

Symbol	Description	Symbol	Description
IMP	Implied addressing mode	$\forall$	Exclusive OR
IMM	Immediate addressing mode	—	Negation
A	Accumulator addressing mode	←	Movement to the arrow direction
DIR	Direct addressing mode	A_{CC}	Accumulator
DIR, b	Direct bit addressing mode	A_{CCH}	Accumulator's upper 8 bits
DIR, X	Direct indexed X addressing mode	A_{CCL}	Accumulator's lower 8 bits
DIR, Y	Direct indexed Y addressing mode	A	Accumulator A
(DIR)	Direct indirect addressing mode	A_H	Accumulator A's upper 8 bits
(DIR, X)	Direct indexed X indirect addressing mode	A_L	Accumulator A's lower 8 bits
(DIR), Y	Direct indirect indexed Y addressing mode	B	Accumulator B
L (DIR)	Direct indirect long addressing mode	B_H	Accumulator B's upper 8 bits
L (DIR), Y	Direct indirect long indexed Y addressing mode	B_L	Accumulator B's lower 8 bits
ABS	Absolute addressing mode	X	Index register X
ABS, b	Absolute bit addressing mode	X_H	Index register X's upper 8 bits
ABS, X	Absolute indexed X addressing mode	X_L	Index register X's lower 8 bits
ABS, Y	Absolute indexed Y addressing mode	Y	Index register Y
ABL	Absolute long addressing mode	Y_H	Index register Y's upper 8 bits
ABL, X	Absolute long indexed X addressing mode	Y_L	Index register Y's lower 8 bits
(ABS)	Absolute indirect addressing mode	S	Stack pointer
L (ABS)	Absolute indirect long addressing mode	PC	Program counter
(ABS, X)	Absolute indexed X indirect addressing mode	PC_H	Program counter's upper 8 bits
STK	Stack addressing mode	PC_L	Program counter's lower 8 bits
REL	Relative addressing mode	PG	Program bank register
DIR, b, REL	Direct bit relative addressing mode	DT	Data bank register
ABS, b, REL	Absolute bit relative addressing mode	DPR	Direct page register
SR	Stack pointer relative addressing mode	DPR_H	Direct page register's upper 8 bits
(SR), Y	Stack pointer relative indirect indexed Y addressing mode	DPR_L	Direct page register's lower 8 bits
		PS	Processor status register
BLK	Block transfer addressing mode	PS_H	Processor status register's upper 8 bits
C	Carry flag	PS_L	Processor status register's lower 8 bits
Z	Zero flag	PS_b	Processor status register's b-th bit
I	Interrupt disable flag	M(S)	Contents of memory at address indicated by stack pointer
D	Decimal operation mode flag		
x	Index register length selection flag	M_b	b-th memory location
m	Data length selection flag	AD_G	Value of 24-bit address's upper 8-bit (A_{23}~A_{16})
V	Overflow flag	AD_H	Value of 24-bit address's middle 8-bit (A_{15}~A_8)
N	Negative flag	AD_L	Value of 24-bit address's lower 8-bit (A_7~A_0)
IPL	Processor interrupt priority level	op	Operation code
+	Addition	n	Number of cycle
−	Subtraction	#	Number of byte
*	Multiplication	i	Number of transfer byte or rotation
/	Division	i_1, i_2	Number of registers pushed or pulled
$\wedge$	Logical AND		
$\vee$	Logical OR		

NOTES

The number of cycle shown in the table is described in case of the fastest mode for each instruction. The number of cycles shown in the table is calculated for DPR_L=0. The number of cycles in the addressing mode concerning the DPR when $DPR_L \neq 0$ must be incremented by 1.

The number of cycles shown in the table differs according to the bytes fetched into the instruction queue buffer, or according to whether the memory read/write address is odd or even. It also differs when the external region memory is accessed by BYTE="H".

Note 1.The operation code at the upper row is used for accumulator A, and the operation at the lower row is used for accumulator B.

Note 2.When setting flag m=0 to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 3.The number of cycles increments by 2 when branching.

Note 4.The operation code on the upper row is used for branching in the range of –128~+127, and the operation code on the lower row is used for branching in the range of –32768~+32767.

Note 5.When handling 16-bit data with flag m=0, the byte in the table is incremented by 1.

Note 6.The number of cycles corresponding to the register to be pushed. The number of cycles when no pushing is done is 12.

Type of register	A	B	X	Y	DPR	DT	PG	PS
Number of cycles	2	2	2	2	2	1	1	1

i_1 indicates the number of registers among A, B, X, Y, DT, and PS to be saved, when i_2 indicates the number of registers among DT and PG to be saved.

Note 7.The number of cycles corresponding to the register to be pulled are added. The number of cycles when no pulling is done is 14.

Type of register	A	B	X	Y	DPR	DT	PS
Number of cycles	3	3	3	3	4	3	3

i_1 indicates the number of registers among A, B, X, Y, DT, and PS to be restored, while i_2=1 when DPR is to be restored.

Note 8.The number of cycles is the case when the number of bytes to be transferred is even. When the number of bytes to be transferred is odd, the number is calculated as; 7 + (i/2) X 7 + 4, note that, (i/2) shows the integer part when i is divided by 2.

Note 9.The number of cycles is the case when the number of bytes to be transferred is even. When the number of bytes to be transferred is odd, the number is calculated as; 9 + (i/2) X 7 + 5, note that, (i/2) shows the integer part when i is divided by 2.

Note 10.The number of cycles is the case in the 16-bit ÷ 8-bit operation. The number of cycles is incremented by 16 for 32-bit ÷ 16-bit operation.

Note 11.The number of cycles is the case in the 8-bit X 8-bit operation. The number of cycles is incremented by 8 for 16-bit X 16-bit operation.

Note 12.When setting flag x=0 to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 13.When flag m is 0, the byte in the table is incremented by 1.

APPENDIX 8.MACHINE INSTRUCTIONS

MACHINE INSTRUCTION

Symbol	Function	Details	Addressing mode																														
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
ADC (Note 1,2)	A_{CC},C ← A_{CC}+M+C	Adds the carry, the accumulator and the memory contents. The result is entered into the accumulator. When the D flag is "0", binary additions is done, and when the D flag is "1", decimal addition is done.				69	2	2				65	4	2				75	5	2				72	6	2	61	7	2	71	8	2
						42 69	4	3				42 65	6	3				42 75	7	3				42 72	8	3	42 61	9	3	42 71	10	3
AND (Note 1,2)	A_{CC} ← A_{CC}∧M	Obtains the logical product of the contents of the accumulator and the contents of the memory. The result is entered into the accumulator.				29	2	2				25	4	2				35	5	2				32	6	2	21	7	2	31	8	2
						42 29	4	3				42 25	6	3				42 35	7	3				42 32	8	3	42 21	9	3	42 31	10	3
ASL (Note 1)	m=0 C ← b_{15} ··· b_0 ← 0 m=1 C ← b_7 ··· b_0 ← 0	Shifts the accumulator or the memory contents one bit to the left. "0" is entered into bit 0 of the accumulator or the memory. The contents of bit 15 (bit 7 when the m flag is "1") of the accumulator or memory before shift is entered into the C flag.							0A	2	1	06	7	2				16	7	2												
									42 0A	4	2																					
BBC (Note 3,5)	Mb=0 ?	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "0".																															
BBS (Note 3,5)	Mb=1 ?	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "1".																															
BCC (Note 3)	C=0 ?	Branches when the contents of the C flag is "0".																															
BCS (Note 3)	C=1 ?	Branches when the contents of the C flag is "1".																															
BEQ (Note 3)	Z=1 ?	Branches when the contents of the Z flag is "1".																															
BMI (Note 3)	N=1 ?	Branches when the contents of the N flag is "1".																															
BNE (Note 3)	Z=0 ?	Branches when the contents of the Z flag is "0".																															
BPL (Note 3)	N=0 ?	Branches when the contents of the N flag is "0".																															
BRA (Note 4)	PC←PC±offset PG←PG+1 (carry occured) PG←PG−1 (borrow occured)	Jumps to the address indicated by the program counter plus the offset value.																															
BRK	PC←PC+2 M(S)←PG S←S−1 M(S)←PC_H S←S−1 M(S)←PC_L S←S−1 M(S)←PS_H S←S−1 M(S)←PS_L S←S−1 I←1 PC_L←AD_L PC_H←AD_H PG←00_{16}	Executes software interruption.	00	15	2																												
BVC (Note 3)	V=0 ?	Branches when the contents of the V flag is "0".																															
BVS (Note 3)	V=1 ?	Branches when the contents of the V flag is "1".																															
CLB (Note 5)	Mb←0	Makes the contents of the specified bit in the memory "0".													14	8	3																
CLC	C←0	Makes the contents of the C flag "0".	18	2	1																												
CLI	I←0	Makes the contents of the I flag "0".	58	2	1																												
CLM	m←0	Makes the contents of the m flag "0".	D8	2	1																												
CLP	PSb←0	Specifies the bit position in the processor status register by the bit pattern of the second byte in the instruction, and sets "0" in that bit.				C2	4	2																									
CLV	V←0	Makes the contents of the V flag "0".	B8	2	1																												
CMP (Note 1,2)	A_{CC}−M	Compares the contents of the accumulator with the contents of the memory.				C9	2	2				C5	4	2				D5	5	2				D2	6	2	C1	7	2	D1	8	2
						42 C9	4	3				42 C5	6	3				42 D5	7	3				42 D2	8	3	42 C1	9	3	42 D1	10	3

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
67	10	2	77	11	2	6D	4	3				7D	6	3	79	6	3	6F	6	4	7F	7	4																						63	5	2	73	8	2				•	•	•	N	V	•	•	•	•	Z	C
42 67	12	3	42 77	13	3	42 6D	6	4				42 7D	8	4	42 79	8	4	42 6F	8	5	42 7F	9	5																						42 63	7	3	42 73	10	3														
27	10	2	37	11	2	2D	4	3				3D	6	3	39	6	3	2F	6	4	3F	7	4																						23	5	2	33	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 27	12	3	42 37	13	3	42 2D	6	4				42 3D	8	4	42 39	8	4	42 2F	8	5	42 3F	9	5																						42 23	7	3	42 33	10	3														
						0E	7	3				1E	8	3																																								•	•	•	N	•	•	•	•	•	Z	C
																																							34	7	4	3C	8	5										•	•	•	•	•	•	•	•	•	•	•
																																							24	7	4	2C	8	5										•	•	•	•	•	•	•	•	•	•	•
																																				90	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				B0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				F0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				30	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				D0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				10	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				80	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				82	4	3																										
																																																						•	•	•	•	•	•	•	•	•	•	•
																																				50	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				70	4	2																•	•	•	•	•	•	•	•	•	•	•
									1C	9	4																																											•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	0
																																																						•	•	•	•	•	•	•	•	0	•	•
																																																						•	•	•	•	•	•	0	•	•	•	•
																																																						•	•	•	Specified flag becomes "0".							
																																																						•	•	•	•	•	0	•	•	•	•	•
C7	10	2	D7	11	2	CD	4	3				DD	6	3	D9	6	3	CF	6	4	DF	7	4																						C3	5	2	D3	8	2				•	•	•	N	•	•	•	•	•	Z	C
42 C7	12	3	42 D7	13	3	42 CD	6	4				42 DD	8	4	42 D9	8	4	42 CF	8	5	42 DF	9	5																						42 C3	7	3	42 D3	10	3														

APPENDIX 8.MACHINE INSTRUCTIONS

Symbol	Function	Details	Addressing mode																																		
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y							
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
CPX (Note 2)	X−M	Compares the contents of the index register X with the contents of the memory.				E0	2	2				E4	4	2																								
CPY (Note 2)	Y−M	Compares the contents of the index register Y with the contents of the memory.				C0	2	2				C4	4	2																								
DEC (Note 1)	A_{CC}←A_{CC}−1 or M←M−1	Decrements the contents of the accumulator or memory by 1.							1A	2	1	C6	7	2				D6	7	2																		
									42 1A	4	2																													
DEX	X←X−1	Decrements the contents of the index register X by 1.	CA	2	1																																	
DEY	Y←Y−1	Decrements the contents of the index register Y by 1.	88	2	1																																	
DIV (Note 2,10)	A(quotient)←B,A/M B(remainder)	The numeral that places the contents of accumulator B to the higher order and the contents of accumulator A to the lower order is divided by the contents of the memory. The quotient is entered into accumulator A and the remainder into accumulator B.				89 29	27	3				89 25	29	3				89 35	30	3				89 32	31	3	89 21	32	3	89 31	33	3						
EOR (Note 1,2)	A_{CC}←A_{CC}∀M	Logical exclusive sum is obtained of the contents of the accumulator and the contents of the memory. The result is placed into the accumulator.				49	2	2				45	4	2				55	5	2				52	6	2	41	7	2	51	8	2						
						42 49	4	3				42 45	6	3				42 55	7	3				42 52	8	3	42 41	9	3	42 51	10	3						
INC (Note 1)	A_{CC}←A_{CC}+1 or M ←M+1	Increments the contents of the accumulator or memory by 1.							3A	2	1	E6	7	2				F6	7	2																		
									42 3A	4	2																													
INX	X←X+1	Increments the contents of the index register X by 1.	E8	2	1																																	
INY	Y←Y+1	Increments the contents of the index register Y by 1.	C8	2	1																																	
JMP	ABS PC_L ← AD_L PC_H ← AD_H ABL PC_L ← AD_L PC_H ← AD_H PG←AD_G (ABS) PC_L ←(AD_H, AD_L) PC_H ←(AD_H, AD_L+1) L(ABS) PC_L ←(AD_H, AD_L) PC_H ←(AD_H, AD_L+1) PG ←(AD_H, AD_L+2) (ABS, X) PC_L ←(AD_H, AD_L+X) PC_H ←(AD_H, AD_L+X +1)	Places a new address into the program counter and jumps to that new address.																																				
JSR	ABS M(S)← PC_H S←S−1 M(S)← PC_L S←S−1 PC_L ← AD_L PC_H ← AD_H ABL M(S)← PG S←S−1 M(S)← PC_H S←S−1 M(S)←PC_L S ← S−1 PC_L ← AD_L PC_H ← AD_H PG ← AD_G (ABS, X) M(S)← PC_H S←S−1 M(S)← PC_L S←S−1 PC_L ←(AD_H, AD_L+X) PC_H ←(AD_H, AD_L+X +1)	Saves the contents of the program counter (also the contents of the program bank register for ABL) into the stack, and jumps to the new address.																																				

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
						EC	4	3																																														•	•	•	N	•	•	•	•	•	Z	C
						CC	4	3																																														•	•	•	N	•	•	•	•	•	Z	C
						CE	7	3				DE	8	3																																								•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
89 27	35	3	89 37	36	3	89 2D	29	4				89 3D	31	4	89 39	31	4	89 2F	31	5	89 3F	32	5																							89 23	30	3	89 33	33	3				•	•	•	N	V	•	•	•	•	Z	C
47	10	2	57	11	2	4D	4	3				5D	6	3	59	6	3	4F	6	4	5F	7	4																							43	5	2	53	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 47	12	3	42 57	13	3	42 4D	6	4				42 5D	8	4	42 59	8	4	42 4F	8	5	42 5F	9	5																							42 43	7	3	42 53	10	3														
						EE	7	3				FE	8	3																																								•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
						4C	2	3										5C	4	4				6C	4	3	DC	8	3	7C	6	3																						•	•	•	•	•	•	•	•	•	•	•
						20	6	3										22	8	4										FC	8	3																						•	•	•	•	•	•	•	•	•	•	•

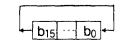
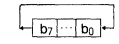
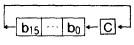
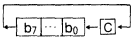
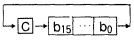
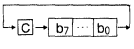
APPENDIX 8.MACHINE INSTRUCTIONS

Symbol	Function	Details	IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
LDA (Note 1,2)	$A_{CC} \leftarrow M$	Enters the contents of the memory into the accumulator.				A9	2	2				A5	4	2				B5	5	2				B2	6	2	A1	7	2	B1	8	2
						42 A9	4	3				42 A5	6	3				42 B5	7	3				42 B2	8	3	42 A1	9	3	42 B1	10	3
LDM (Note 5)	M ← IMM	Enters the immediate value into the memory.										64	4	3				74	5	3												
LDT	DT ← IMM	Enters the immediate value into the data bank register.				89 C2	5	3																								
LDX (Note 2)	X ← M	Enters the contents of the memory into index register X.				A2	2	2				A6	4	2							B6	5	2									
LDY (Note 2)	Y ← M	Enters the contents of the memory into index register Y.				A0	2	2				A4	4	2				B4	5	2												
LSR (Note 1)	m=0 0→[b_{15} ··· b_0]→C m=1 0→[b_7 ··· b_0]→C	Shifts the contents of the accumulator or the contents of the memory one bit to the right. The bit 0 of the accumulator or the memory is entered into the C flag. "0" is entered into bit 15 (bit 7 when the m flag is "1".)							4A	2	1	46	7	2				56	7	2												
									42 4A	4	2																					
MPY (Note 2,11)	B, A←A∗M	Multiplies the contents of accumulator A and the contents of the memory. The higher order of the result of operation are entered into accumulator B, and the lower order into accumulator A.				89 09	16	3				89 05	18	3				89 15	19	3				89 12	20	3	89 01	21	3	89 11	22	3
MVN (Note 8)	Mn+i←Mm+i	Transmits the data block. The transmission is done from the lower order address of the block.																														
MVP (Note 9)	Mn−i←Mm−i	Transmits the data block. Transmission is done form the higher order address of the data block.																														
NOP	PC←PC+1	Advances the program counter, but performs nothing else.	EA	2	1																											
ORA (Note 1,2)	$A_{CC} \leftarrow A_{CC} \vee M$	Logical sum per bit of the contents of the accumulator and the contents of the memory is obtained. The result is entered into the accumulator.				09	2	2				05	4	2				15	5	2				12	6	2	01	7	2	11	8	2
						42 09	4	3				42 05	6	3				42 15	7	3				42 12	8	3	42 01	9	3	42 11	10	3
PEA	M(S)←IMM_2 S←S−1 M(S)←IMM_1 S←S−1	The 3rd and the 2nd bytes of the instruction are saved into the stack, in this order.																														
PEI	M(S)←M((DPR)+IMM +1) S←S−1 M(S)←M((DPR)+IMM) S←S−1	Specifies 2 sequential bytes in the direct page in the 2nd byte of the instruction, and saves the contents into the stack.																														
PER	EAR←PC+IMM_2,IMM_1 M(S)←EAR_H S←S−1 M(S)←EAR_L S←S−1	Regards the 2nd and 3rd bytes of the instruction as 16-bit numerals, adds them to the program counter, and saves the result into the stack.																														
PHA	m=0 M(S)←A_H S←S−1 M(S)←A_L S←S−1 m=1 M(S)←A_L S←S−1	Saves the contents of accumulator A into the stack.																														
PHB	m=0 M(S)←B_H S←S−1 M(S)←B_L S←S−1 m=1 M(S)←B_L S←S−1	Saves the contents of accumulator B into the stack.																														

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
A7	10	2	B7	11	2	AD	4	3				BD	6	3	B9	6	3	AF	6	4	BF	7	4																						A3	5	2	B3	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 A7	12	3	42 B7	13	3	42 AD	6	4				42 BD	8	4	42 B9	8	4	42 AF	8	5	42 BF	9	5																						42 A3	7	3	42 B3	10	3														
						9C	5	4				9E	6	4																																								•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	•
						AE	4	3							BE	6	3																																					•	•	•	N	•	•	•	•	•	Z	•
						AC	4	3				BC	6	3																																								•	•	•	N	•	•	•	•	•	Z	•
						4E	7	3				5E	8	3																																								•	•	•	0	•	•	•	•	•	Z	C
89 07	24	3	89 17	25	3	89 0D	18	4				89 1D	20	4	89 19	20	4	89 0F	20	5	89 1F	21	5																						89 03	19	3	89 13	22	3				•	•	•	N	•	•	•	•	•	Z	0
																																																		54	$7 + \frac{i}{2} \times 7$	3	•	•	•	•	•	•	•	•	•	•	•
																																																			44	$9 + \frac{i}{2} \times 7$	3	•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	•
07	10	2	17	11	2	0D	4	3				1D	6	3	19	6	3	0F	6	4	1F	7	4																						03	5	2	13	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 07	12	3	42 17	13	3	42 0D	6	4				42 1D	8	4	42 19	8	4	42 0F	8	5	42 1F	9	5																						42 03	7	3	42 13	10	3														
																																	F4	5	3																					•	•	•	•	•	•	•	•	•	•	•
																																	D4	5	2																					•	•	•	•	•	•	•	•	•	•	•
																																	62	5	3																					•	•	•	•	•	•	•	•	•	•	•
																																	48	4	1																					•	•	•	•	•	•	•	•	•	•	•
																																	42 48	6	2																					•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																																		
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y							
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#					
PHD	M(S)←DPR_H S←S−1 M(S)←DPR_L S←S−1	Saves the contents of the direct page register into the stack.																																			
PHG	M(S)←PG S←S−1	Saves the contents of the program bank register into the stack.																																			
PHP	M(S)←PS_H S←S−1 M(S)←PS_L S←S−1	Saves the contents of the program status register into the stack.																																			
PHT	M(S)←DT S←S−1	Saves the contents of the data bank register into the stack.																																			
PHX	x=0 M(S)←X_H S←S−1 M(S)←X_L S←S−1 x=1 M(S)←X_L S←S−1	Saves the contents of the index register X into the stack.																																			
PHY	x=0 M(S)←Y_H S←S−1 M(S)←Y_L S←S−1 x=1 M(S)←Y_L S←S−1	Saves the contents of the index register Y into the stack.																																			
PLA	m=0 S←S+1 A_L←M(S) S←S+1 A_H←M(S) m=1 S←S+1 A_L←M(S)	Restores the contents of the stack on the accumulator A.																																			
PLB	m=0 S←S+1 B_L←M(S) S←S+1 B_H←M(S) m=1 S←S+1 B_L←M(S)	Restores the contents of the stack on the accumulator B.																																			
PLD	S←S+1 DPR_L←M(S) S←S+1 DPR_H←M(S)	Restores the contents of the stack on the direct page register.																																			
PLP	S←S+1 PS_L←M(S) S←S+1 PS_H←M(S)	Restores the contents of the stack on the processor status register.																																			
PLT	S←S+1 DT←M(S)	Restores the contents of the stack on the data bank register.																																			
PLX	x=0 S←S+1 X_L←M(S) S←S+1 X_H←M(S) x=1 S←S+1 X_L←M(S)	Restores the contents of the stack on the index register X.																																			

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																	0B	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	4B	3	1																			•	•	•	•	•	•	•	•	•	•	•
																																	08	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	8B	3	1																			•	•	•	•	•	•	•	•	•	•	•
																																	DA	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	5A	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	68	5	1																			•	•	•	N	•	•	•	•	•	Z	•
																																	42 68	7	2																			•	•	•	N	•	•	•	•	•	Z	•
																																	2B	5	1																			•	•	•	•	•	•	•	•	•	•	•
																																	28	6	1																			Value saved in stack.										
																																	AB	6	1																			•	•	•	N	•	•	•	•	•	Z	•
																																	FA	5	1																			•	•	•	N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																																			
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y								
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#						
PLY	x=0 S←S+1 Y_L←M(S) S←S+1 Y_H←M(S) x=1 S←S+1 Y_L←M(S)	Restores the contents of the stack on the index register Y.																																				
PSH (Note 6)	M(S)←A, B, X···	Saves the registers among accumulator, index register, direct page register, data bank register, program bank register, or processor status register, specified by the bit pattern of the second byte of the instruction into the stack.																																				
PUL (Note 7)	A, B, X···←M(S)	Restores the contents of the stack to the registers among accumulator, index register, direct page register, data bank register, or processor status register, specified by the bit pattern of the second byte of the instruction.																																				
RLA (Note 13)	m=0 n bit rotate left  m=1 n bit rotate left 	Rotates the contents of the accumulator A, n bits to the left.				89 49	6 + i	3																																
ROL (Note 1)	m=0  m=1 	Links the accumulator or the memory to C flag, and rotates result to the left by 1 bit.							2A 42 2A	2 4	1 2	26	7	2				36	7	2																						
ROR (Note 1)	m=0  m=1 	Links the accumulator or the memory to C flag, and rotates result to the right by 1 bit.							6A 42 6A	2 4	1 2	66	7	2				76	7	2																						
RTI	S←S+1 PS_L←M(S) S←S+1 PS_H←M(S) S←S+1 PC_L←M(S) S←S+1 PC_H←M(S) S←S+1 PG←M(S)	Returns from the interruption routine.	40	11	1																																		
RTL	S←S+1 PC_L←M(S) S←S+1 PC_H←M(S) S←S+1 PG←M(S)	Returns from the subroutine. The contents of the program bank register are also restored.	6B	8	1																																		
RTS	S←S+1 PC_L←M(S) S←S+1 PC_H←M(S)	Returns from the subroutine. The contents of the program bank register are not restored.	60	5	1																																		
SBC (Note 1,2)	A_{CC}, C←A_{CC}−M−$\overline{C}$	Subtracts the contents of the memory and the borrow from the contents of the accumulator.				E9 42 E9	2 4	2 3				E5 42 E5	4 6	2 3				F5 42 F5	5 7	2 3				F2 42 F2	6 8	2 3	E1 42 E1	7 9	2 3	F1 42 F1	8 10	2 3						

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																	7A	5	1																			•	•	•	N	•	•	•	•	•	Z	•
																																	EB	$12+2i_1+i_2$	2																			•	•	•	•	•	•	•	•	•	•	•
																																	FB	$14+3i_1+4i_2$	2																			If restored the contents of PS, it becomes its value. And the other case is no change.										
																																																				•	•	•	•	•	•	•	•	•	•	•
						2E	7	3				3E	8	3																																								•	•	•	N	•	•	•	•	•	Z	C
						6E	7	3				7E	8	3																																								•	•	•	N	•	•	•	•	•	Z	C
																																																				Value saved in stack.										
																																																				•	•	•	•	•	•	•	•	•	•	•
																																																				•	•	•	•	•	•	•	•	•	•	•
E7	10	2	F7	11	2	ED	4	3				FD	6	3	F9	6	3	EF	6	4	FF	7	4																						E3	5	2	F3	8	2				•	•	•	N	V	•	•	•	•	Z	C
42 E7	12	3	42 F7	13	3	42 ED	6	4				42 FD	8	4	42 F9	8	4	42 EF	8	5	42 FF	9	5																						42 E3	7	3	42 F3	10	3														

APPENDIX 8.MACHINE INSTRUCTIONS

Symbol	Function	Details	Addressing mode																															
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
SEB (Note 5)	Mb←1	Makes the contents of the specified bit in the memory "1".													04	8	3																	
SEC	C←1	Makes the contents of the C flag "1".	38	2	1																													
SEI	I←1	Makes the contents of the I flag "1".	78	2	1																													
SEM	m←1	Makes the contents of the m flag "1".	F8	2	1																													
SEP	PSb←1	Set the specified bit of the processor status register's lower byte (PS_L) to "1".				E2	3	2																										
STA (Note 1)	M←A_{CC}	Stores the contents of the accumulator into the memory.										85	4	2				95	5	2				92	7	2	81	7	2	91	7	2
												42 85	6	3				42 95	7	3				42 92	9	3	42 81	9	3	42 91	9	3
STP		Stops the oscillation of the oscillator.	DB	3	1																													
STX	M←X	Stores the contents of the index register X into the memory.										86	4	2							96	5	2											
STY	M←Y	Stores the contents of the index register Y into the memory.										84	4	2				94	5	2														
TAD	DPR←A	Transmits the contents of the accumulator A to the direct page register.	5B	2	1																													
TAS	S←A	Transmits the contents of the accumulator A to the stack pointer.	1B	2	1																													
TAX	X←A	Transmits the contents of the accumulator A to the index register X.	AA	2	1																													
TAY	Y←A	Transmits the contents of the accumulator A to the index register Y.	A8	2	1																													
TBD	DPR←B	Transmits the contents of the accumulator B to the direct page register.	42 5B	4	2																													
TBS	S←B	Transmits the contents of the accumulator B to the stack pointer.	42 1B	4	2																													
TBX	X←B	Transmits the contents of the accumulator B to the index register X.	42 AA	4	2																													
TBY	Y←B	Transmits the contents of the accumulator B to the index register Y.	42 A8	4	2																													
TDA	A←DPR	Transmits the contents of the direct page register to the accumulator A.	7B	2	1																													
TDB	B←DPR	Transmits the contents of the direct page register to the accumulator B.	42 7B	4	2																													
TSA	A←S	Transmits the contents of the stack pointer to the accumulator A.	3B	2	1																													
TSB	B←S	Transmits the contents of the stack pointer to the accumulator B.	42 3B	4	2																													
TSX	X←S	Transmits the contents of the stack pointer to the index register X.	BA	2	1																													
TXA	A←X	Transmits the contents of the index register X to the accumulator A.	8A	2	1																													
TXB	B←X	Transmits the contents of the index register X to the accumulator B.	42 8A	4	2																													
TXS	S←X	Transmits the contents of the index register X to the stack pointer.	9A	2	1																													
TXY	Y←X	Transmits the contents of the index register X to the index register Y.	9B	2	1																													
TYA	A←Y	Transmits the contents of the index register Y to the accumulator A.	98	2	1																													
TYB	B←Y	Transmits the contents of the index register Y to the accumulator B.	42 98	4	2																													
TYX	X←Y	Transmits the contents of the index register Y to the index register X.	BB	2	1																													
WIT		Stops the internal clock.	CB	3	1																													
XAB	A⇄B	Exchanges the contents of the accumulator A and the contents of the accumulator B.	89 28	6	2																													

Addressing mode																																																									Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
									0C	9	4																																															•	•	•	•	•	•	•	•	•	•	•
																																																								•	•	•	•	•	•	•	•	•	•	1
																																																								•	•	•	•	•	•	•	•	1	•	•
																																																								•	•	•	•	•	1	•	•	•	•	•
																																																								•	•	•	Specified flag becomes "1".							
87	10	2	97	11	2	8D	5	3				9D	5	3	99	5	3	8F	6	4	9F	7	4																						83	5	2	93	8	2				•	•	•	•	•	•	•	•	•	•	•
42 87	12	3	42 97	13	3	42 8D	7	4				42 9D	7	4	42 99	7	4	42 8F	8	5	42 9F	9	5																						42 83	7	3	42 93	10	3														
																																																								•	•	•	•	•	•	•	•	•	•	•
						8E	5	3																																																	•	•	•	•	•	•	•	•	•	•	•
						8C	5	3																																																	•	•	•	•	•	•	•	•	•	•	•
																																																									•	•	•	•	•	•	•	•	•	•	•
																																																									•	•	•	•	•	•	•	•	•	•	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	•	•	•	•	•	•	•	•
																																																									•	•	•	•	•	•	•	•	•	•	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	•	•	•	•	•	•	•	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	N	•	•	•	•	•	Z	•
																																																									•	•	•	•	•	•	•	•	•	•	•
																																																									•	•	•	N	•	•	•	•	•	Z	•

MEMO

APPENDIX 9
MASK ROM ORDERING METHOD

APPENDIX 9.MASK ROM ORDERING METHOD

Mask ROM Ordering Method

Mitsubishi Electric Corporation accepts order to transfer EPROM supplied program data into the mask ROMs in single-chip 16-bit microcomputers. When placing such order, please submit the information description below.

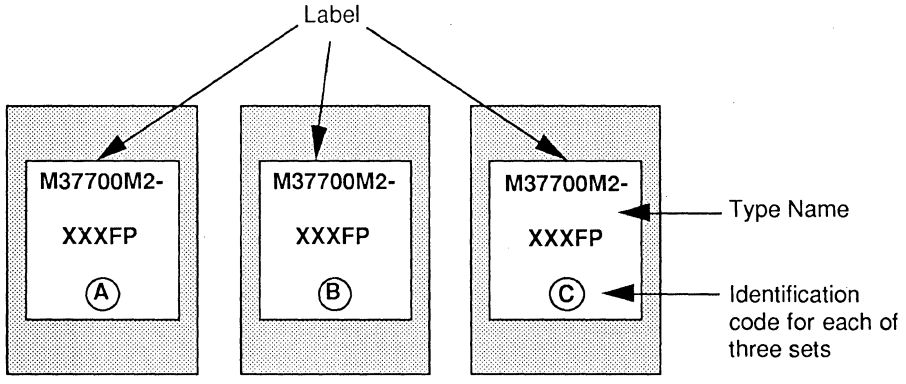
1.Mask ROM Order Confirmation Form ..... 1 set
(There is a specific form to be used for each model.)
2.Data to be written into mask ROM ..... EPROM
(Please provide three sets containing the identical data.)
3.Mark Specification Form ..... 1 set

Please use the Mask ROM Order Confirmation Form and Mark Specification Form on the latest data book.

Notes

(1)Acceptable EPROM type
Any EPROM made by Mitsubishi Electric Corporation that is listed in the Mask ROM Order Confirmation Form may be used.
(2)EPROM window labeling
Please write the type name and the identification code (A, B, C) on the label for each of the three sets of data EPROMs provided.

Example :



(3)Calculation and indication of check sum code
Please calculate the total number of data in words in the EPROM, and write the number in 4-digit hexadecimal form in the check sum code field of the Mask ROM Order Confirmation Form.
(4)Options
Refer to the appropriate data book entry and write the desired options on the Mask ROM Order Confirmation Form.
(5)Marking specification method
The permissible marking specifications differ depending on the shape of package. Please fill out the Marking Specification Form and attach it to the Mask ROM Order Confirmation Form.

Mark Specification Form

Mark specification format differs depending on the package type.
Fill out the Mark Specification Form for the package type being ordered, and submit the form with the Mask ROM Order Confirmation Form.

APPENDIX 10
Q & A

1.Parallel ports

(1) Port P4

Q :What is read from port $P4_0$ ($\overline{HOLD}$) and $P4_1$ ($\overline{RDY}$) when port P4 is read in memory expansion mode or microprocessor mode?

A :The $P4_0$ and $P4_1$ level is obtained.

2.Interrupt

(1) Interrupt priority level

Q :Which type of interrupts has higher priority; software interrupts or other interrupts?

A :When a **BRK** instruction is executed or a zero divide is performed in an interrupt handling routine, a **BRK** interrupt or a zero divide interrupt occurs and that interrupt is serviced. However, if multiple interrupts are enabled by setting the I flag to "0", interrupts with priority higher than IPL are accepted because the IPL is not changed. Furthermore, the WDT interrupt is always enabled.

(2) Processor mode register

Q :What are bits 4 and 5 (interrupt priority level detection interval selection bits) in the processor mode register ($5E_{16}$) used for?

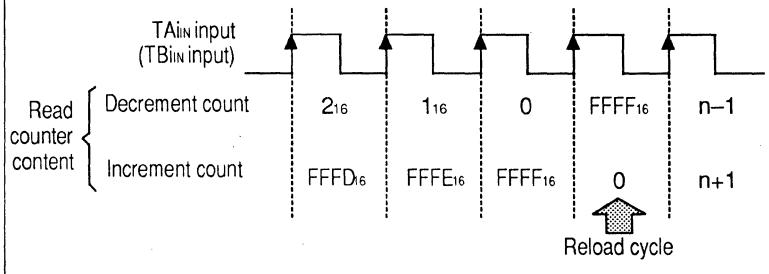
A :These bits are reserved for future functional enhancements when the processor speed or the number of interrupt sources is increased. When the oscillating frequency is 8MHz or 16MHz and the number of interrupt sources is 19, the shortest interval (bits 5,4=1,0) can be used.

3.Timer/Counter

(1) Reading the timer register

Q :Is the result of reading the timer register in timer A or timer B timer mode/event counter mode not the same as the value set in the reload register?

A :The following figure shows an example where the rising edge is selected for event counter mode. (Falling edge and timer mode is also the same.)



In other words, for decrement count, $FFFF_{16}$ is returned at the reload cycle and n–1 (n=reload value) is returned at the next cycle. For increment count, 0 is returned at the reload cycle and n+1 is returned at the next cycle. Therefore, the value set in the timer register (n) cannot be read while the timer is active. However, the value n is returned if no count source is provided after writing a value in the timer register or if the count start flag is set to disable.

On the other hand, in timer A one-shot pulse generation mode and PWM mode, the value set in the reload register can always be read correctly.

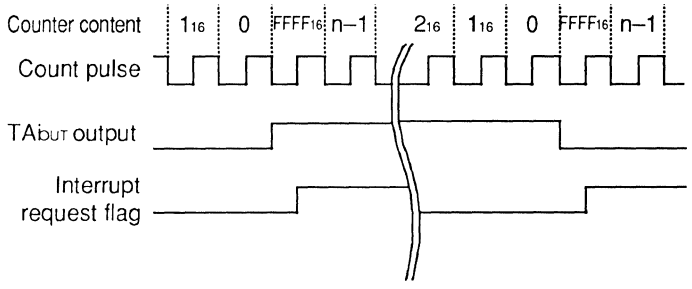
(2) Timer interrupt request

Q : When is the interrupt request flag set? How is it related to the output pulse change timing?

A : • Timer mode and event counter mode

The TA_{iOUT} output pulse changes direction at the effective edge of the count pulse after the counter has reached "0000_{16}" ('$FFFF_{16}$" if increment count is selected in event counter mode).
The interrupt request flag is set when the effective edge changes direction (same for timer B).

Example) Timer A timer mode and event counter mode (falling edge effective, decrement count)



• One-shot pulse mode and PWM mode

The interrupt request flag is set at the falling edge of the TA_{iOUT} output.

(3) Timer I/O pins

Q : Can the pin levels of pins TA_{iIN}, TA_{iOUT}, and TB_{iIN} be read when these pins are used as timer I/O pins?

A : If the direction register is set to "input", the pin level can be read. If the direction register is set to "output" the port latch data is returned instead of the pin level.

(4) Timer A

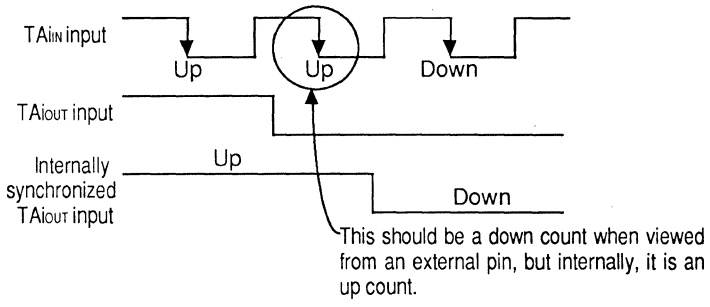
Q : Is it possible to use timer A as an 8-bit timer+8-bit prescaler in modes other than pulse width modulation mode?

A : Timers operate basically in 16-bit mode. However, when performing pulse width modulation, the output pulse frequency can only be selected with the clock source (four types) if the timer is 16 bits. Therefore, in 8-bit pulse width modulation mode, the timer is divided into 8 bit halves with the low-order 8 bits used to select the pulse output frequency and the high-order 8 bits used to set the "H" width of the pulse output. In other modes, only 16-bit timers are available because they are more precise than 8-bit prescaler+8-bit timer.

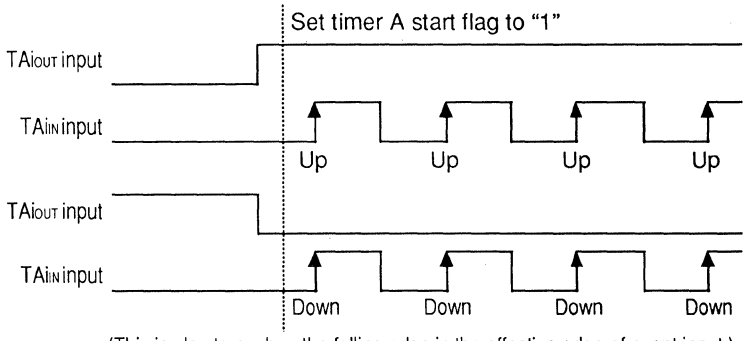
(5) Event counter mode

Q : What is the increment/decrement switching of timer A in event counter mode synchronized to?

A : The increment/decrement function of timer A in event counter mode uses an internally synchronized TAiOUT input so that it can be switched while counting.
The following is an example when the falling edge is selected.



In other words, TAiOUT input is obtained while the TAiIN input is at "L" and it is made effective at "H". Therefore, the second count appears as an up count in the above figure. The third count is a down count because the internally synchronized TAiOUT has already changed to "L". This is also true when the count direction is changed with software.
The count start flag must be set after setting the count direction (internally or externally).
Therefore, the recommended external input signal and event input signal for switching the count direction are as follows:



(This is also true when the falling edge is the effective edge of event input.)

(6) Watchdog timer

Q : 1. When does the watchdog timer start?
2. What action should be taken when the watchdog timer is not used?

A : 1. The watchdog timer starts as soon as reset is deactivated.
2. Apply 2 x V_{CC} to the $\overline{RESET}$ pin.
The watchdog timer may overflow between the time a reset is activated and the $\overline{RESET}$ pin reaches 2 x V_{CC}. This must be compensated by software with one of the following methods:

a) Set the watchdog timer interrupt vector equal to the reset vector.
b) Perform a software reset in the interrupt service routine.
c) Execute an **RTI** instruction without doing anything else.

4.Serial Port
(1) Clock synchronous serial I/O

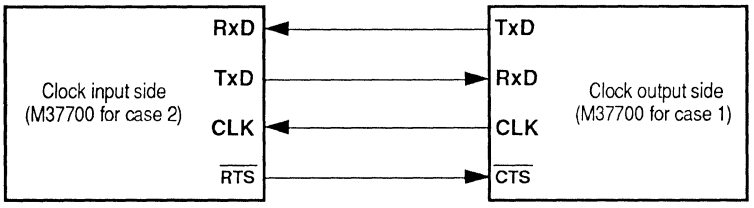
Q : Can the same clock be used to perform output and input (concurrently)?

A : 1.When M37700 outputs the clock

If the communication partner (clock input side) can perform concurrent data I/O using an external clock, data output and input can be performed concurrently according to the procedure shown by the flowchart in Figure 1. However, in this example, the clock input side is assumed to set the M37700 $\overline{CTS}$ pin to "L" when it is ready to receive data and has set the output data.

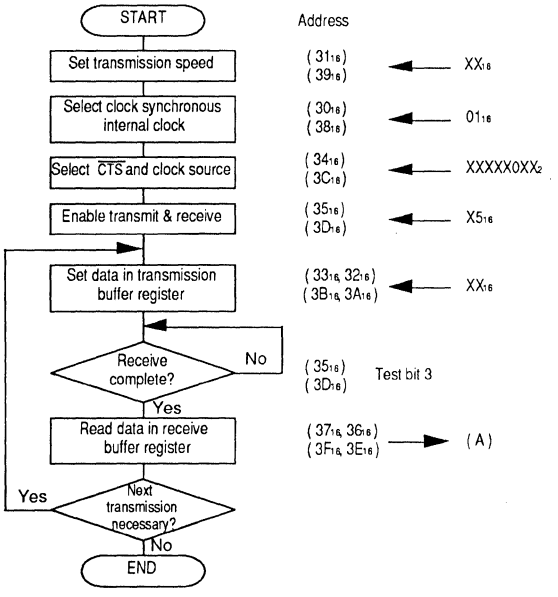
2.When M37700 inputs the clock

If the communication partner (clock output side) can perform concurrent data I/O using its own output clock, data output and input can be performed concurrently according to the procedure shown by the flowchart in Figure 1. However, in this example, the clock output side is assumed to start clock and data output when "L" is output from the M37700 $\overline{RTS}$ pin.



[Serial Connection Diagram]

1) When clock is output



2) When clock is input

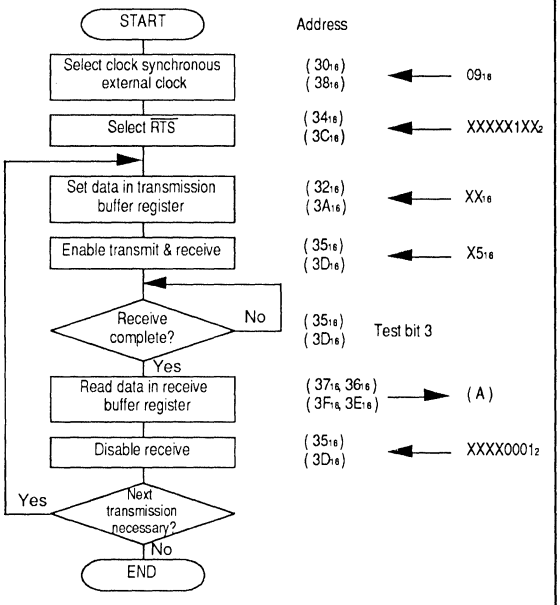


Fig.1 Flowchart when M37700 Performs Clock Input/Output

(2)Clock synchronous serial I/O

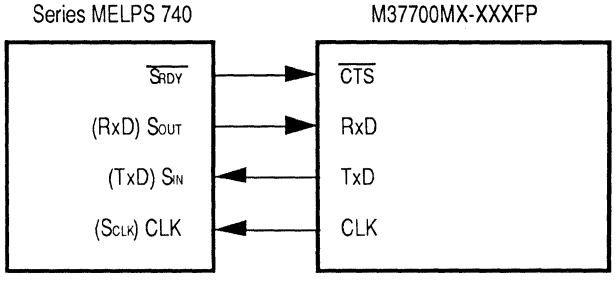
Q : If the synchronous clock is externally input, what happens when clock is input nine or more times?

A : 1.For transmit, shifting stops if the next data is not set. D_7 output from the TxD pin continues.
2.For receive, shifting stops after writing 8 bits of data in the receive buffer register. However, receiving of next data starts when dummy data is written in the transmission register.

(3) Serial Communication with the series MELPS 740

Q : Is it possible to connect the series MELPS 740 microcomputer and the M37700 through a clock synchronous serial I/O?

A : Yes. A connection example is shown below.



Note: When the M37700 $\overline{CTS}$/$\overline{RTS}$ pin is used as an $\overline{RTS}$ pin, use the series MELPS740 $\overline{SRDY}$ pin as normal data input pin and check the level by software or use an external interrupt pin.

5.A-D Conversion
(1) Analog input level

Q : What happens to the A-D converter value when the analog input pin level changes during A-D conversion?

A : The error will become greater when the analog input pin level changes during A-D conversion because the analog input pin is not held. The analog input pin level should be kept constant during A-D conversion which is ϕ_{AD}x57 cycles (ϕ_{AD} is $f(X_{IN})/8$ or $f(X_{IN})/4$) for single conversion of one channel.

(2) V_{REF} pin

Q : How should V_{REF} be set during A-D conversion?

A : Set the voltage between 2V and AV_{CC}.

(3) A-D conversion mode

Q : What happens to the analog input selection bit when the operating mode is changed?

A : Nothing. This bit is ignored in single sweep mode and repeat sweep mode.

(4) A-D conversion rate

Q : Why is the A-D conversion reference clock selectable?

A : This is for future functional enhancement when the M37700 CPU clock is increased. ϕ_{AD} can be either $f(X_{IN})/4$ or $f(X_{IN})/8$ when the oscillating frequency ($f(X_{IN})$) is 8MHz or 16MHz.

(5) A-D converter input current

Q : What is the A-D converter input current?

A : The input current (reference value) to the A-D input port is:
1 to 2 μA ($f(X_{IN})$=8MHz)
2 to 3 μA ($f(X_{IN})$=16MHz)

6.Others

(1) SFR

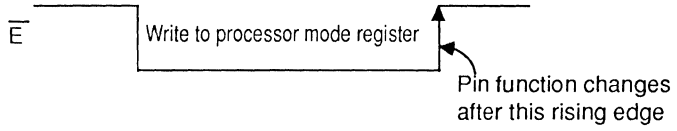
Q:What happens when data is written in an unused or fixed SFR area?

A: 1.One-shot start flag (42_{16}) bit 7 and processor mode register PMR ($5E_{16}$) bit 6 must be set to "0". The result is unpredictable when either is set to "1".
2.Writing to unused registers or bits is ignored, but the result is unpredictable when these values are read.

(2) Switching processor Mode

Q:What happens when the processor mode is changed in the middle of a program?

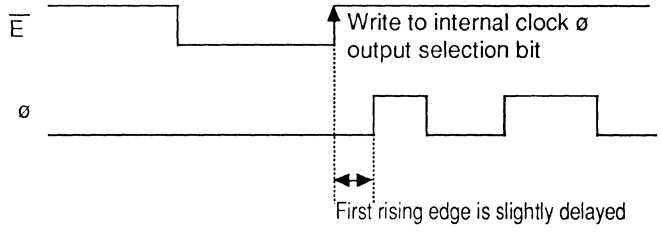
A:When the mode is changed from single-chip mode to other mode or from non-single-chip mode to single-chip mode, the pin function changes to port or bus at the rising edge of $\overline{E}$ after writing to the processor mode register.



(3) ø output

Q:When does ø output start if the internal clock ø output selection bit (bit 7) of the processor mode register PMR ($5E_{16}$) is set to "1" to output ø?

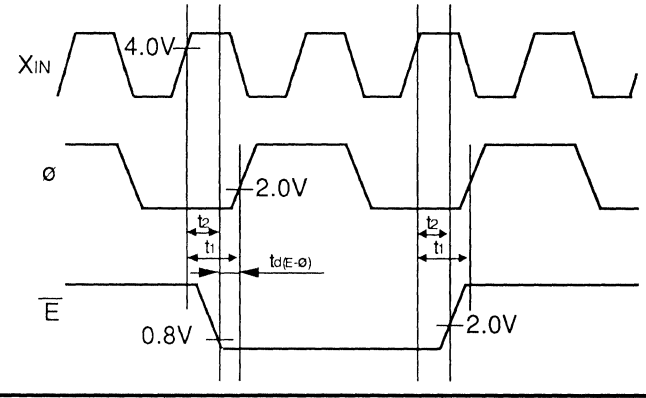
A:ø output starts at the rising edge of $\overline{E}$ that has set the internal clock ø output selection bit to "1". However, the rising edge is delayed for the first "H".



(4) ø output

Q:What is the difference between X_{IN} and output signal ø and $\overline{E}$.

A:The differences are as follows:



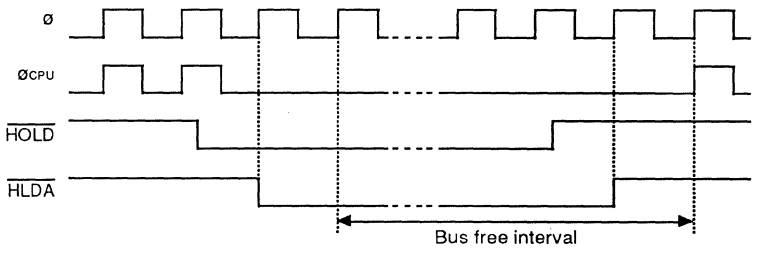
Clock / Parameter	8MHz		16MHz	
	Min.	Max.	Min.	Max.
td(E–ø)	0	30ns	0	20ns
X_{IN}–ø(t_1)	10ns	60ns	10ns	60ns
X_{IN}–$\overline{E}$(t_2)	10ns	50ns	10ns	50ns

(5) HOLD

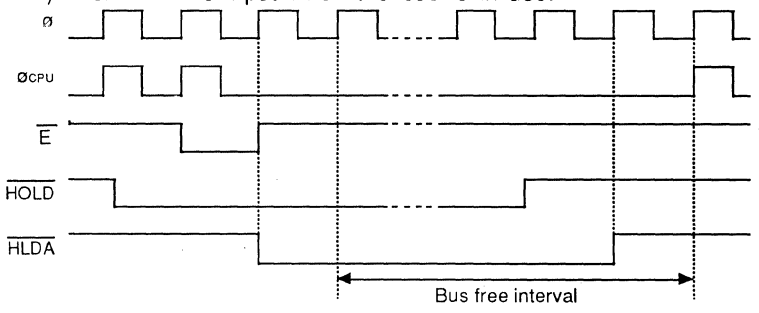
Q: When there is a $\overline{HOLD}$ input, when does the $\overline{HOLD}$ state start and when is $\overline{HLDA}$ output?

A: The timing are as follows:

1)When $\overline{HOLD}$ is input when the bus is not used.



2)When $\overline{HOLD}$ is input when the bus is in use.



Note: $ø_{CPU}$ is an internal signal and cannot be verified externally.

(6) Cycles

Q: If the number of cycles required to execute instructions is not constant for the series MELPS7700, what is meant by "cycles" in the user's manual?

A: They are the cycles required to execute each instruction under the best conditions which are:

- Use internal ROM and RAM
- m=1, x=1
- $DPR_L=00_{16}$
- Two or three data in the instruction queue buffer
- Bus is free when the instruction is started

In actual programming, the memory area, flags, and registers can be controlled from the program, but the number of cycles required to execute an instruction also depends on the status of the instruction queue buffer when the previous instruction completes. Therefore, the total processing speed cannot be calculated just by adding the minimum cycle counts.

(7) Wait

Q: What is the difference between a software wait using a wait bit and a hardware wait using the $\overline{RDY}$ signal? What are the merits of each method?

A: The differences are listed in the table below. The top three items are the merits of hardware wait and the bottom two items are the merits of software wait.

Parameter	Software wait	Hardware wait
Time required for one wait	Twice the time required for zero wait	1.5 times the time required for zero wait
Wait area	All external memory area	Required area only
Number of wait states	One cycle only	1 to ∞
External hardware	Not required	Required
Contribution to access time	Two ø cycles	n ø cycle (n is inserted wait count)

(8) Program Execution Speed

Q:Is there any difference in the program execution speed between single-chip mode and memory expansion mode or microprocessor mode?

A:The instruction execution speed for memory expansion mode or microprocessor mode is the same as in single-chip mode if there is zero wait state due to wait bit or $\overline{RDY}$ input and if "L" level is applied to the BYTE pin (16-bit external bus).
If the above conditions are not satisfied, the execution speed for memory expansion mode and microprocessor mode is slower. Figure 2 shows the difference in execution speed when a sample program is executed under various conditions.

Sample program (24 bit x 24 bit matrix conversion)

```
        LDY     #69 (m=0, x=1)
LOOP0:  LDX     #69
LOOP1:  ASL     ss,X
        SEM
        ROL     ss+2,X
        ROL     B
        CLM
        ROR     A
        DEX
        DEX
        DEX
        BNE     LOOP1
        STA     A,dddd,Y
        SEM
        STA     B,dddd+2,Y
        CLM
        DEY
        DEY
        DEY
        BNE     LOOP0
```

Execution speed in memory expansion mode and microprocessor mode
(Assuming 8MHz single-chip mode execution speed is 1.)

Wait bit	External bus width (and used memory)	Execution Time (ms)	Ratio
1	16 bits	5.54	1.00
	8 bits (internal memory is only RAM) (**Note**)	5.98	1.08
	8 bits	6.52	1.18
0	16 bits (internal memory is only RAM) (**Note**)	6.79	1.23
	8 bits (internal memory is only RAM) (**Note**)	9.78	1.77
	8 bits	10.86	1.96

Note : Memory expansion mode only

Fig.2 Sample Program Execution Speed Comparison

CONTACT ADDRESSES FOR FURTHER INFORMATION

JAPAN
Semiconductor Marketing Division
Mitsubishi Electric Corporation
2-3, Marunouchi 2-chome
Chiyoda-ku, Tokyo 100, Japan
Telex: 24532 MELCO J
Telephone: (03) 218-3473
(03) 218-3499
Facsimile: (03) 214-5570

Overseas Marketing Manager
Kita-Itami Works
4-1, Mizuhara, Itami-shi,
Hyogo-ken 664, Japan
Telex: 526408 KMELCO J
Telephone: (0727) 82-5131
Facsimile: (0727) 72-2329

HONG KONG
MITSUBISHI ELECTRIC (H.K.) LTD.
25 Floor, Leighton Centre,
77, Leighton Road. Causeway Bay.
Hong Kong
Telex: 60800 MELCO HX
Telephone: (5) 773901-3
Facsimile: (5) 895-3104

SINGAPORE
MELCO SALES SINGAPORE PTE. LTD.
230 Upper Bukit Timah Road # 03-01/15
Hock Soon Industrial Complex
Singapore 2158
Telex: RS 20845 MELCO
Telephone: 4695255
Facsimile: 4695347

TAIWAN
MELCO-TAIWAN CO., Ltd.
1st fl., Chung-Ling Bldg.,
363, Sec. 2, Fu-Hsing S Road,
Taipei R.O.C.
Telephone: (02) 735-3030
Facsimile: (02) 735-6771
Telex: 25433 CHURYO "MELCO-TAIWAN"

U.S.A.

NORTHWEST
Mitsubishi Electronics America, Inc.
1050 East Arques Avenue
Sunnyvale, CA 94086
Telephone: (408) 730-5900
Facsimile: (408) 730-4972

SAN DIEGO
Mitsubishi Electronics America, Inc.
11545 West Bernardo Court
Suite 100
San Diego, CA 92128
Telephone: (619) 592-1445
Facsimile: (619) 592-0242

DENVER
Mitsubishi Electronics America, Inc.
4600 South Ulster Street
Metropoint Building, 7th Floor
Denver, CO 80237
Telephone: (303) 740-6775
Facsimile: (303) 694-0613

SOUTHWEST
Mitsubishi Electronics America, Inc.
991 Knox Street
Torrance, CA 90502
Telephone: (213) 515-3993
Facsimile: (213) 217-5781

SOUTH CENTRAL
Mitsubishi Electronics America, Inc.
1501 Luna Road, Suite 124
Carrollton, TX 75006
Telephone: (214) 484-1919
Facsimile: (214) 243-0207

NORTHERN
Mitsubishi Electronics America, Inc.
15612 Highway 7 #243
Minnetonka, MN 55345
Telephone: (612) 938-7779
Facsimile: (612) 938-5125

NORTH CENTRAL
Mitsubishi Electronics America, Inc.
800 N. Bierman Circle
Mt. Prospect, IL 60056
Telephone: (312) 298-9223
Facsimile: (312) 298-0567

NORTHEAST
Mitsubishi Electronics America, Inc.
200 Unicorn Park Drive
Woburn, MA 01801
Telephone: (617) 932-5700
Facsimile: (617) 938-1075

MID-ATLANTIC
Mitsubishi Electronics America, Inc.
800 Cottontail Lane
Somerset, NJ 08873
Telephone: (201) 469-8833
Facsimile: (201) 469-1909

SOUTH ATLANTIC
Mitsubishi Electronics America, Inc.
2500 Gateway Center Blvd., Suite 300
Morrisville. NC 27560
Telephone: (404) 368-4850
Facsimile: (404) 662-5208

SOUTHEAST
Mitsubishi Electronics America, Inc.
Town Executive Center
6100 Glades Road #210
Boca Raton, FL 33433
Telephone: (407) 487-7747
Facsimile: (407) 487-2046

CANADA
Mitsubishi Electronics America, Inc.
6185 Ordan Drive, Unit #110
Mississauga, Ontario, Canada L5T 2E1
Telephone: (416) 670-8711
Facsimile: (416) 670-8715

Mitsubishi Electronics America, Inc.
300 March Road, Suite 302
Kanata, Ontario, Canada K2K 2E2
Telephone: (416) 670-8711
Facsimile: (416) 670-8715

WEST GERMANY
Mitsubishi Electric Europe GmbH
Headquarters:
Gothear Str. 8
4030 Ratingen 1, West Germany
Telex: 8585070 MED D
Telephone: (02102) 4860
Facsimile: (02102) 486-115

Munich Office:
Arabellastraße 31
8000 München 81, West Germany
Telex: 5214820
Telephone: (089) 919006-09
Facsimile: (089) 9101399

FRANCE
Mitsubishi Electric Europe GmbH
55, Avenue de Colmar
92563 Rueil Malmaison Cedex
Telex: 632326
Telephone: 47087871
Facsimile: 47513622

ITALY
Mitsubishi Electric Europe GmbH
Centro Direzionale Colleoni
Palazzo Cassiopea 1
20041 Agrate Brianza I-Milano
Telephone: (039) 636011
Facsimile: (039) 6360120

SWEDEN
Mitsubishi Electric Europe GmbH
Lastbilsvägen 6B
5-19149 Sollentuna, Sweden
Telex: 10877 (meab S)
Telephone: (08) 960468
Facsimile: (08) 966877

U.K.
Mitsubishi Electric (U.K.) Ltd.
Travellers Lane
Hatfield
Herts AL10 8XB, England, U.K.
Telephone: (0044) 7072 76100
Facsimile: (0044) 7072 78692

AUSTRALIA
Mitsubishi Electric Australia Pty. Ltd.
73-75, Epping Road, North Ryde,
P.O. Box 1567, Macquarie Centre,
N.S.W., 2113, Australia
Telex: MESYD AA 26614
Telephone: (02) (888) 5777
Facsimile: (02) (887) 3635

MITSUBISHI SEMICONDUCTORS
M37700M2-XXXFP, M37700M2AXXXFP USER'S MANUAL

December First Edition 1989
Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

MITSUBISHI SEMICONDUCTORS

M37700M2-XXXFP
M37700M2AXXXFP

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

H-E0214-A Printed in Japan (ROD)

New publication, effective Dec. 1989.
Specifications subject to change without notice.