USER'S MANUAL

7470 Series

MITSUBISHI ELECTRIC

MITSUBISHI SEMICONDUCTORS

7470 Series

USER'S MANUAL

MITSUBISHI ELECTRIC

FOREWORD

This users manual describes the hardware function of the CMOS 8-bit single-chip microcomputers, 7470 series. The contents of this manual are Overview, Functional description, Electrical characteristics, Built-in PROM version microcomputers description, Emulator MCU description, and Applications.
Please use the "Series MELPS 740 Software Manual" about their instruction set.

CONTENTS

1. **OVERVIEW** 1
1.1 Overview 2
1.2 7470 series 3
1.3 Functional descriptions 4
1.4 Pin configuration 6
1.5 Pin description 9
1.6 Block diagram 10
2. **FUNCTIONAL DESCRIPTION** 13
2.1 Central processing unit (CPU) 14
2.1.1 Accumulator (A) 15
2.1.2 Index register X (X), index register Y (Y) 15
2.1.3 Stack pointer (S) 15
2.1.4 Program counter (PC) 16
2.1.5 Processor status register (PS) 16
2.2 Access area 18
2.2.1 Zero page 19
2.2.2 Special page 19
2.3 Memory map 20
2.4 Input/Output ports 26
2.4.1 Input/Output port 27
2.4.2 Pin description 30
2.5 Interrupts 35
2.5.1 Interrupt causes 36
2.5.2 Interrupt control 40
2.5.3 Processing at interrupt acceptance 41
2.5.4 Timing after interrupt 42
2.6 Timers 43
2.6.1 Timer 1, timer 2, timer 3 and timer 4 45
2.6.2 PWM mode 51
2.6.3 Input latch function 52
2.7 Serial I/O 54
2.7.1 Structure of serial I/O section 54
2.7.2 Serial I/O data receive 56
2.7.3 Serial I/O data transmit 57
2.7.4 Byte specification mode 61
2.7.5 $\overline{\text{SRDY}}$ signal and SARDY signal 62
2.8 A-D converter 66
2.8.1 Block description 67
2.8.2 Method of use 68
2.8.3 Operation 69
2.8.4 Definition of A-D conversion precision 70
2.9 Key on wake up 72
2.10 Reset circuit 73
2.10.1 Reset operation 73
2.10.2 Internal status of microcomputer at reset release 74
2.11 Oscillation circuit 75
2.11.1 Oscillation circuit 75
2.11.2 Oscillating circuit for clock-function clock 77
2.12 Low power consumption modes 78
2.12.1 Stop mode 78
2.12.2 Wait mode 79
2.13 Status transitions 80

3.ELECTRIC CHARACTERISTICS 83
3.1 Electric characteristics of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP 84
3.2 Electric characteristics of M37471M2-XXXSP/FP, M37471M4-XXXSP/FP, and M37471M8-XXXSP/FP 87
3.3 Standard characteristics 90
4.BUILT-IN PROGRAMMABLE ROM VERSION 91
4.1 Overview 92
4.2 Pin configuration 93
4.3 Block diagram 95
4.4 EPROM mode 98
4.4.1 EPROM mode 98
4.4.2 Pin description 100
4.4.3 Reading, writing, and erasure of built-in PROM 102
4.4.4 Notes on handling 103
4.5 Electric Characteristics 104
4.5.1 Electric characteristics of M37470E4-XXXSP and M37470E8-XXXSP 104
4.5.2 Electric characteristics of M37471E4-XXXSP/FP, M37471E8-XXXSP/FP and M37471E8SS .. 107
APPENDIX 111
Appendix 1 Handling of unused pins 112
Appendix 2 Package outline 113
Appendix 3 Notes for usage 116
Appendix 4 7470 series memory map 118
Appendix 5 SFR memory map 121
Appendix 6 Control registers 122
Appendix 7 Machine language instruction table 131
Appendix 8 Instruction code table 137
Appendix 9 Mask ROM ordering method 138
Appendix 10 Emulator MCU M37471RSS 154
Appendix 11 Applecation circuit example 156

FIGURES

Fig.1.4.1 Pin configuration (M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP) 6
Fig.1.4.2 Pin configuration (M37471M2-XXXSP, M37471M4-XXXSP, and M37471M8-XXXSP) 7
Fig.1.4.3 Pin configuration (M37471M2-XXXFP, M37471M4-XXXFP, and M37471M8-XXXFP) 8
Fig.1.6.1 Block diagram of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP 10
Fig.1.6.2 Block diagram of M37471M2-XXXSP, M37471M4-XXXSP, and M37471M8-XXXSP 11
Fig.1.6.3 Block diagram of M37471M2-XXXFP, M37471M4-XXXFP, and M37471M8-XXXFP 12
Fig.2.1.1 Register structure 14
Fig.2.1.2 Stack store and restore sequence when executing interrupt or subroutine calls 16
Fig.2.2.1 Access area (M37470M2-XXXSP) 18
Fig.2.2.2 Zero page and special page addressing mode 19
Fig.2.3.1 Memory map for M37470M2, and M37471M2 21
Fig.2.3.2 Memory map for M37470M4, and M37471M4 22
Fig.2.3.3 Memory map for M37470M8, and M37471M8 23
Fig.2.3.4 SFR memory map 24
Fig.2.3.5 Interrupt vector area memory map 25
Fig.2.4.1 I/O ports of M37470M2, M37470M4, and M37470M8 26
Fig.2.4.2 I/O ports of M37471M2, M37471M4, and M37471M8 26
Fig.2.4.3 Structure of port P0 pull-up control register 27
Fig.2.4.4 Structure of ports P1 to P5 pull-up control register 28
Fig.2.4.5 Port P0, $P1_0$–$P1_3$ block diagram 31
Fig.2.4.6 Port $P1_4$–$P1_7$ block diagram 32
Fig.2.4.7 Port P2–P4 block diagram 33
Fig.2.4.8 Port P5 block diagram 34
Fig.2.5.1 Structure of interrupt related registers (1) 37
Fig.2.5.2 Structure of interrupt related registers (2) 38
Fig.2.5.3 Structure of interrupt related registers (3) 39
Fig.2.5.4 Interrupt control 40
Fig.2.5.5 Change of stack pointer and program counter at interrupt receive 41
Fig.2.5.6 Timing chart after interrupt 42
Fig.2.5.7 Operation time before the interrupt routine execution 42
Fig.2.6.1 Timer block diagram 44
Fig.2.6.2 Timer count timing 45
Fig.2.6.3 Structure of timer 1, 2 mode register 46
Fig.2.6.4 Timing chart for timer 2 when its count source is timer 1 overflow signal 47
Fig.2.6.5 Structure of timer 3, 4 mode register 49
Fig.2.6.6 Structure of timer FF register 49
Fig.2.6.7 PWM waveform 51
Fig.2.6.8 Structure of timer mode register 2 51
Fig.2.6.9 Structure of input latch register 52
Fig.2.6.10 Structure of active edge selection register 53
Fig.2.7.1 Serial I/O block diagram 54
Fig.2.7.2 Structure of serial I/O mode register 55
Fig.2.7.3 Example of the serial I/O mode register setting (data receive) 56
Fig.2.7.4 Serial I/O register at receiving 56
Fig.2.7.5 Example of the serial I/O mode register setting (data transmit) 57
Fig.2.7.6 Serial I/O register at transmitting 57
Fig.2.7.7 Connection example of serial I/O transfer (normal mode) 58
Fig.2.7.8 Serial I/O data transfer operation sequence (normal mode) 59
Fig.2.7.9 Timing diagram of serial I/O data transfer (normal mode) 60
Fig.2.7.10 Example of the serial I/O mode register setting (byte specification mode) 61
Fig.2.7.11 Connection example of serial I/O transfer (byte specification mode) 63
Fig.2.7.12 Serial I/O data transfer operation sequence (byte specification mode) 64
Fig.2.7.13 Timing diagram of serial I/O data transfer(byte specification mode with SARDY signal) 65

CONTENTS

Fig.2.8.1 A-D converter block diagram ... 66
Fig.2.8.2 Structure of A-D control register ... 67
Fig.2.8.3 Definition of A–D conversion precision ... 71
Fig.2.9.1 Block diagram of interrupt input and key on wake up circuit ... 72
Fig.2.10.1 Timing diagram at reset ... 73
Fig.2.10.2 Internal status of microcomputer at reset release ... 74
Fig.2.11.1 Clock generation circuit block diagram ... 75
Fig.2.11.2 Oscillation circuit using ceramic resonator or crystal oscillator (M37470) ... 76
Fig.2.11.3 Oscillation circuit using ceramic resonator or crystal oscillator (M37471) ... 76
Fig.2.11.4 External clock input circuit (M37470) ... 76
Fig.2.11.5 External clock input circuit (M37471) ... 76
Fig.2.11.6 Structure of CPU mode register ... 76
Fig.2.13.1 Status transitions ... 80
Fig.4.2.1 Pin configuration (M37470E4-XXXSP and M37470E8-XXXSP) ... 93
Fig.4.2.2 Pin configuration (M37471E4-XXXSP, M37471E8-XXXSP and M37471E8SS) ... 93
Fig.4.2.3 Pin configuration (M37471E4-XXXFP and M37471E8-XXXFP) ... 94
Fig.4.3.1 Block diagram of M37470E4-XXXSP and M37470E8-XXXSP ... 95
Fig.4.3.2 Block diagram of M37471E4-XXXSP and M37471E8-XXXSP(M37471E8SS) ... 96
Fig.4.3.3 Block diagram of M37471E4-XXXFP and M37471E8-XXXFP ... 97
Fig.4.4.1 Pin connection at EPROM mode (M37470E4-XXXSP and M37470E8-XXXSP) ... 98
Fig.4.4.2 Pin connection at EPROM mode (M37471E4-XXXSP, M37471E8-XXXSP and M37471E8SS) ... 99
Fig.4.4.3 Pin connection at EPROM mode (M37471E4-XXXFP and M37471E8-XXXFP) ... 99
Fig.4.4.4 Writing and test for blank one-time programmable type ... 103

TABLES

Table 1.2.1 7470 series 3
Table 1.3.1 Functions of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP 4
Table 1.3.2 Functions of M37471M2-XXXSP/FP, M37471M4-XXXSP/FP, and M37471M8-XXXSP/FP 5
Table 1.5.1 Pin description 9
Table 2.5.1 Interrupt vector addresses and priority 35
Table 2.7.1 Corresponding between serial I/O relation ports and serial I/O mode register. 58
Table 2.8.1 Change of A-D conversion register during A-D conversion 69
Table 2.12.1 Functions operating in stop mode 78
Table 2.12.2 Functions operating in wait mode 79
Table 4.1.1 Functions of built-in PROM versions (M37470E4-XXXSP, M37470E8-XXXSP, M37471E4-XXXSP/FP, M37471E8-XXXSP/FP, and M37471E8SS) 92
Table 4.4.1 Pin correspondence at the EPROM mode 98
Table 4.4.2 Pin description 100
Table 4.4.3 Input/Output signal at each mode 102

CHAPTER 1

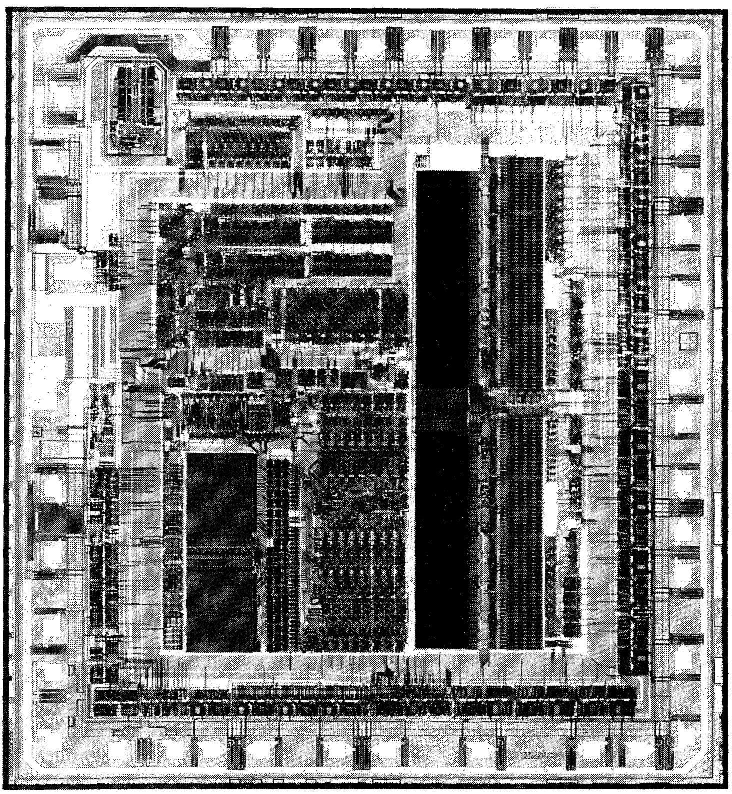
OVERVIEW

1.1 Overview

The 7470 series are 8-bit single-chip microcomputers created by silicon gate CMOS processing. Built into these microcomputers are:

- MELPS 740 CPU core
- Four 8-bit timers
- Serial I/O function (with multi-connection compatibility)
- A-D converter (successive approximation comparison method)
- Key on wake up function

Designed as compact microcomputers for controlling household appliances and every-day electrical equipment, their reduced power dissipations enabled by the use of CMOS processing also make these microcomputers extremely suitable for applications using battery power.



M37471M2-XXXSP chip

1.2 7470 series

The 7470 series is an extended family consisting of the chips listed in Table 1.2.1, with the M37470M2-XXXSP being the base chip. These chips differ only in memory characteristics, memory size, and number of ports, to enable the user to select the chip best suited to the system. Throughout this manual, all of the 7470 series of microcomputers are referred to as the M37470, unless there is a real difference between models.

Table 1.2.1 7470 series

Type	ROM (byte)	RAM (byte)	I/O ports	Input ports	Analog input	Package	Remarks
M37470M2-XXXSP	4K	128					
M37470M4-XXXSP	8K	192					—
M37470M8-XXXSP	16K	384					
M37470E4-XXXSP	8K	192	22	4	4	32P4B	
M37470E8-XXXSP	16K	384					*1
M37470E4SP	8K	192					
M37470E8SP	16K	384					
M37471M2-XXXSP	4K	128					
M37471M4-XXXSP	8K	192				42P4B	
M37471M8-XXXSP	16K	384					
M37471M2-XXXFP	4K	128					—
M37471M4-XXXFP	8K	192				56P6N	
M37471M8-XXXFP	16K	384					
M37471E4-XXXSP	8K	192				42P4B	
M37471E8-XXXSP	16K	384					
M37471E4-XXXFP	8K	192	28	8	8	56P6N	
M37471E8-XXXFP	16K	384					
M37471E4SP	8K	192				42P4B	*1
M37471E8SP	16K	384					
M37471E4FP	8K	192				56P6N	
M37471E8FP	16K	384					
M37471E8SS	16K	384				42S1B	*2
M37471RSS	63.5K (Note)	384				42S1M	*3

*1: One-time programmable version
*2: Window-type EPROM version
*3: Dedicated emulator MCU
Note: Address space that can be used as ROM area

(1) One-time programmable version
Non-erasable programs can be written into the internal PROM of this one-time programmable microcomputer. For details of the functions of this version, see Chapter 4, "Built-in programmable ROM version".

(2) Window-type EPROM version
Erasable programs can be written into the built-in EPROM of this built-in EPROM microcomputer. For details of the functions of this version, see Chapter 4, "Built-in programmable ROM version".

(3) Dedicated MCU for emulator
This dedicated MCU for emulator version is designed for program development—since it makes it easy for the user to develop programs, it is most suitable as an element in the trial manufacture of systems. For details of the functions of this version, see Appendix 11, "Dedicated MCU for emulator M37471RSS".

1.3 Functional descriptions

Table 1.3.1 shows the functional descriptions of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP. Same as this, the functional description of M37471M2-XXXSP/FP, M37471M4-XXXSP/FP, and M37471M8-XXXSP/FP are shown in Table 1.3.2.

Table 1.3.1 Functions of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP

Parameter			Functions
Basic instructions			69
Instruction execution time			1.0μs (minimum instructions, at 4MHz)
Clock frequency			4MHz
Memory size	ROM	M37470M2	4096 bytes
		M37470M4	8192 bytes
		M37470M8	16384 bytes
	RAM	M37470M2	128 bytes
		M37470M4	192 bytes
		M37470M8	384 bytes
Input/Output ports	P0,P1	I/O	8-bit × 2
	P2	I/O	4-bit × 1
	P3	Input	4-bit × 1
	P4	I/O	2-bit × 1
Serial I/O			8-bit × 1
Timers			8-bit × 4 (with 8-bit latch)
A-D converter (successive approximation comparison)			8-bit × 1 (4-channel)
Interrupts			External 5, internal 6 and software 1
Subroutine nesting	M37470M2		64
	M37470M4		96
	M37470M8		192
Clock generating circuit (Note 1)			Built-in with internal feedback resistor (connect an external ceramic or quartz crystal oscillator)
Power supply			2.7 to 5.5V
Power dissipation (typ.)			17.5mW (at 4MHz)
Input/Output characteristics	Input/Output voltage		5V
	Output current		–5 to 10mA (P0, P1, P2 and P4: CMOS 3-state)
Operating temperature range			–20 to 85°C
Device structure			CMOS silicon gate
Package			32-pin shrink plastic molded DIP

Note 1: Clock generating circuit for clock function is not built-in.

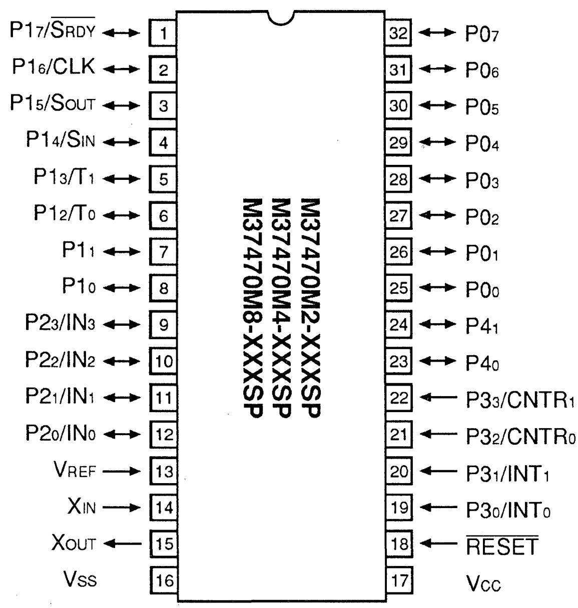
Table 1.3.2 Functions of M37471M2-XXXSP/FP, M37471M4-XXXSP/FP, and M37471M8-XXXSP/FP

Parameter			Functions
Basic instructions			69
Instruction execution time			1.0μs (minimum instructions, at 4MHz)
Clock frequency			4MHz
Memory size	ROM	M37471M2	4096 bytes
		M37471M4	8192 bytes
		M37471M8	16384 bytes
	RAM	M37471M2	128 bytes
		M37471M4	192 bytes
		M37471M8	384 bytes
Input/Output ports	P0,P1	I/O	8-bit × 2
	P2	I/O	8-bit × 1
	P3	Input	4-bit × 1
	P4	I/O	4-bit × 1
	P5	Input	4-bit × 1
Serial I/O			8-bit × 1
Timers			8-bit × 4 (with 8-bit latch)
A-D converter (successive approximation comparison)			8-bit × 1 (8-channel)
Interrupts			External 5, internal 6 and software 1
Subroutine nesting	M37471M2		64
	M37471M4		96
	M37471M8		192
Clock generating circuit			Two built-in with internal feedback resistor (connect an external ceramic or quartz crystal oscillator)
Power supply			2.7 to 5.5V
Power dissipation (typ.)			17.5mW (at 4MHz)
Input/Output characteristics	Input/Output voltage		5V
	Output current		–5 to 10mA (P0, P1, P2 and P4: CMOS 3-state)
Operating temperature range			–20 to 85°C
Device structure			CMOS silicon gate
Package	M37471M2-XXXSP M37471M4-XXXSP M37471M8-XXXSP		42-pin shrink plastic molded DIP
	M37471M2-XXXFP M37471M4-XXXFP M37471M8-XXXFP		56-pin plastic molded QFP

1.4 Pin configuration

Figure 1.4.1 shows the pin configuration of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP. And the pin configuration of M37471M2-XXXSP, M37471M4-XXXSP, and M37471M8-XXXSP are shown in Figure 1.4.2, and the pin configuration of M37471M2-XXXFP, M37471M4-XXXFP, and M37471M8-XXXFP are shown in Figure 1.4.3.

Pin configuration (top view)



Outline 32P4B

Fig.1.4.1 Pin configuration (M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP)

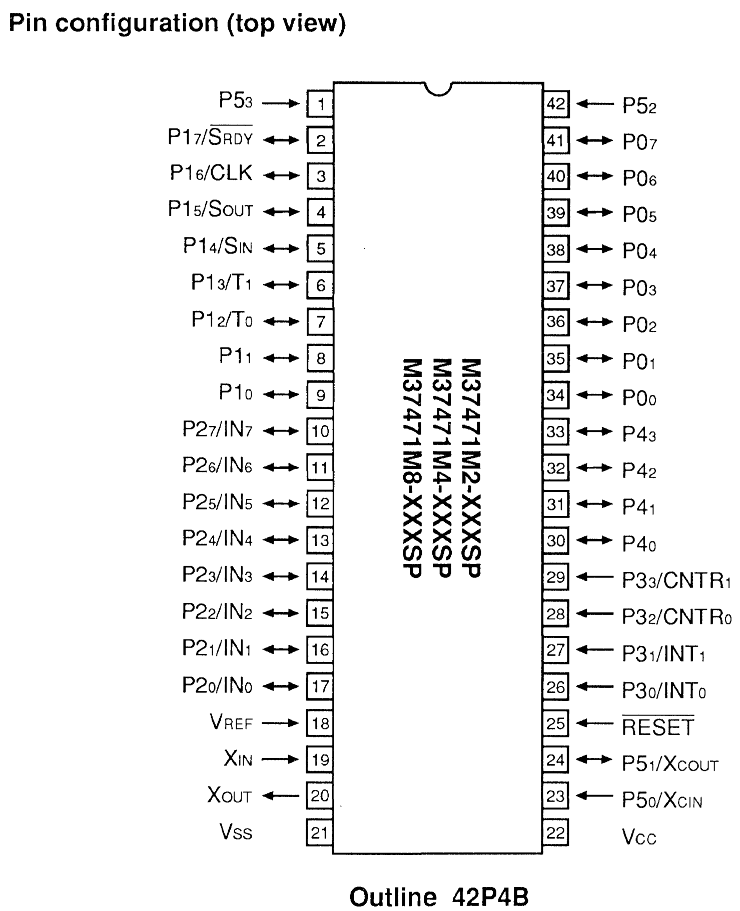
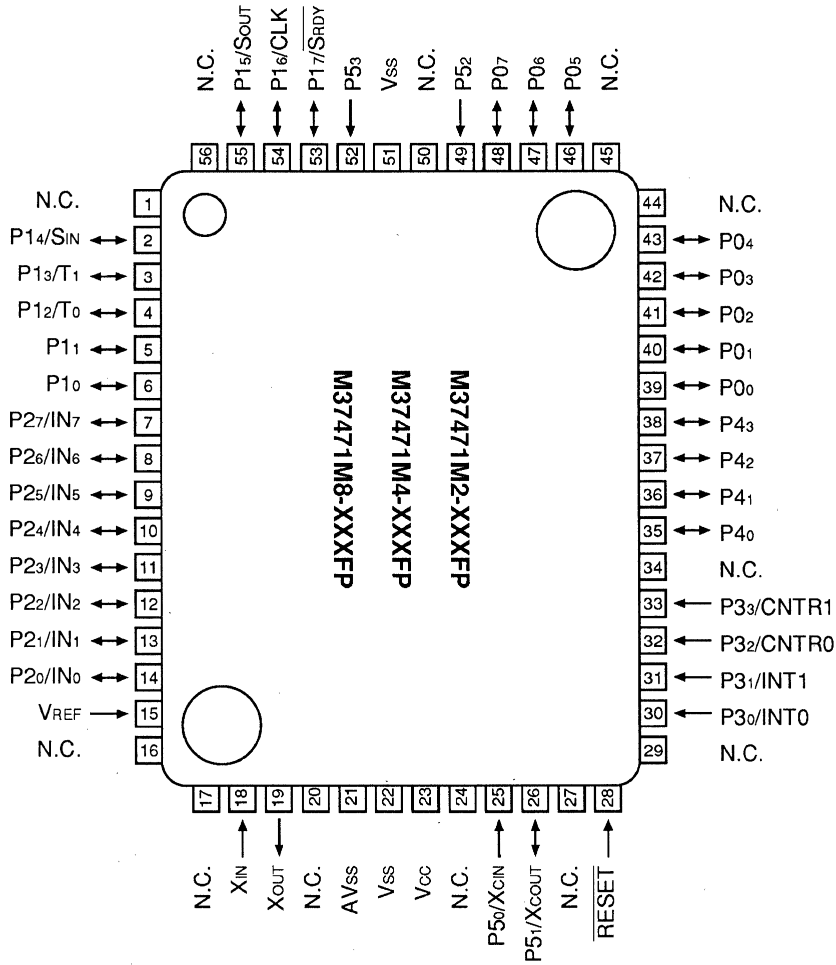


Fig.1.4.2 Pin configuration (M37471M2-XXXSP, M37471M4-XXXSP, and M37471M8-XXXSP)

Pin configuration (top view)



Outline 56P6N

N.C. : No connection

Fig.1.4.3 Pin configuration (M37471M2-XXXFP, M37471M4-XXXFP, and M37471M8-XXXFP)

1.5 Pin description

Pin description is shown in Table 1.5.1

Table 1.5.1 Pin description

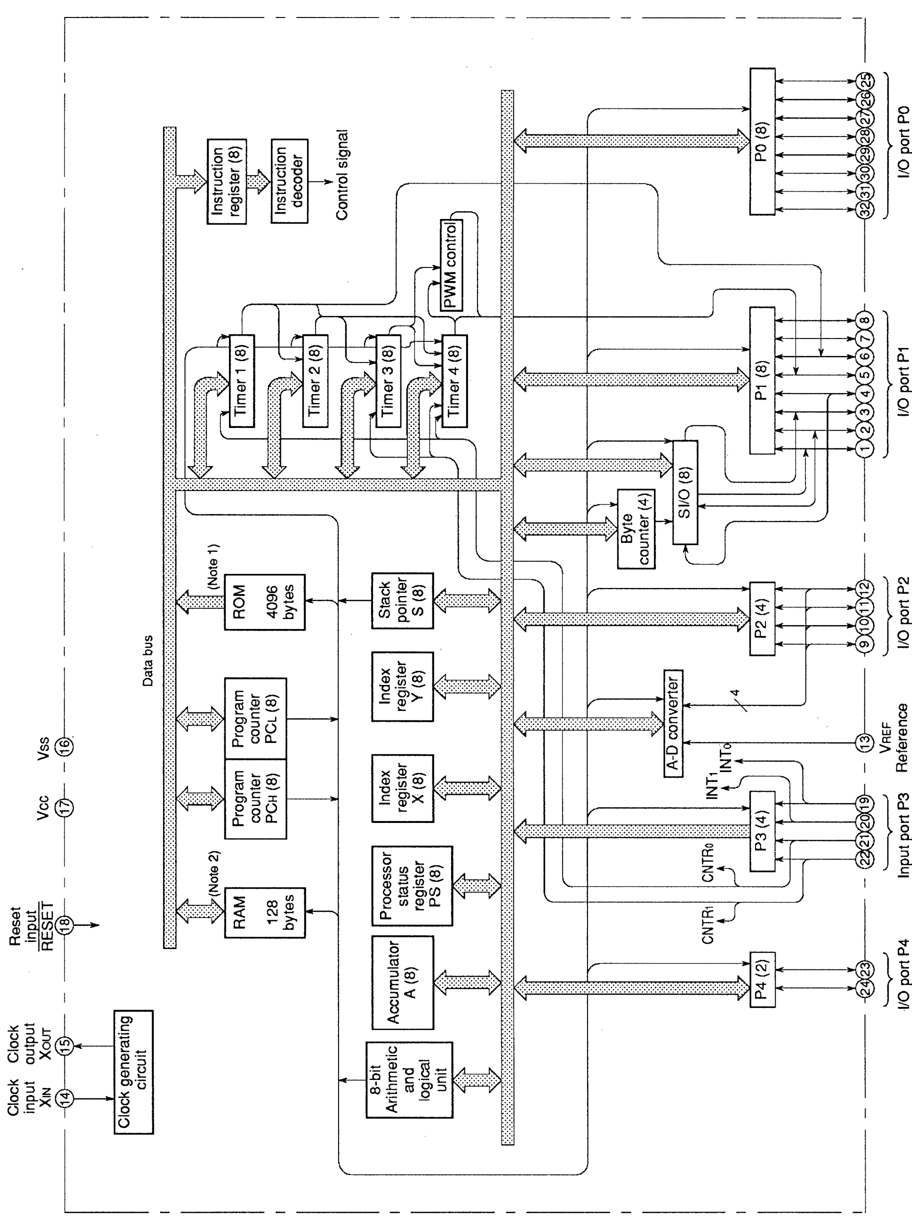
Pin	Name	Input/ Output	Function
V_{CC}, V_{SS}	Supply voltage		Supply 2.7 to 5.5V to V_{CC}, and 0V to V_{SS}.
AV_{SS}	Analog power supply		Acts as ground level input pin for A-D converter. Same voltage as V_{SS} is applied. (Note 1)
V_{REF}	Reference voltage input	Input	Acts as reference voltage input pin for the A-D converter.
$\overline{RESET}$	Reset input	Input	Specifies reset when held at "L" for at least 2μs.
X_{IN}	Clock input	Input	Acts as input and output pins interfacing with the internal clock generating circuit. Connect a ceramic resonator or crystal oscillator between the X_{IN} and X_{OUT} pins to set the oscillator frequency. An internal feedback resistor is connected between the X_{IN} and X_{OUT} pins. If an external clock is used, connect the clock source to the X_{IN} pin and leave the X_{OUT} pin open.
X_{OUT}	Clock output	Output	
$P0_0$–$P0_7$	I/O port P0	I/O	Acts as 8-bit I/O port with CMOS output format. When input port is selected, these pins can be connected individually to pull-up transistors. A key on wake up function is also provided.
$P1_0$–$P1_7$	I/O port P1	I/O	Acts as 8-bit I/O port with CMOS output format. When input port is selected, these pins can be connected in groups of four to pull-up transistors. $P1_2$ and $P1_3$ can also be used as timer outputs T_0 and T_1, and $P1_4$, $P1_5$, $P1_6$, and $P1_7$ can also be used as S_{IN}, S_{OUT}, CLK, and $\overline{S_{RDY}}$ of the serial I/O function. S_{OUT} and $\overline{S_{RDY}}$ outputs can be set to N-channel open drain output.
$P2_0$–$P2_7$ (Note 2)	I/O port P2	I/O	Acts as 8-bit I/O port with CMOS output format. When input port is selected, these pins can be connected in groups of four to pull-up transistors. These pins can also be used as analog inputs IN_0 to IN_7.
$P3_0$–$P3_3$	Input port P3	Input	Acts as 4-bit input port. $P3_0$ and $P3_1$ can also be used as external interrupt input pins INT_0 and INT_1, and $P3_2$ and $P3_3$ can also be used as timer input pins $CNTR_0$ and $CNTR_1$.
$P4_0$–$P4_3$ (Note 3)	I/O port P4	I/O	Acts as 4-bit I/O port with CMOS output format. When input port is selected, these pins can be connected in groups of four to pull-up transistors.
$P5_0$–$P5_3$ (Note 4)	Input port P5	Input	Acts as 4-bit input port that can be connected as a group of four pins to pull-up transistors. $P5_0$ and $P5_1$ can also be used as the X_{CIN} and X_{COUT} pins for the clock-function clock generating circuit. When using these pins as X_{CIN} and X_{COUT} pins, an internal feedback resistor is connected between them. To enable external clock input, connect the clock source to the X_{CIN} pin and leave the X_{COUT} pin open.

Note 1: For 56-pin QFP type only.
2: Only $P2_0$–$P2_3$ (IN_0–IN_3) 4-bit for M37470M2, M37470M4, and M37470M8.
3: Only $P4_0$ and $P4_1$ 2-bit for M37470M2, M37470M4, and M37470M8.
4: This port is not included in M37470M2, M37470M4, and M37470M8.

1.6 Block diagram

Figure 1.6.1 shows the block diagram of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP. And the block diagram of M37471M2-XXXSP, M37471M4-XXXSP, and M37471M8-XXXSP are shown in Figure 1.6.2, and the block diagram of M37471M2-XXXFP, M37471M4-XXXFP, and M37471M8-XXXFP are shown in Figure 1.6.3.

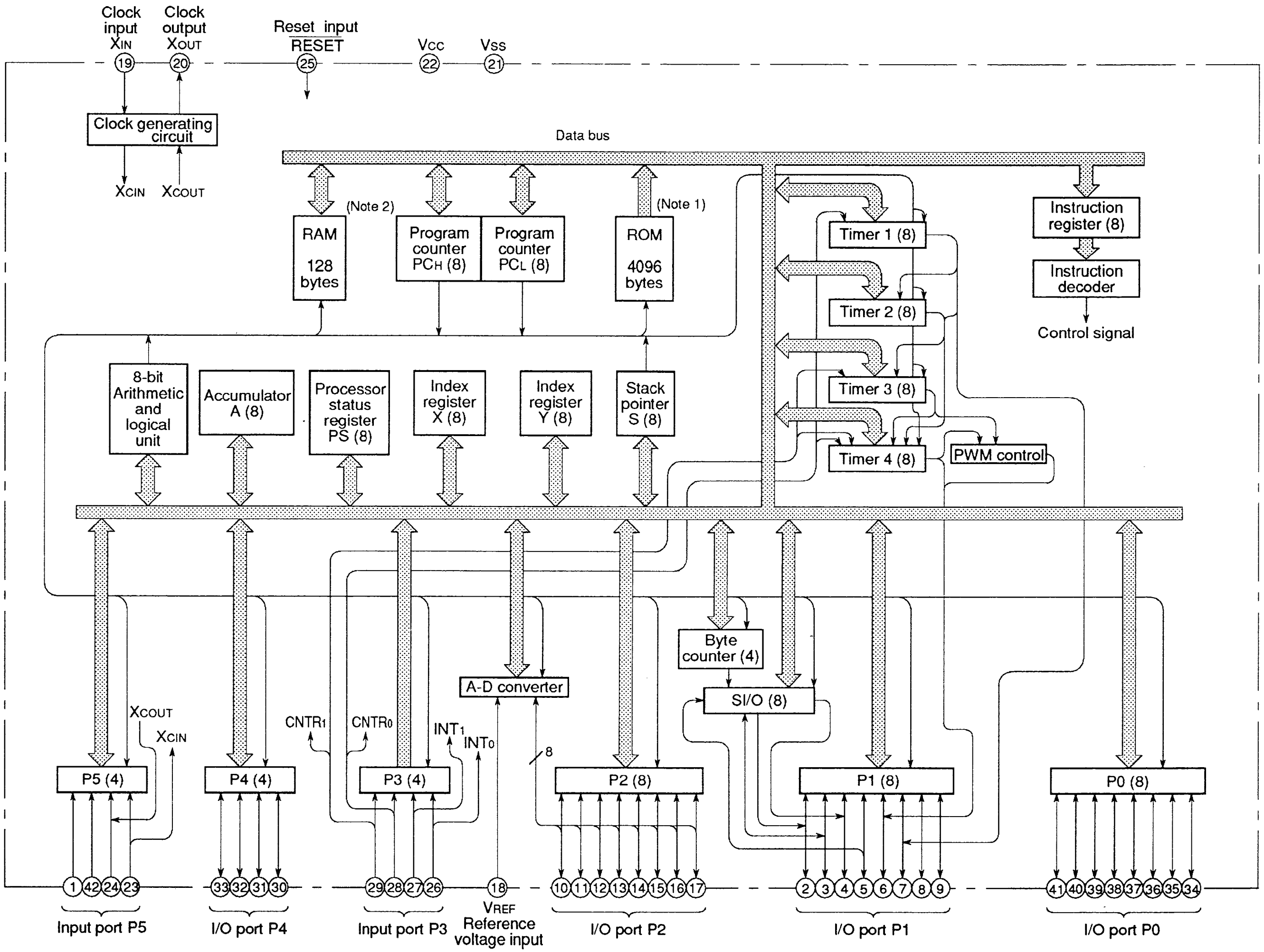
M37470M2-XXXSP block diagram



Note 1 : 8192 bytes for M37470M4, and 16384 bytes for M37470M8.
2 : 192 bytes for M37470M4, and 384 bytes for M37470M8.

Fig.1.6.1 Block diagram of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP

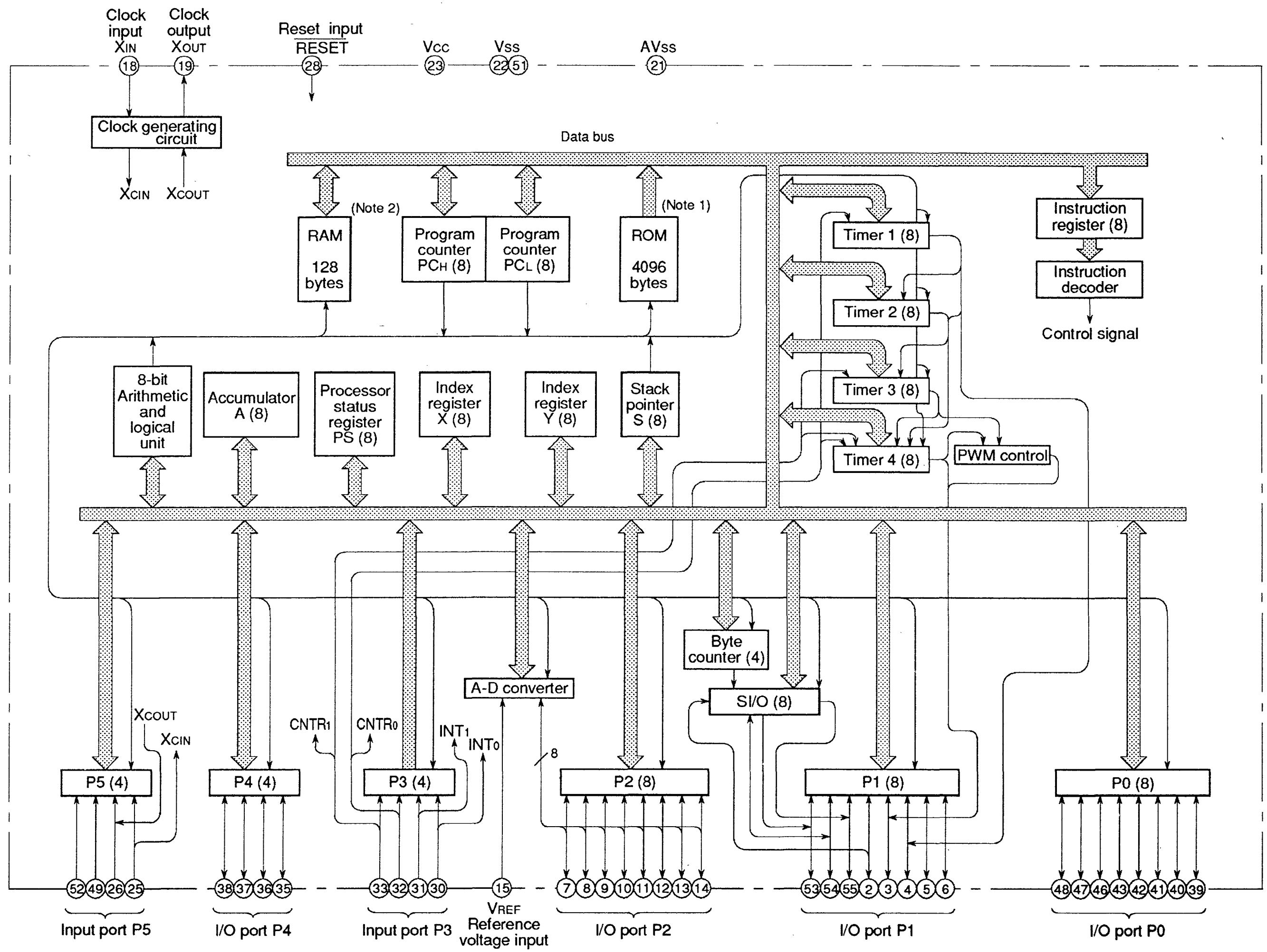
M37471M2-XXXSP block diagram



Note 1 : 8192 bytes for M37471M4–XXXSP, and 16384 bytes for M37471M8–XXXSP.
2 : 192 bytes for M37471M4–XXXSP, and 384 bytes for M37471M8–XXXSP.

Fig.1.6.2 Block diagram of M37471M2-XXXSP, M37471M4-XXXSP, and M37471M8-XXXSP

M37471M2-XXXFP block diagram



Note 1 : 8192 bytes for M37471M4–XXXFP, and 16384 bytes for M37471M8–XXXFP.
2 : 192 bytes for M37471M4–XXXFP, and 384 bytes for M37471M8–XXXFP.

Fig.1.6.3 Block diagram of M37471M2-XXXFP, M37471M4-XXXFP, and M37471M8-XXXFP

CHAPTER 2

FUNCTIONAL DESCRIPTION

2.1 Central processing unit (CPU)

The central processing unit (CPU) of the M37470 has six internal registers. Five of these registers (the accumulator (A), index register X (X), index register Y (Y), stack pointer (S), and processor status register (PS)) have an 8-bit configuration, but the program counter (PC) has a 16-bit configuration consisting of two 8-bit registers (PCH and PCL).
After a reset, the I flag of the processor status register is set to "1", the contents of address $FFFF_{16}$ are placed in the high-order 8 bits of the program counter, and the contents of address $FFFE_{16}$ are placed in the low-order 8 bits of the program counter. The contents of the rest of the PS and the other registers are undefined, so initialization may be necessary for some programs.
The register structure of the M37470 is shown in Figure 2.1.1.

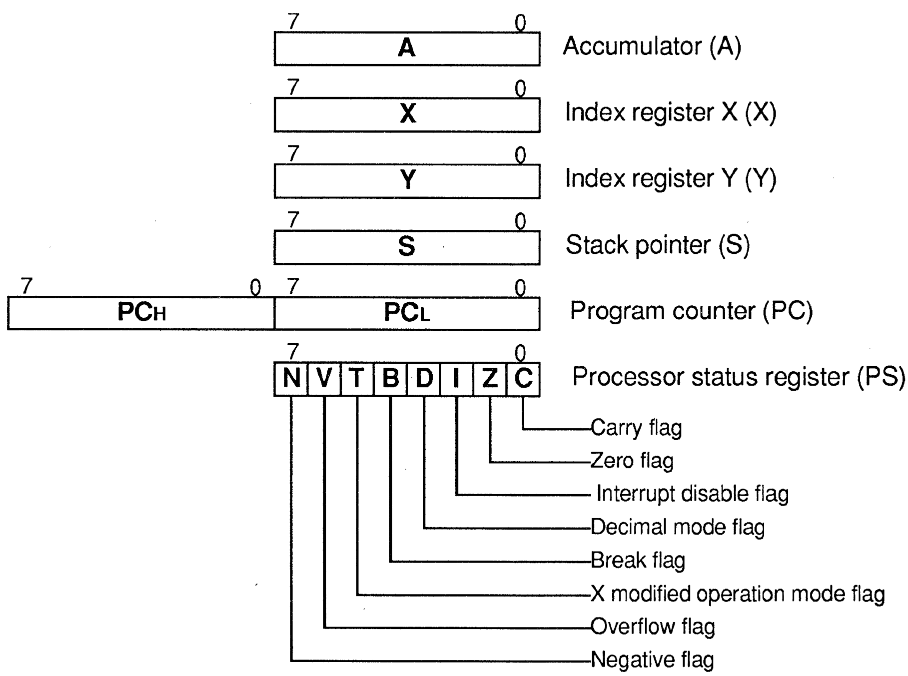


Fig.2.1.1 Register structure

2.1.1 Accumulator (A)

The accumulator is the central register of the microcomputer, and it has an 8-bit configuration. This is the most frequently used general-purpose register—it is used for arithmetic operations, data transfer, temporary storage, and condition judgements.

2.1.2 Index register X (X), index register Y (Y)

Both index register X and index register Y have 8-bit configurations. In addressing modes that use these index registers, the contents of these registers are added to the contents of the specified address to give the actual address to be used. These addressing modes are useful for referencing subroutine tables and memory tables.
These index registers also have increment, decrement, comparison, and data transfer functions, so they can be used as accumulators.
When the T flag of the processor status register (described below) is "1", the contents of index register X are used as they are as an operand address.

2.1.3 Stack pointer (S)

The stack pointer is an 8-bit register used during subroutine calls and interrupts.
When processing branches from the currently executing routine to a subroutine or interrupt processing routine, the return address must be saved. This return address is usually stored in the internal RAM, in an area called the stack area. The contents of the stack pointer indicate the storage location of stack data in the stack area. The operations of pushing register contents onto the stack and popping them from the stack are shown in Figure 2.1.2. These operations are performed automatically when an interrupt is received or a subroutine is called.
The contents of registers other than the program counter and processor status register are not automatically pushed onto the stack, and at subroutine call only the contents of the program counter are pushed. If pushing is necessary, specify it in the program: use the PHA and PLA instructions to push and pop the contents of the accumulator, and the PHP and PLP instructions to push and pop the contents of the processor status register.
RAM in the zero page is usually used as the stack area for this storage, but 1 page (addresses 0100_{16} to $01FF_{16}$) can also be used as a stack area by setting bit 2 of the CPU mode register (address $00FB_{16}$) to "1" (M37470M8 and M37471M8 only). The M37470M2, M37470M4, M37471M2, and M37471M4 do not have 1 page in RAM, so always set this bit to "0" in these microcomputers.

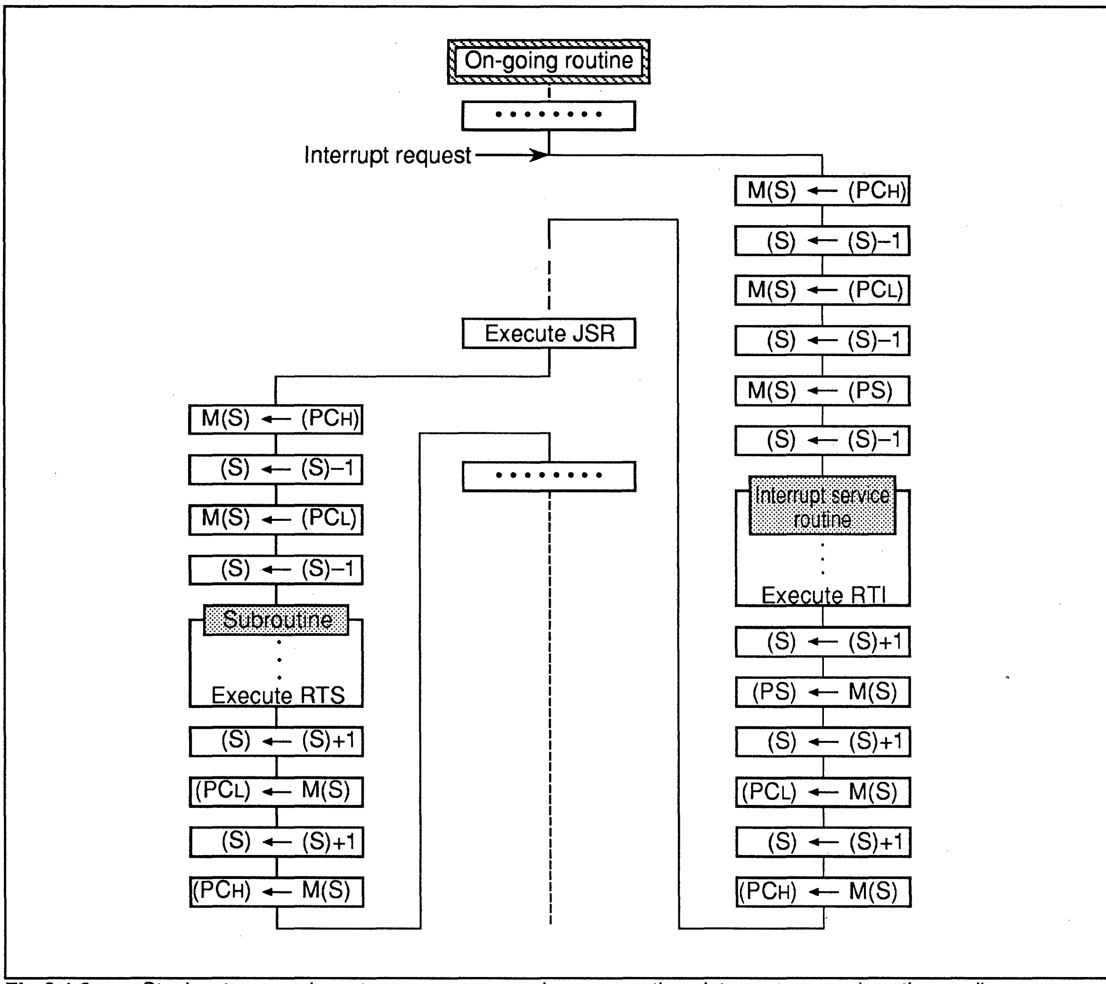


Fig.2.1.2 Stack store and restore sequence when executing interrupt or subroutine calls

2.1.4 Program counter (PC)

The program counter is a 16-bit counter consisting of two 8-bit registers: PCH and PCL. It contains the address of the next instruction to be executed.

2.1.5 Processor status register (PS)

The processor status register is an 8-bit register consisting of various flags such as those holding the status after an arithmetic operation.
After reset, the I flag is set to "1", but all other flags are undefined. Since the T and D flags directly affect arithmetic operations, always initialize them before such an operation.
The bits of the processor status register are described below.

(1) Carry flag (C)
The C flag stores a carry or borrow sent from the arithmetic and logic unit after an arithmetic operation. It is also changed by a shift or rotate instruction.
The C flag can be set by the SEC instruction and cleared by the CLC instruction.

(2) Zero flag (Z)
The Z flag is set if the result of an arithmetic operation or a data transfer is "0", or cleared if the result is anything other than "0".
The Z flag is not valid in decimal mode.

(3) Interrupt disable flag (I)
The I flag disables all interrupts except for the interrupt generated by the BRK instruction. Interrupts are disabled when the I flag is "1". It is automatically set to "1" when an interrupt is received, preventing multiple interrupts.
The I flag can be set by the SEI instruction and cleared by the CLI instruction.

(4) Decimal mode flag (D)
The D flag determines whether additions and subtractions are performed in binary or decimal notation. Ordinary binary arithmetic is performed when the D flag is "0"; decimal arithmetic with two digits per word is performed when it is "1". Decimal correction is automatic in decimal mode, but only the ADC and SBC instructions can be used for decimal arithmetic.
The D flag can be set by the SED instruction and cleared by the CLD instruction. Once the D flag has been set, it is valid until it is cleared by the CLD instruction or other cause.
Since the D flag directly affects calculations, always initialize it after a reset.

(5) Break flag (B)
The BRK instruction can be used during program debugging to give the same effect as an interrupt. The B flag determines whether an interrupt was generated by executing the BRK instruction. It is set to "1" if an interrupt was generated by the BRK instruction; or cleared to "0" and pushed onto the stack if any other interrupt was generated.

(6) Index X mode flag (T)
When the T flag is "0", arithmetic operations are performed between accumulator and memory; when it is "1", direct arithmetic operations and direct data transfers are enabled between memory and memory, memory and I/O, and I/O and I/O, without using the accumulator. In this case, the result of an arithmetic operation performed on data in memory location 1 and memory location 2 is stored in memory location 1. The address of memory location 1 is specified by the index register X, and that of memory location 2 is specified by the normal addressing mode.
The T flag can be set by the SET instruction and cleared by the CLT instruction.
Since the T flag directly affects calculations, always initialize it after a reset.

(7) Overflow flag (V)
The V flag is valid during the addition or subtraction of binary, one-word signed data. It is set if the result is outside the range of +127 to −128. It is also used after the BIT instruction is executed, to contain the value in bit 6 of the memory location operated on by the BIT instruction.
The V flag can be cleared by the CLV instruction, but it cannot be set by an instruction.
The V flag is not valid in decimal mode.

(8) Negative flag (N)
The N flag is set if the result of an arithmetic operation or data transfer is negative (bit 7 is "1"). It is also used after the BIT instruction is executed, to contain the value in bit 7 of the memory location operated on by the BIT instruction.
The N flag cannot be set or cleared directly by any instruction.
The N flag is not valid in decimal mode.

2.2 Access area

In the M37470, all the ROM, RAM, I/O functions, and control registers are located in the same memory area. This means that there is no need for programs to distinguish between memory and I/O operations—the same instructions can both transfer data and operate on data.
The program counter of the M37470 has a 16-bit configuration, and 64K bytes of memory area (from addresses 0000_{16} to $FFFF_{16}$) can be accessed. Of this 64K-byte memory area, the first 256 bytes are called the zero page area which holds frequently used memory functions such as internal RAM, I/O ports, and timers. The last 256 bytes of the 64K-byte memory area are called the special page area. The zero page and special page areas can be accessed in 2-byte units by using special addressing modes.

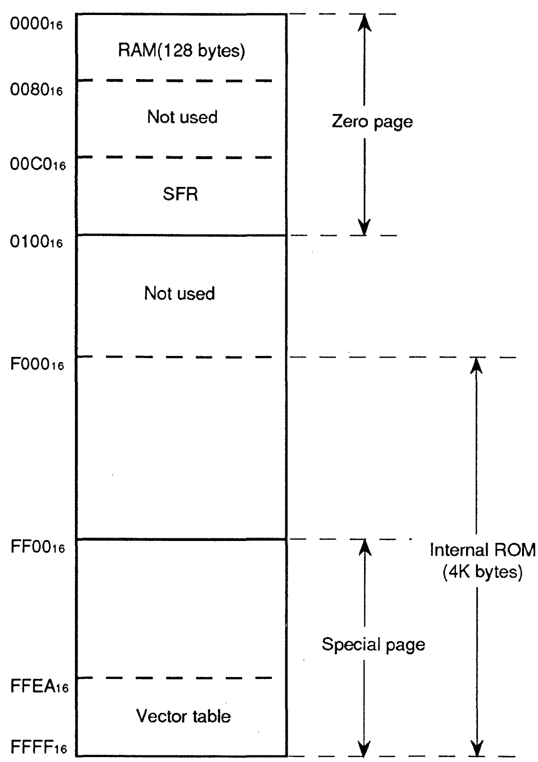


Fig.2.2.1 Access area (M37470M2-XXXSP)

2.2.1 Zero page (addresses 0000_{16} to $00FF_{16}$)
The 256 bytes from address 0000_{16} to address $00FF_{16}$ are called the zero page area. The internal RAM, access flags, and the special function register (SFR) are allocated to this area.
Use the zero page addressing mode shown in Figure 2.2.2 to specify memory and registers in the zero page area. This dedicated zero page addressing mode is particularly useful because it enables access to this area with even shorter instruction cycles.

2.2.2 Special page (addresses $FF00_{16}$ to $FFFF_{16}$)
The 256 bytes from address $FF00_{16}$ to address $FFFF_{16}$ are called the special page area.
Use the special page addressing mode shown in Figure 2.2.2 to specify memory in the special page area. This dedicated special page addressing mode is particularly useful because it enables access to this area with even shorter instruction cycles.
Frequently used subroutines are normally stored in this area.

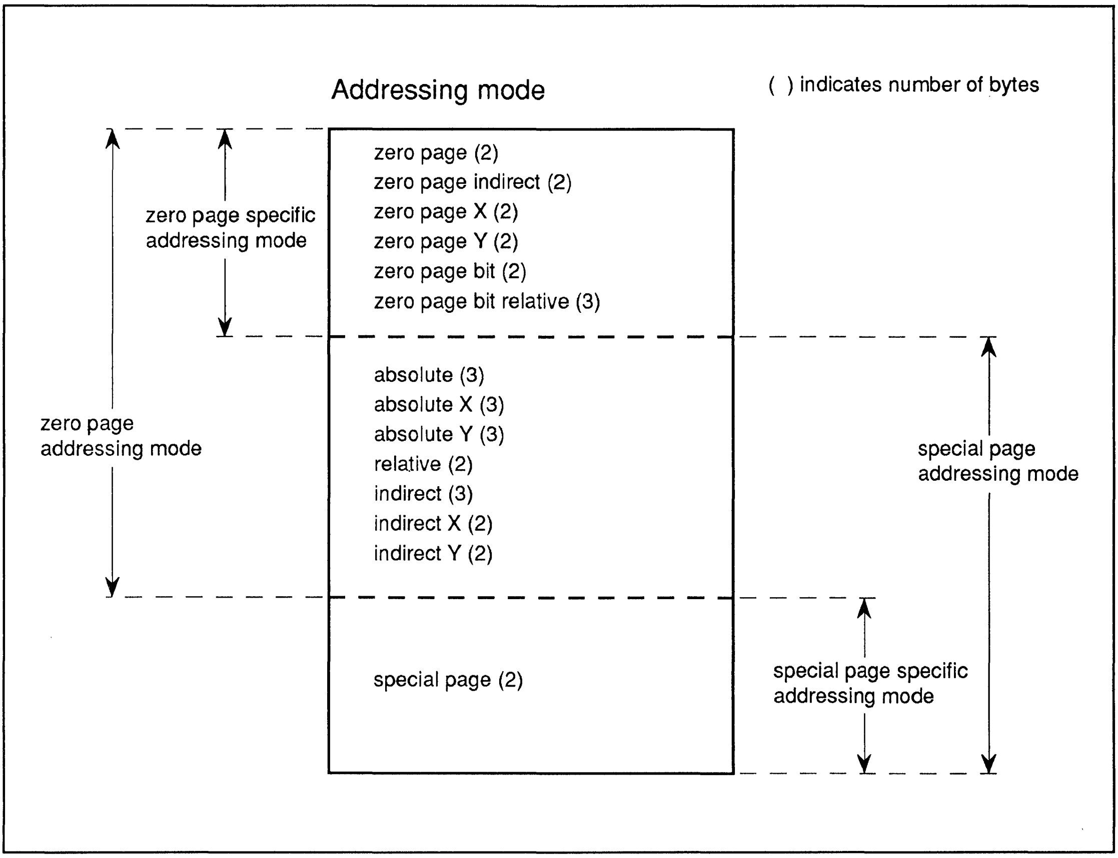


Fig.2.2.2 Zero page and special page addressing mode

2.3 Memory map

The memory map of the 7470 series is shown in Figures 2.3.1 to 2.3.3. Memory, I/O, and other functions allocated to the address spaces are described below.

●RAM 0000_{16} to $007F_{16}$: M37470M2, and M37471M2
0000_{16} to $00BF_{16}$: M37470M4, and M37471M4
0000_{16} to $00BF_{16}$
and
0100_{16} to $01BF_{16}$: M37470M8, and M37471M8

In the M37470M2, and M37471M2, static RAM with a capacity of 128 × 8 bits is allocated to addresses 0000_{16} to $007F_{16}$.
In the M37470M4, and M37471M4, static RAM with a capacity of 192 × 8 bits is allocated to addresses 0000_{16} to $00BF_{16}$.
In the M37470M8, and M37471M8, static RAM with a capacity of 384 × 8 bits is allocated to addresses 0000_{16} to $00BF_{16}$ and 0100_{16} to $01BF_{16}$.
Internal RAM is used for data storage as well as a stack area for subroutine call and interrupt generation. Therefore, when using RAM as a stack area, be careful that subroutine nesting and interrupt levels do not become too complex, to ensure that data stored in that RAM area is not destroyed.

●SFR
The area from address $00C0_{16}$ to address $00FF_{16}$ is allocated to the special function register (SFR), with the memory map shown in Figure 2.3.4. The SFR contains registers relating to functions such as I/O ports, timers, serial I/O, A-D converter, and interrupts.

●ROM $F000_{16}$ to $FFFF_{16}$: M37470M2, and M37471M2
$E000_{16}$ to $FFFF_{16}$: M37470M4, and M37471M4
$C000_{16}$ to $FFFF_{16}$: M37470M8, and M37471M8

In the M37470M2, and M37471M2, mask ROM with a capacity of 4K × 8 bits is allocated to addresses $F000_{16}$ to $FFFF_{16}$.
In the M37470M4, and M37471M4, mask ROM with a capacity of 8K × 8 bits is allocated to addresses $E000_{16}$ to $FFFF_{16}$.
In the M37470M8, and M37471M8, mask ROM with a capacity of 16K × 8 bits is allocated to addresses $C000_{16}$ to $FFFF_{16}$.
Addresses $FFEA_{16}$ to $FFFF_{16}$ in internal ROM are allocated as a vector area for storing jump destination addresses used at reset or when an interrupt is generated. A memory map of this vector area is shown in Figure 2.3.5.

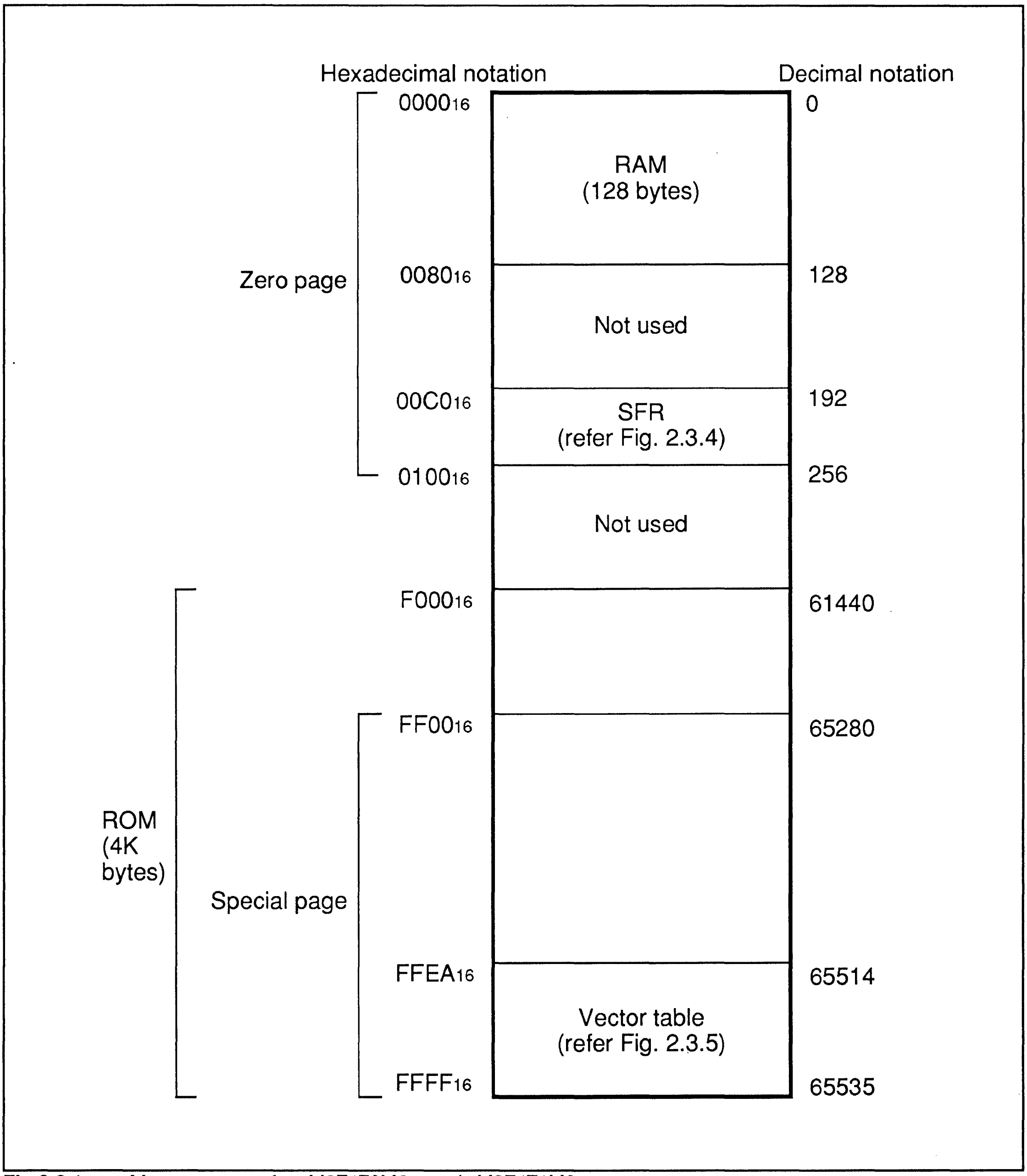


Fig.2.3.1 Memory map for M37470M2, and M37471M2

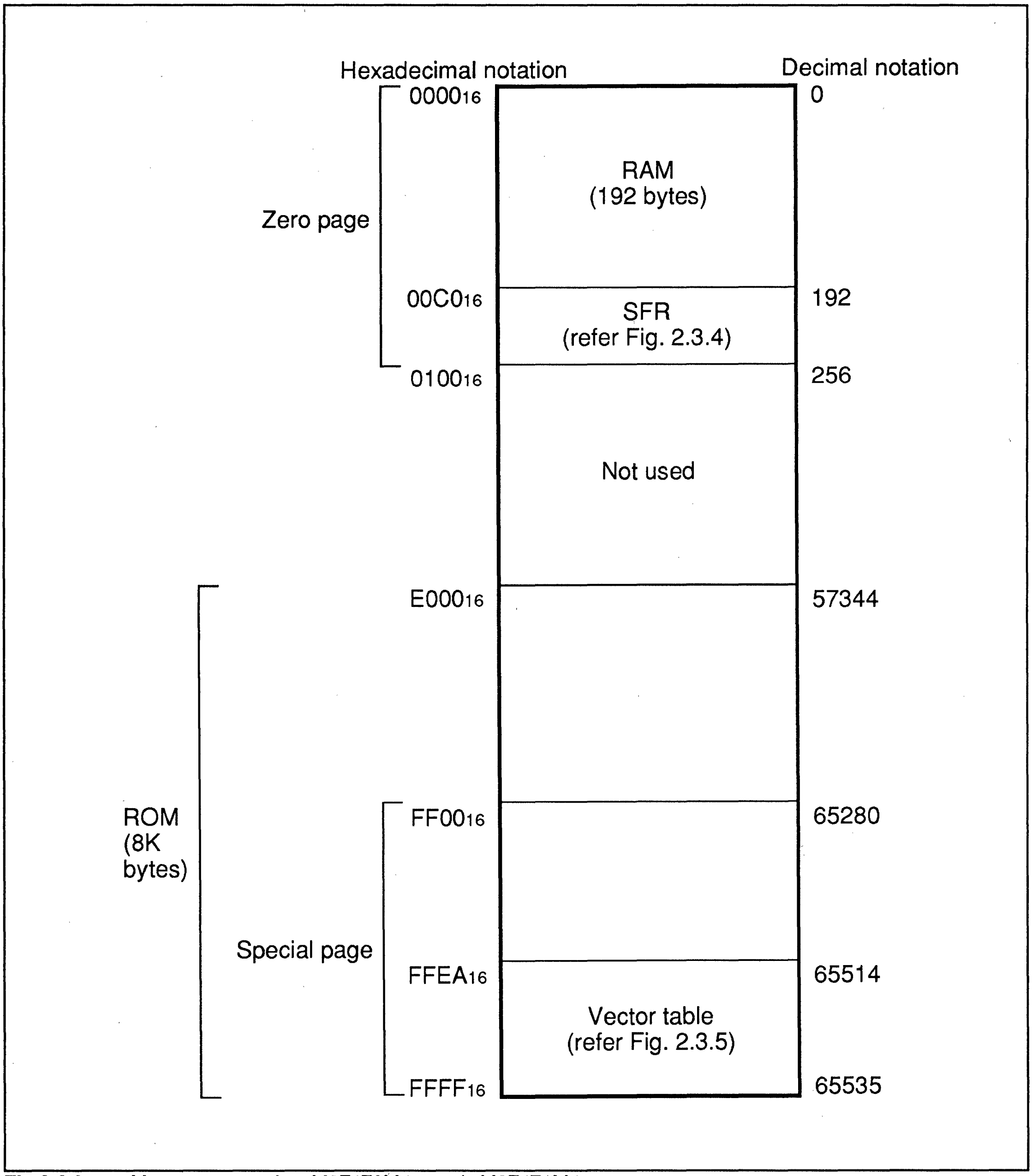


Fig.2.3.2 Memory map for M37470M4, and M37471M4

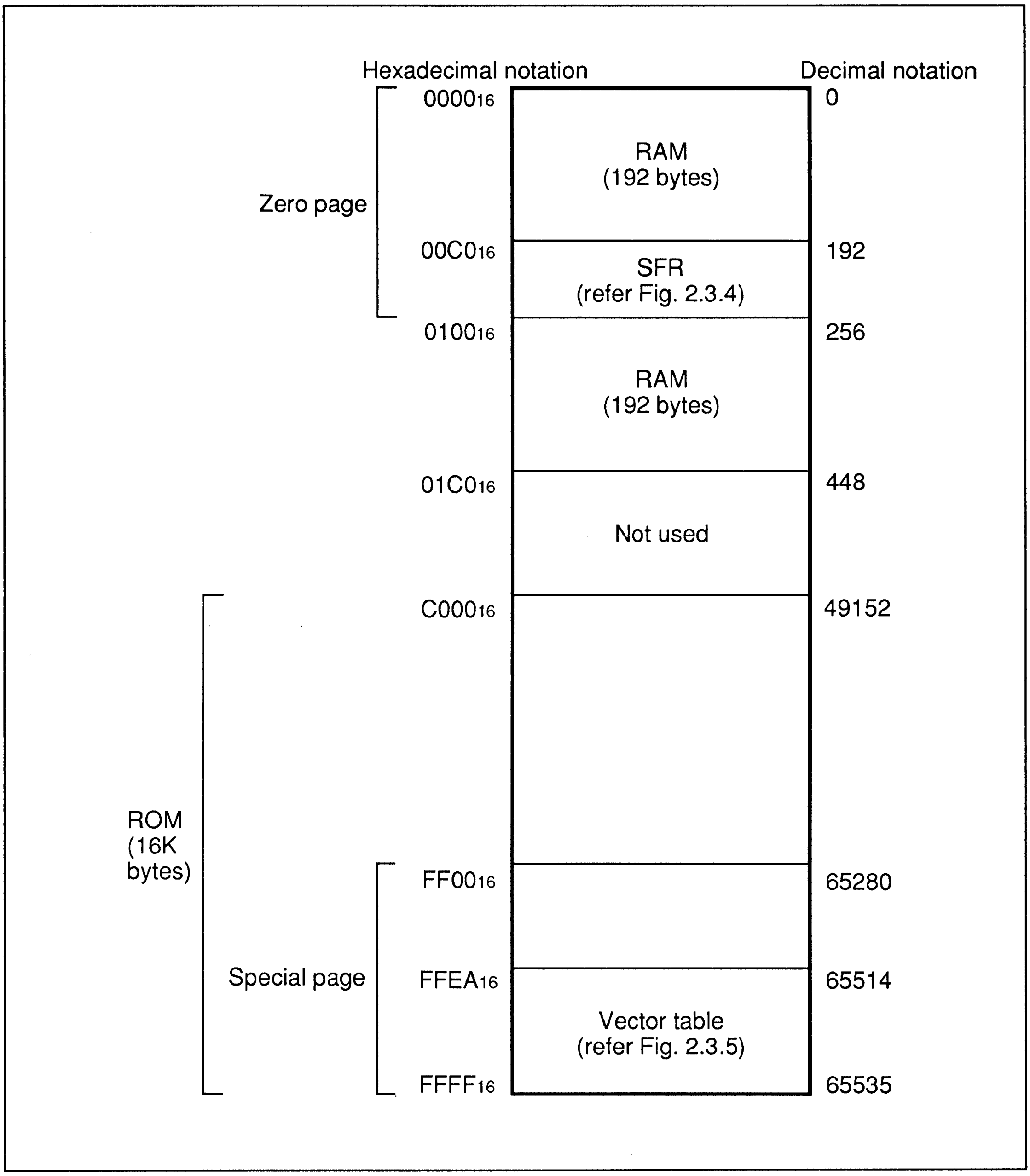


Fig.2.3.3 Memory map for M37470M8, and M37471M8

Address	Register	Address	Register
$00C0_{16}$	Port P0	$00E0_{16}$	
$00C1_{16}$	Port P0 direction register	$00E1_{16}$	
$00C2_{16}$	Port P1	$00E2_{16}$	
$00C3_{16}$	Port P1 direction register	$00E3_{16}$	
$00C4_{16}$	Port P2	$00E4_{16}$	
$00C5_{16}$	Port P2 direction register	$00E5_{16}$	
$00C6_{16}$	Port P3	$00E6_{16}$	
$00C7_{16}$		$00E7_{16}$	
$00C8_{16}$	Port P4	$00E8_{16}$	
$00C9_{16}$	Port P4 direction register	$00E9_{16}$	
$00CA_{16}$	Port P5 (Note 1)	$00EA_{16}$	
$00CB_{16}$		$00EB_{16}$	
$00CC_{16}$		$00EC_{16}$	
$00CD_{16}$		$00ED_{16}$	
$00CE_{16}$		$00EE_{16}$	
$00CF_{16}$		$00EF_{16}$	
$00D0_{16}$	Port P0 pull-up control register	$00F0_{16}$	Timer 1
$00D1_{16}$	Ports P1 to P5 pull-up control register	$00F1_{16}$	Timer 2
$00D2_{16}$		$00F2_{16}$	Timer 3
$00D3_{16}$		$00F3_{16}$	Timer 4
$00D4_{16}$	Edge polarity selection register	$00F4_{16}$	
$00D5_{16}$		$00F5_{16}$	
$00D6_{16}$	Input latch register	$00F6_{16}$	
$00D7_{16}$		$00F7_{16}$	Timer FF register
$00D8_{16}$		$00F8_{16}$	Timer 12 mode register
$00D9_{16}$	A-D control register	$00F9_{16}$	Timer 34 mode register
$00DA_{16}$	A-D conversion register	$00FA_{16}$	Timer mode register 2
$00DB_{16}$		$00FB_{16}$	CPU mode register
$00DC_{16}$	Serial I/O mode register	$00FC_{16}$	Interrupt request register 1
$00DD_{16}$	Serial I/O register	$00FD_{16}$	Interrupt request register 2
$00DE_{16}$	Serial I/O counter / Byte counter	$00FE_{16}$	Interrupt control register 1
$00DF_{16}$		$00FF_{16}$	Interrupt control register 2

Note 1 : This address is not used in M37470M2, M37470M4, and M37470M8.
2 : Shaded area is not used.

Fig.2.3.4 SFR memory map

$FFEA_{16}$	BRK instruction interrupt	L
$FFEB_{16}$		H
$FFEC_{16}$	A - D conversion interrupt	L
$FFED_{16}$		H
$FFEE_{16}$	Serial I/O interrupt	L
$FFEF_{16}$		H
$FFF0_{16}$	Timer 4 interrupt	L
$FFF1_{16}$		H
$FFF2_{16}$	Timer 3 interrupt	L
$FFF3_{16}$		H
$FFF4_{16}$	Timer 2 interrupt	L
$FFF5_{16}$		H
$FFF6_{16}$	Timer 1 interrupt	L
$FFF7_{16}$		H
$FFF8_{16}$	$CNTR_0$/$CNTR_1$ interrupt	L
$FFF9_{16}$		H
$FFFA_{16}$	INT_1 / Key on wake up interrupt	L
$FFFB_{16}$		H
$FFFC_{16}$	INT_0 interrupt	L
$FFFD_{16}$		H
$FFFE_{16}$	Reset	L
$FFFF_{16}$		H

L means low-order 8-bit of jump distination address and H means high-order 8-bit of jump distination address.

Fig.2.3.5 Interrupt vector area memory map

2.4 Input/Output ports

Figure 2.4.1 shows I/O ports of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP. And I/O ports of M37471M2-XXXSP/FP, M37471M4-XXXSP/FP, and M37471M8-XXXSP/FP are shown in Figure 2.4.2.

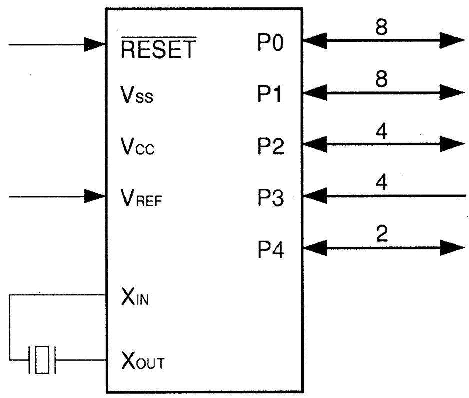


Fig.2.4.1 I/O ports of M37470M2, M37470M4, and M37470M8

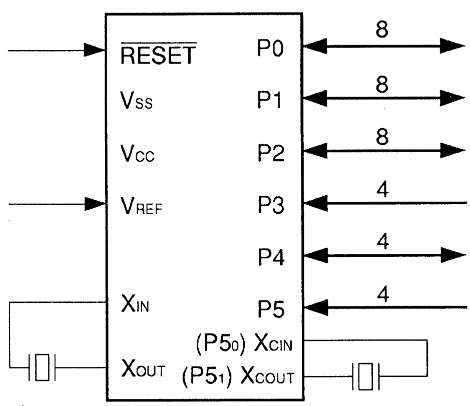


Fig.2.4.2 I/O ports of M37471M2, M37471M4, and M37471M8

2.4.1 Input/Output port

(1) Port P0

Port P0 is an 8-bit input/output port. Output is in CMOS output format. Port P0 is handled as memory at address $00C0_{16}$ in the zero page. It has a direction registers (address $00C1_{16}$) whose bits can be individually programmed to set each of the pins of port P0 to either input or output. If a bit of direction register is set by program to "1", the corresponding pin is set to be an output pin; if the bit of direction register is set to "0"; the pin is set to be an input pin. Data written to a pin that has been programmed as an output pin is written to a port latch, then it is output without change to the output pin. When data is read from a pin that has been programmed as an output pin, the contents of the output pin itself are not read; the contents of the port latch are read instead. This ensures that the previously output value is read correctly, even if some cause such as an external load has driven an output "H" voltage down or a "L" voltage up. A pin programmed to be an input pin is floating, and its status can always be read. When data is written to such a pin, it is written only to the port latch, so the pin itself stays floating.

At reset, all the pins of port P0 are set to input. When the port is set to input, individual pins can be connected to pull-up transistors (see Figure 2.4.3).

If a key matrix is created for input to port P0, the M37470 can be returned from wait or stop mode to normal operating mode by simply pressing a key. This generates an interrupt by applying an "L" level voltage to port P0.

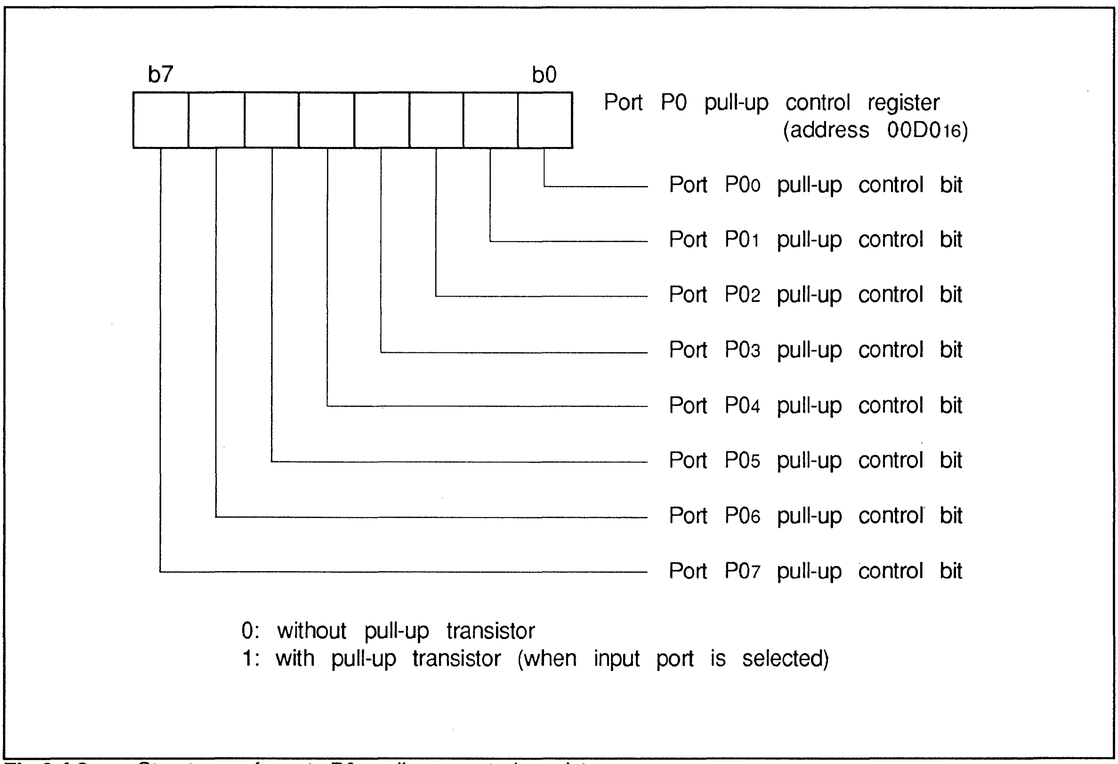


Fig.2.4.3 Structure of port P0 pull-up control register

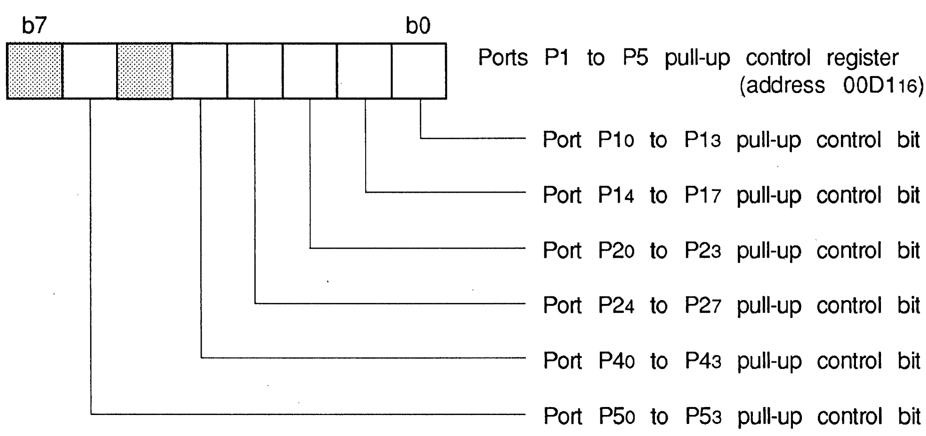
(2) Port P1
Port P1 has the same functions as port P0. Pins $P1_2$ to $P1_7$ have double functions—their functions can be selected by program. When port P1 is selected to be an input port, its pins can be connected in groups of four to pull-up transistors (see Figure 2.4.4).

(3) Port P2
Port P2 is an 8-bit I/O port with the same functions as port P0. In the M37470M2, M37470M4, and M37470M8, this port is $P2_0$ to $P2_3$, a 4-bit I/O port. This port can also be used as an analog voltage input pin. When port P2 is selected to be an input port, its pins can be connected in groups of four to pull-up transistors (see Figure 2.4.4).

(4) Port P3
Port P3 is a 4-bit input port. Its pins can also be used as external interrupt input pins and timer input pins.

(5) Port P4
Port P4 is an 4-bit I/O port with the same functions as port P0. In the M37470M2, M37470M4, and M37470M8, this port is $P4_0$ and $P4_1$, a 2-bit I/O port. When port P4 is selected to be an input port, its pins can be connected in groups of four to pull-up transistors (see Figure 2.4.4).

(6) Port P5
Port P5 is a 4-bit input port that can be connected as a group of four pins to a pull-up transistor (see Figure 2.4.4). Pins $P5_0$ and $P5_1$ can also be used as I/O pins for the clock-function clock generating circuit. Note that the M37470M2, M37470M4, and M37470M8 do not have this port.



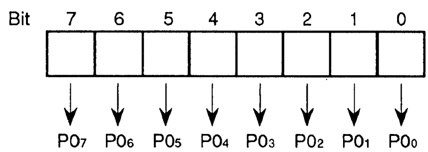
0: without pull-up transistor
1: with pull-up transistor (when input port is selected)

Note: M37470M2, M37470M4, and M37470M8 do not have bit 3 and bit 6.

Fig.2.4.4 Structure of ports P1 to P5 pull-up control register

Switch the programmable I/O ports between input and output as shown below.

The direction register corresponding to each port is located in the SFR area assigned to addresses $00C0_{16}$ to $00FF_{16}$. Each bit of this direction register determines the corresponding port direction. Bit and pin correspond in the following manner:

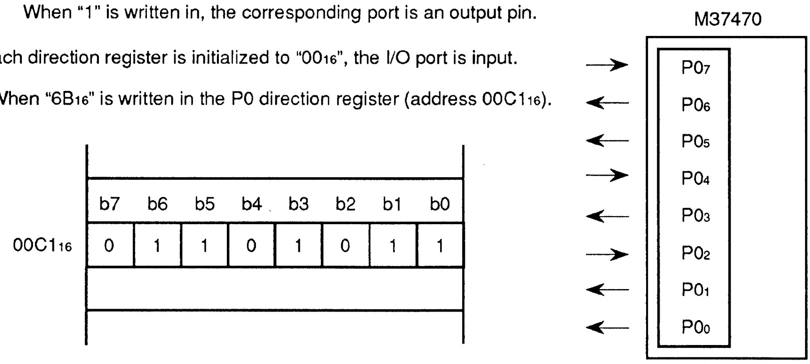


On the corresponding bit of the pin:

When "0" is written in, the corresponding port is an input pin.
When "1" is written in, the corresponding port is an output pin.

At reset, each direction register is initialized to "00_{16}", the I/O port is input.

Example: When "$6B_{16}$" is written in the P0 direction register (address $00C1_{16}$).



2.4.2 Pin description
The functions of the other I/O pins of the M37470 are described below.

(1) X_{IN} and X_{OUT} pins
The X_{IN} and X_{OUT} pins are clock I/O pins. The M37470 has a built-in clock generating circuit whose oscillation frequency can be set by a ceramic resonator or a crystal oscillator. A feedback resistor is built in between the X_{IN} and X_{OUT} pins.
When using an external clock, connect the clock oscillation source to the X_{IN} pin, and leave the X_{OUT} pin open.

(2) $\overline{RESET}$ pin
The system is reset when the $\overline{RESET}$ pin is held at "L" for at least 2µs.

(3) V_{SS} and V_{CC} pins
The V_{SS} and V_{CC} pins supply power to the chip.

(4) INT_0 pin ($P3_0/INT_0$)
The interrupt input pin INT_0 can also be used as port $P3_0$. When either a rising edge or a falling edge is input to this pin, the INT_0 interrupt request bit (bit 0 of address $00FD_{16}$) is set to "1".

(5) INT_1 pin ($P3_1/INT_1$)
The interrupt input pin INT_1 can also be used as port $P3_1$. When either a rising edge or a falling edge is input to this pin, the INT_1 interrupt request bit (bit 1 of address $00FD_{16}$) is set to "1".

(6) $CNTR_0$ pin ($P3_2/CNTR_0$)
The timer input pin $CNTR_0$ can also be used as port $P3_2$.

(7) $CNTR_1$ pin ($P3_3/CNTR_1$)
The timer input pin $CNTR_1$ can also be used as port $P3_3$.

(8) T_0 pin ($P1_2/T_0$)
The timer output pin T_0 can also be used as port $P1_2$.

(9) T_1 pin ($P1_3/T_1$)
The timer output pin T_1 can also be used as port $P1_3$.

(10) IN_0 to IN_7 pins ($P2_0/IN_0$ to $P2_7/IN_7$)
The analog input pins IN_0 to IN_7 can also be used as port $P2_0$ to $P2_7$. M37470M2, M37470M2A, M37470M4, and M37470M8 do not have IN_4–IN_7 pins.

(11) V_{REF} pin
The V_{REF} pin is a reference voltage input pin for the A-D converter.

(12) X_{CIN} and X_{COUT} pins ($P5_0/X_{CIN}$ and $P5_1/X_{COUT}$)
The X_{CIN} and X_{COUT} pins are clock I/O pins for the clock function, but they can also be used as ports $P5_0$ and $P5_1$. A feedback resistor is built in between the X_{CIN} and X_{COUT} pins, but it is disconnected when these pins are used as ordinary ports. The M37470M2, M37470M4, and M37470M8 do not have these pins.

(13) AV_{SS} pin
The AV_{SS} pin is a ground level input pin for A-D converter. Same voltage as V_{SS} is applied. This pin is for 56-pin QFP type only.

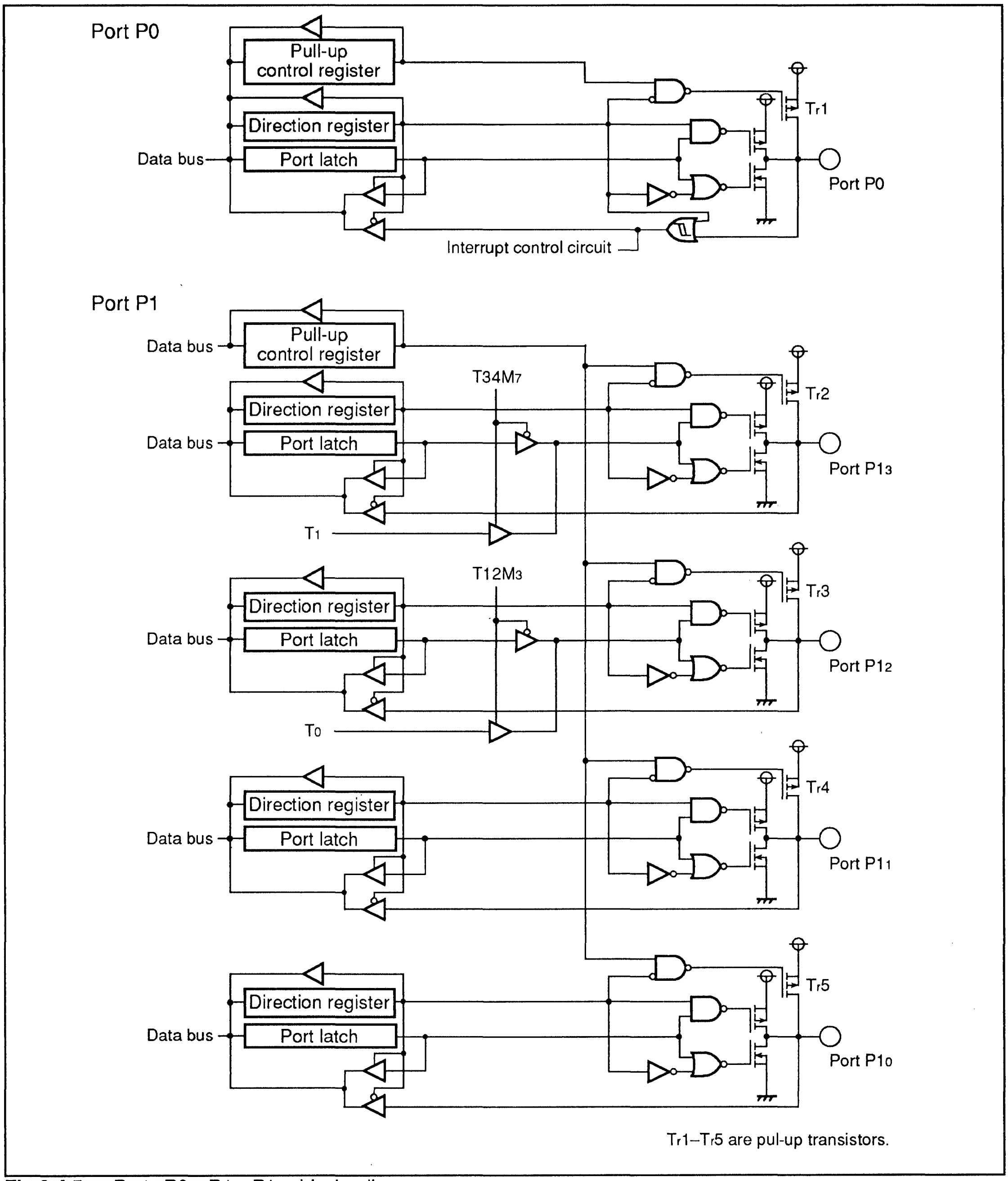


Fig.2.4.5 Port P0, $P1_0$–$P1_3$ block diagram

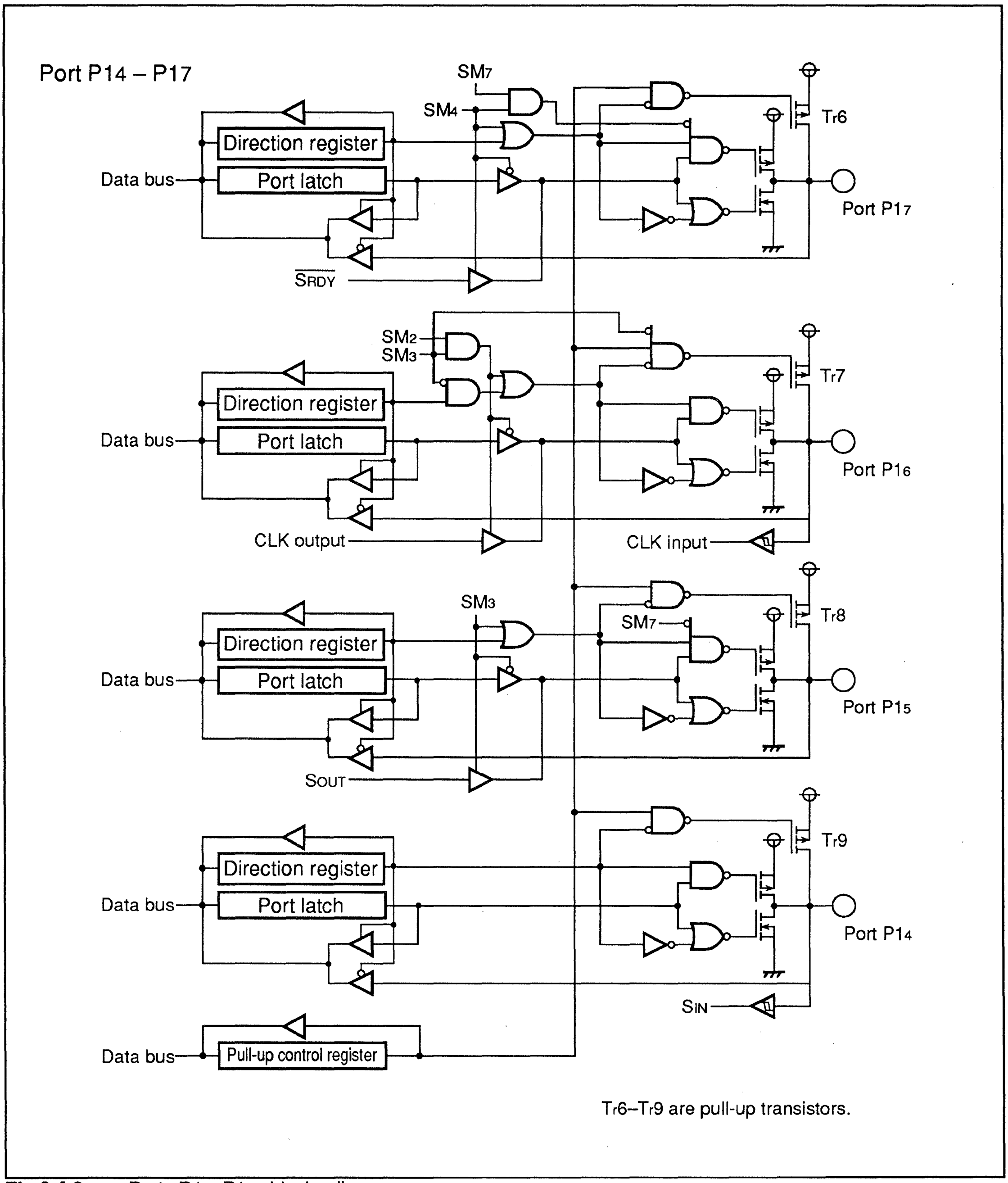


Fig.2.4.6 Port P14–P17 block diagram

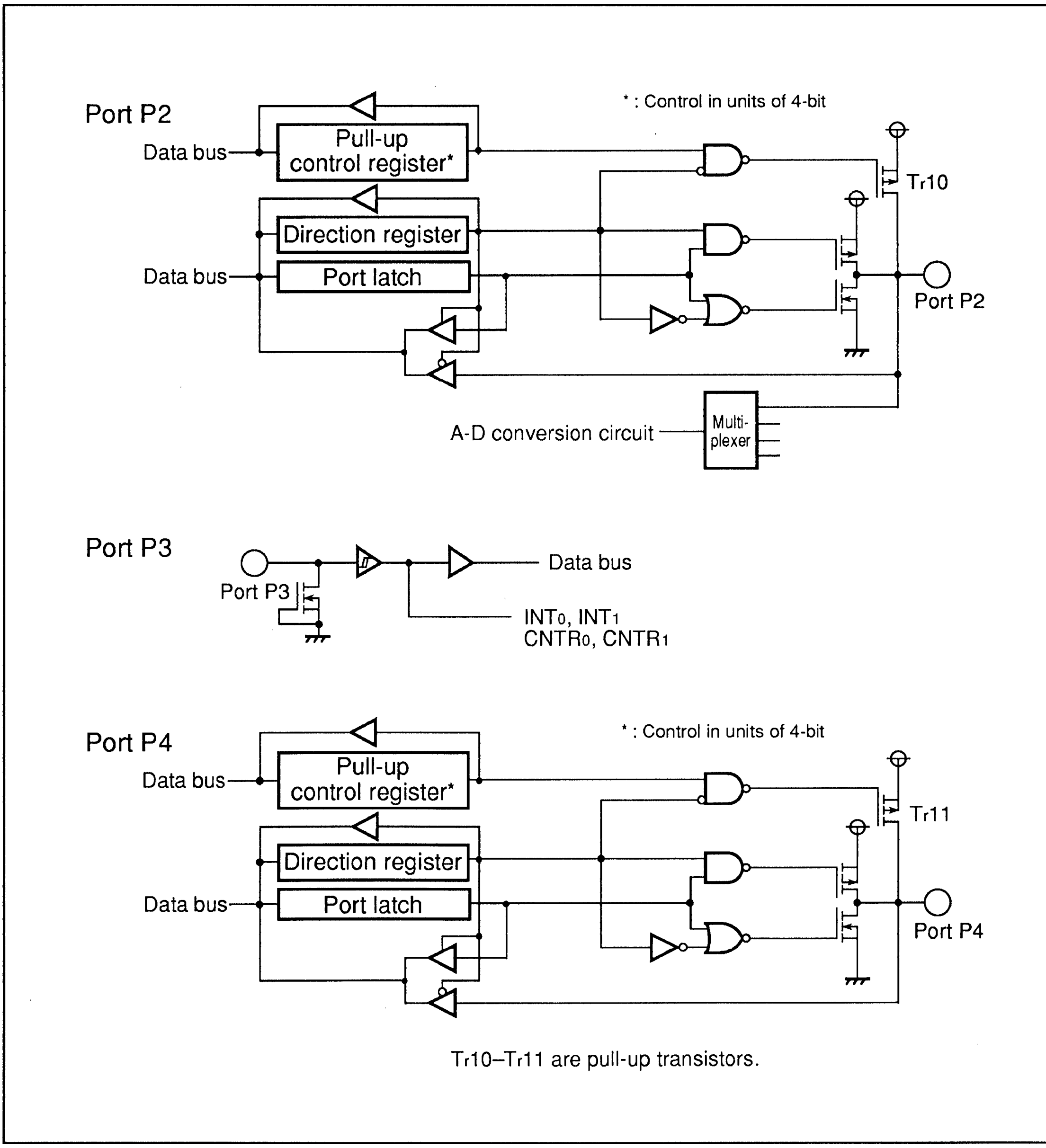


Fig.2.4.7 Port P2–P4 block diagram

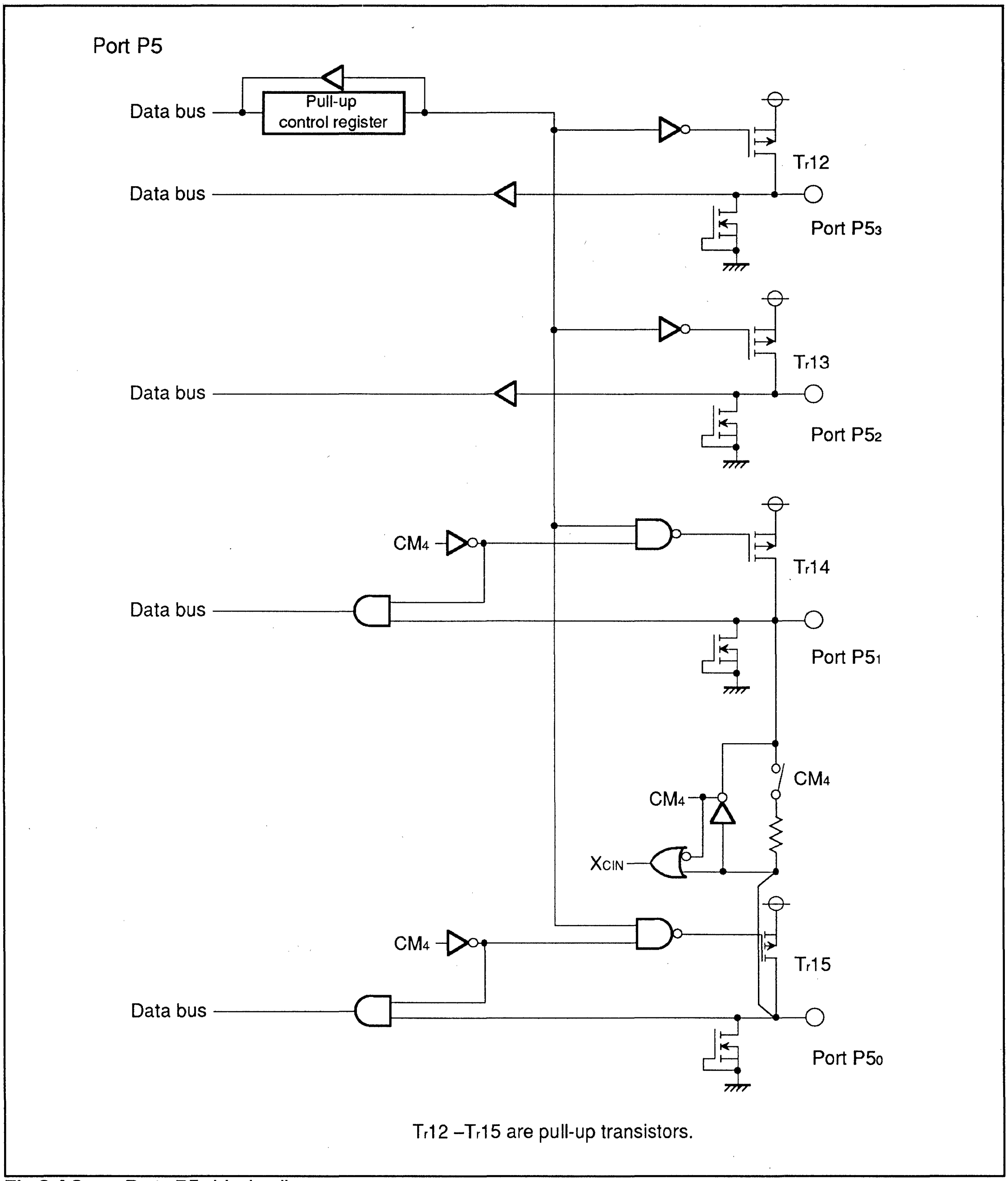


Fig.2.4.8 Port P5 block diagram

2.5 Interrupts

Interrupts are used in the following cases:

- When processing that is more important that the currently executing processing routine is requested.
- When processing must be executed at a specific timing.

The M37470 can be interrupted by 12 causes. These interrupts are vector interrupts with a fixed priority sequence. Interrupt causes, jump destination addresses, and interrupt priorities are listed in Table 2.5.1.

Table 2.5.1 Interrupt vector addresses and priority

Priority	Interrupt cases	Jump destination addresses		Remarks
		Higher	Lower	
1	Reset (note)	$FFFF_{16}$	$FFFE_{16}$	Non-maskable
2	INT_0 interrupt	$FFFD_{16}$	$FFFC_{16}$	Polarity programmable
3	INT_1/Key on wake up interrupt	$FFFB_{16}$	$FFFA_{16}$	Polarity programmable (INT_1)
4	$CNTR_0$/$CNTR_1$ interrupt	$FFF9_{16}$	$FFF8_{16}$	Polarity programmable
5	Timer 1 interrupt	$FFF7_{16}$	$FFF6_{16}$	
6	Timer 2 interrupt	$FFF5_{16}$	$FFF4_{16}$	
7	Timer 3 interrupt	$FFF3_{16}$	$FFF2_{16}$	
8	Timer 4 interrupt	$FFF1_{16}$	$FFF0_{16}$	
9	Serial I/O interrupt	$FFEF_{16}$	$FFEE_{16}$	
10	A-D conversion interrupt	$FFED_{16}$	$FFEC_{16}$	
11	BRK instruction interrupt	$FFEB_{16}$	$FFEA_{16}$	Non-maskable software interrupt

Note: Reset is included in this table because it operates in the same way as an interrupt.

These 12 interrupts have the priorities listed in Table 2.5.1 (but note that a reset input has a higher priority than all these interrupts). If two or more interrupts are requested in the same sampling period, the interrupt with the highest priority is the only one that is accepted. The priority sequence is determined by the hardware, but a variety of interrupt processing options can also be set by software, using the interrupt control flags (the interrupt enable bits and the interrupt disable flag).

2.5.1 Interrupt causes

The various interrupt causes are described below.

(1) INT_0, INT_1 and key on wake up interrupts
An interrupt request is generated when either a rising edge or a falling edge is detected in the level of the INT_0 pin or the INT_1 pin. The active edge that is detected in this way can be selected by setting bit 0 or bit 1 of the active edge selection register (address $00D4_{16}$).
The INT_0 and INT_1 pins can also be used as the $P3_0$ and $P3_1$ pins; the levels at ports $P3_0$ and $P3_1$ are always detected.
The active edge selection register is cleared to "00_{16}" by a reset, so requests for INT_0 and INT_1 interrupts are generated when falling edges are subsequently detected at those pins.
After a low power consumption mode has been set by the STP instruction or the WIT instruction, interrupts are key on wake up interrupts if bit 5 of the active edge selection register is "1", or INT_1 interrupts if it is "0". If key on wake up interrupts are validated, an interrupt request is generated whenever an "L" level voltage is applied to any of the pins of port P0 that have been set to input mode.
When a non low power consumption mode is set, both key on wake up and INT_1 interrupts are invalid if bit 5 of the active edge selection register is set to "1".

(2) $CNTR_0$ and $CNTR_1$ interrupts
An interrupt request is generated when either a rising edge or a falling edge is detected in the level of the $CNTR_0$ pin or the $CNTR_1$ pin. Use bit 4 of the active edge selection register to determine whether the $CNTR_0$ pin or the $CNTR_1$ pin is the interrupt input pin. The active edge can be selected for each pin by setting bits 2 and 3 of the active edge selection register.

(3) Timer 1, timer 2, timer 3, and timer 4 interrupts
An interrupt request is generated when the timer overflows.

(4) Serial I/O interrupt
An interrupt request is generated when serial I/O transmit or receive is completed.

(5) A-D conversion interrupt
An interrupt request is generated when A-D conversion is completed.

(6) BRK instruction interrupt
The BRK instruction interrupt has the lowest priority of software interrupts; it does not have a corresponding interrupt enable bit and the interrupt disable flag has no effect on it (it is non-maskable).

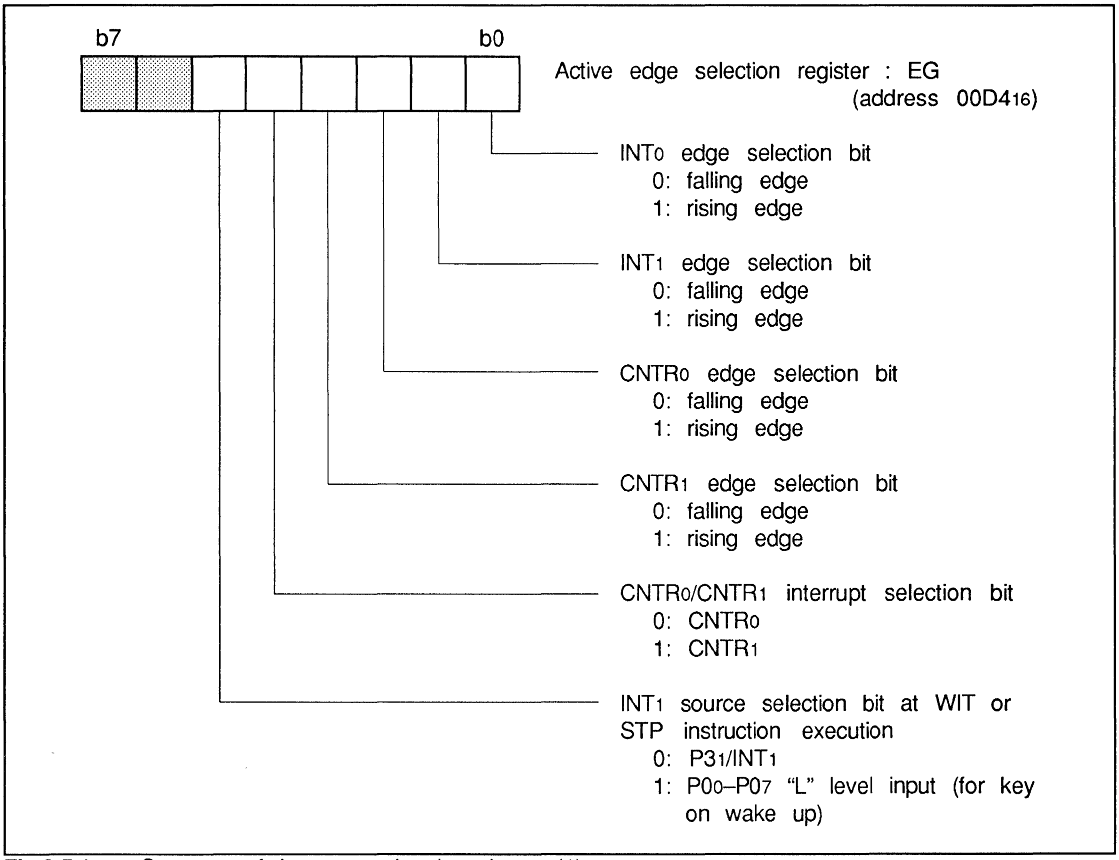


Fig.2.5.1 Structure of interrupt related registers (1)

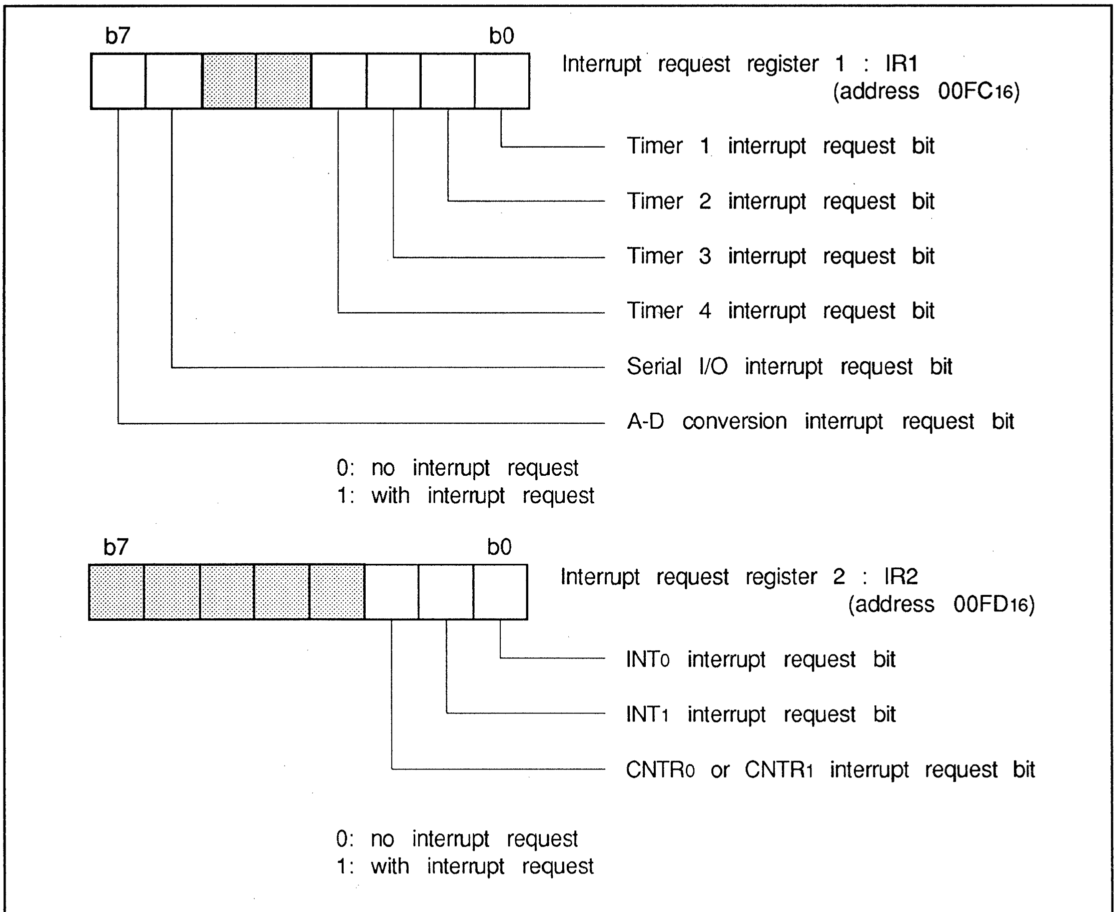


Fig.2.5.2 Structure of interrupt related registers (2)

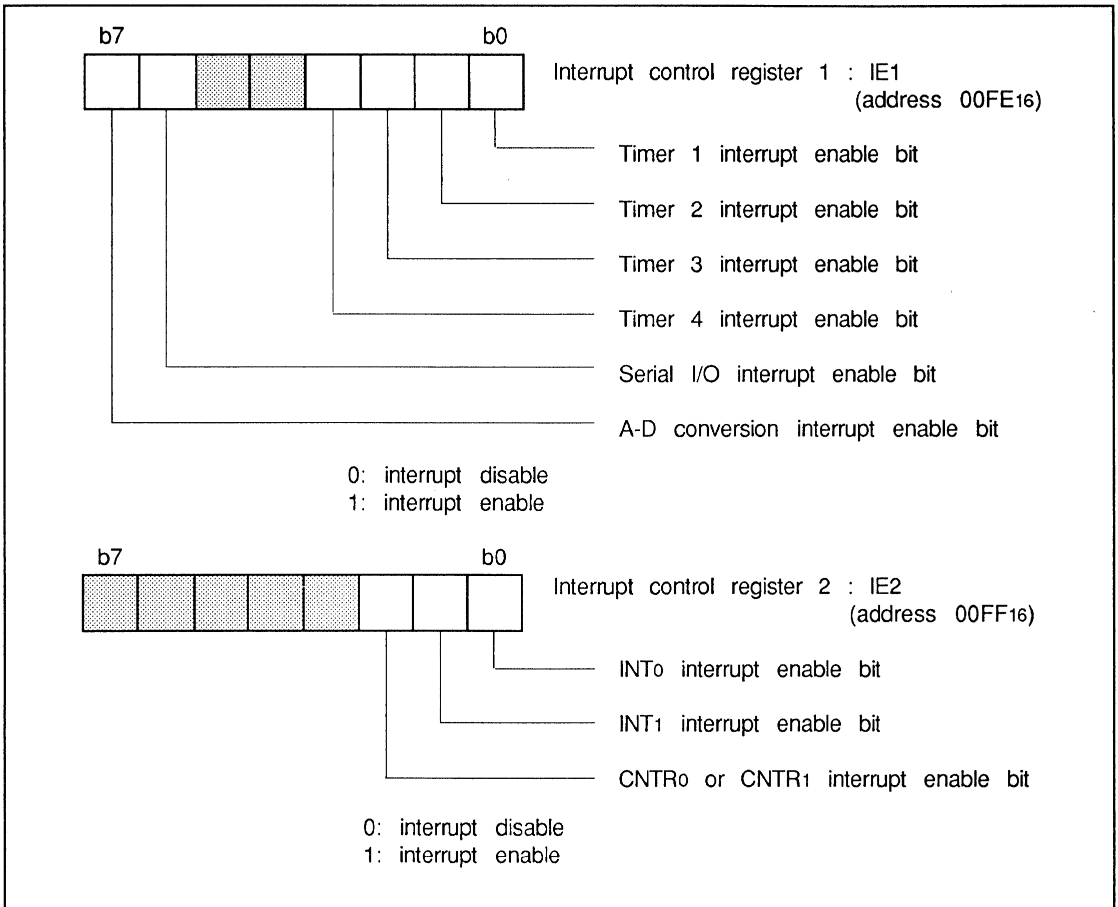


Fig.2.5.3 Structure of interrupt related registers (3)

For further details of the various interrupts, see the sections on the corresponding functions.

2.5.2 Interrupt control

Each of the M37470's interrupts is controlled by two bits and one flag: its interrupt request bit, its interrupt enable bit, and the interrupt disable flag, as shown in Figure 2.5.4, except for the software interrupt set by the BRK instruction. These control bits and control flag are described below.

(1) Interrupt request bits
When an interrupt request is generated, the corresponding request bit is set to "1". The interrupt request bit remains set until the interrupt is accepted; it is cleared at the same time that the interrupt is accepted. These bits can be cleared by a program, but they cannot be set.

(2) Interrupt enable bits
The interrupt enable bits control the acceptance of interrupts. When the bit corresponding to an interrupt is "0", the acceptance of that interrupt is disabled; when the bit is "1", the corresponding interrupt is enabled.

(3) Interrupt disable flag (I)
The I flag is allocated to bit 2 of the processor status register. This flag disables all interrupts except the BRK instruction interrupt. When the interrupt disable flag is set to "1", interrupts are disabled; when it is cleared to "0", interrupts are enabled. Use the SEI instruction to set the interrupt disable flag, and the CLI instruction to clear it.
Once an interrupt service routine has started, the I flag is automatically set to disable multiple interrupts. To enable multiple interrupts, specify the CLI instruction within the interrupt service routine to clear the I flag.

The above control bits and control flag are independent; they do not have any effect on each other. An interrupt is generated when the corresponding interrupt request and enable bits are "1" and the

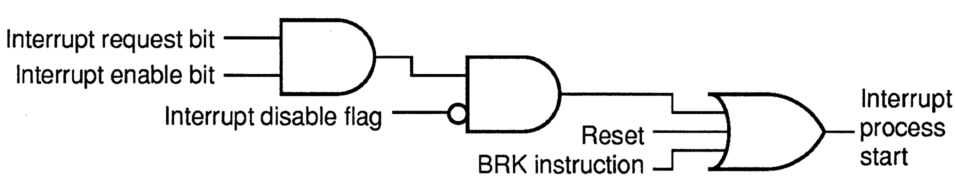


Fig.2.5.4 Interrupt control

interrupt disable flag is "0".

2.5.3 Processing at interrupt acceptance

When an interrupt is accepted, the currently executing processing is temporarily halted and the appropriate interrupt service routine is executed. After the interrupt service routine ends, the program flow must be such that the previous processing continues.

When the M37470 accepts an interrupt, it automatically pushes the contents of the program counter and the processor status register onto the stack. At the same time, it takes the contents of the vector corresponding to the accepted interrupt (the start address of the interrupt service routine) from the interrupt vector table and puts them into the program counter, then it executes the interrupt service routine.

When the interrupt service routine starts, the request bit corresponding to that interrupt is cleared to "0", and the interrupt disable flag becomes "1" to disable multiple interrupts. (To enable multiple interrupts, specify the CLI instruction within the interrupt service routine to clear the I flag.)

Before an interrupt service routine can be executed, a jump destination address must be set in the vector table to correspond to each interrupt. The jump destination addresses of all the interrupts are listed in Table 2.5.1.

Changes in the stack pointer and program counter when an interrupt is accepted are shown in Figure 2.5.5.

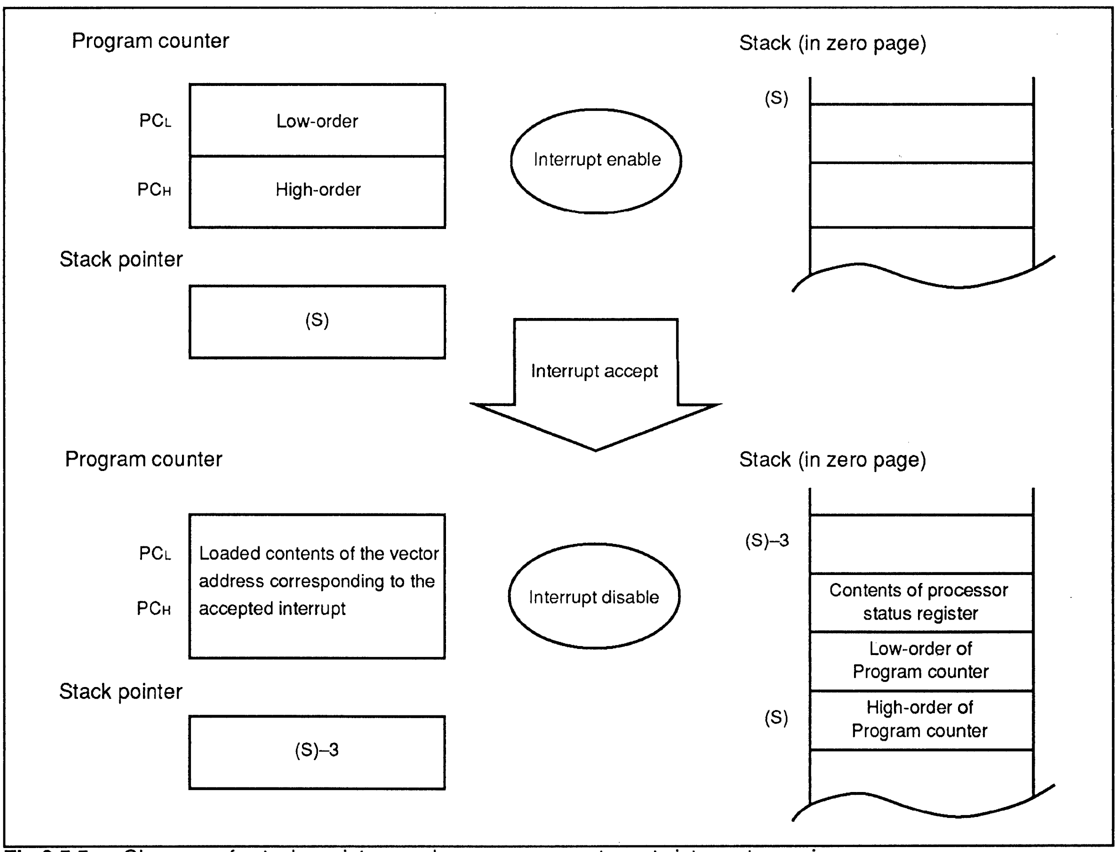
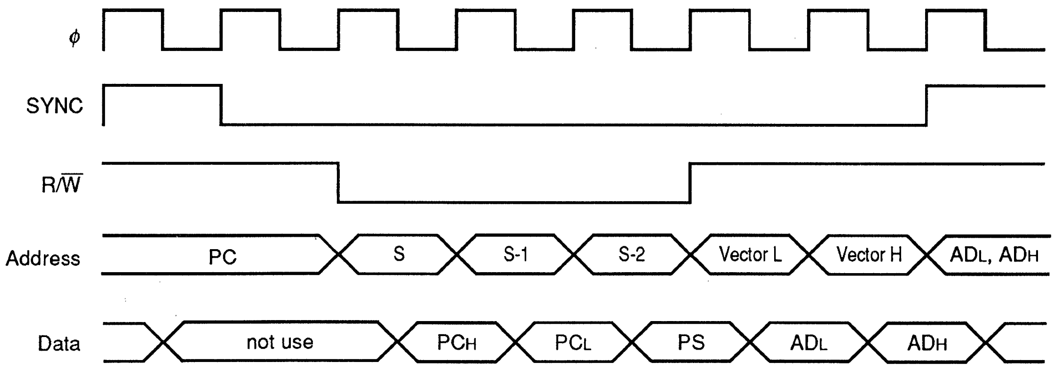


Fig.2.5.5 Change of stack pointer and program counter at interrupt receive

2.5.4 Timing after interrupt

Figure 2.5.6 shows the timing chart of stack save and interrupt routine start at the interrupt occurrence. Figure 2.5.7 shows the operation time before interrupt routine execution.



Note : Vector L and vector H are the vector address of each interrupt, and ADH, ADL are the contents of these vector address (jump addresses). See Table 2.5.1 for the vector address of each interrupt.

Fig.2.5.6 Timing chart after interrupt

The interrupt service routine will start after the completion of the instruction being executed when the request occurs. The two conditions that allow for the interrupt to be accepted is the interrupt disable flag must be "0" and the interrupt enable bit must be "1" (with the exception of the BRK instruction interrupt).

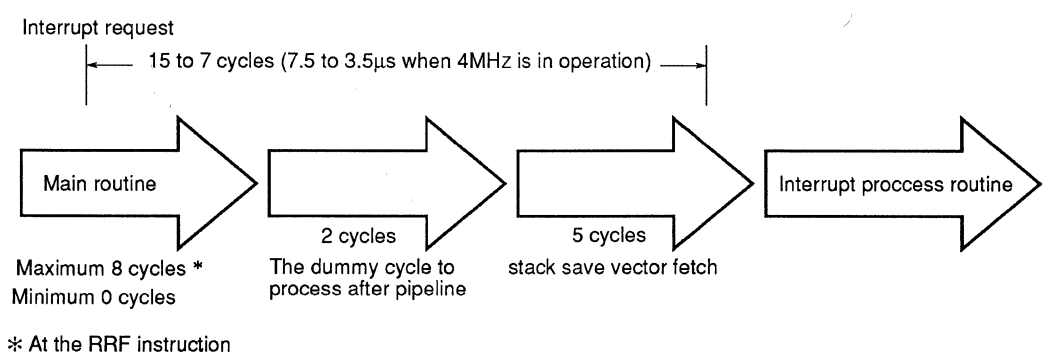


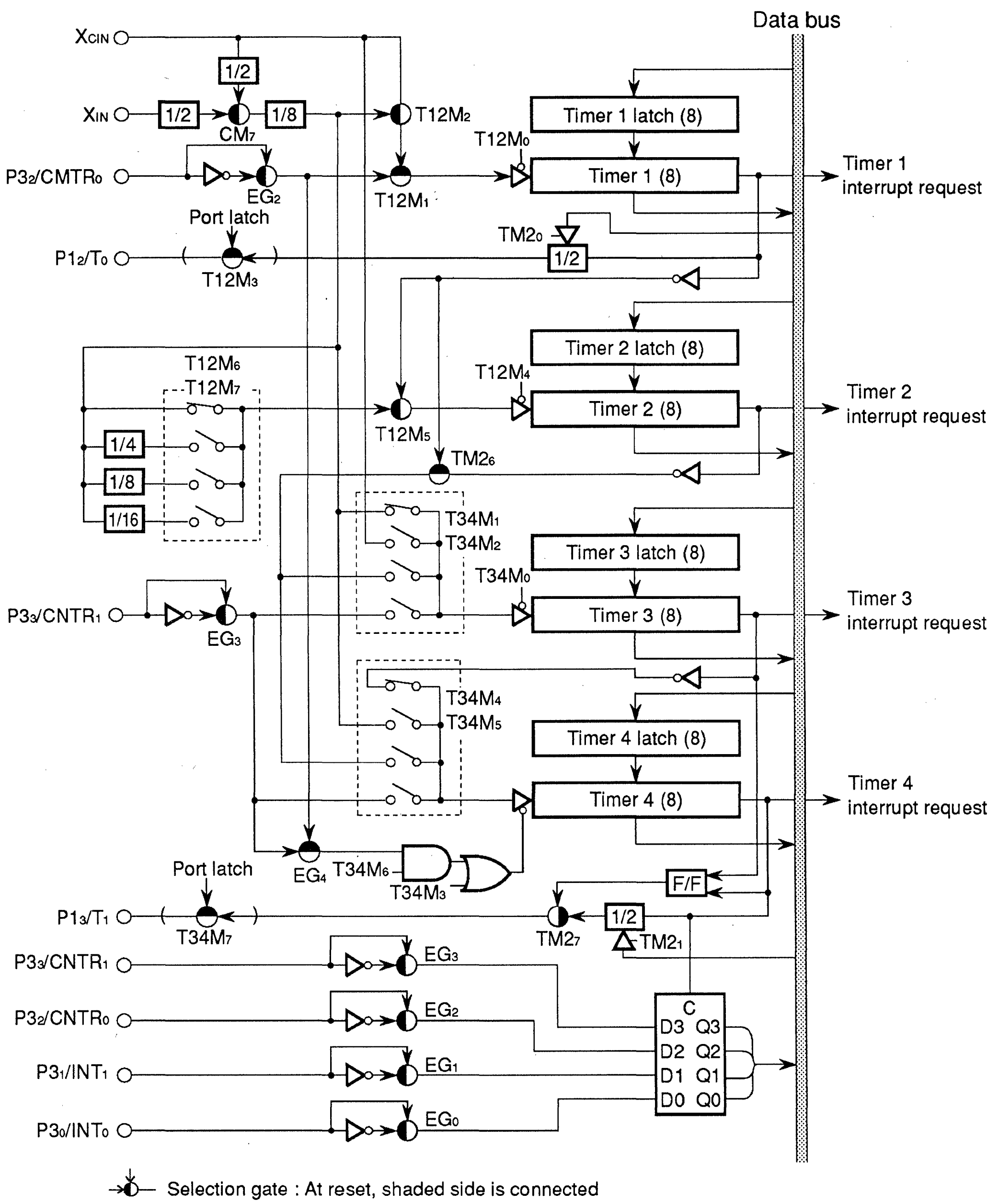
Fig.2.5.7 Operation time before the interrupt routine execution

2.6 Timers

The M37470 has four 8-bit built-in timers, each with an 8-bit timer latch: timer 1, timer 2, timer 3, and timer 4. The timers are all of the count-down type—when the counter of a timer reaches "FF_{16}", the contents of the timer latch decremented by 1 at the next count pulse, is reloaded into the timer, and an interrupt request is generated. The divide ratio of a timer is given by 1/(n + 1), where n is the contents of the latch (n = 0 to 255).
Timers can be set by software, and they can be selected in the following modes:

- Timer mode
- Event count mode
- Pulse output mode
- Pulse width measuring mode
- PWM mode

A block diagram of the timers is shown in Figure 2.6.1.



Note: The M37470M2, M37470M4, and M37470M8 do not have XCIN input.

Fig.2.6.1 Timer block diagram

2.6.1 Timer 1, timer 2, timer 3 and timer 4

Each of timer 1, timer 2, timer 3, and timer 4 is an 8-bit timer with an 8-bit timer latch. If a timer is specified for a write, the corresponding timer latch is also specified at the same time. Therefore, if the value set in the timer is assumed to be n_{16}, the value of the timer latch is also n_{16}. When the timer starts to count, the timer's value decrements at the fall of each count pulse, in the sequence: $n_{16} \rightarrow (n_{16}-1) \rightarrow (n_{16}-2) \rightarrow .....1_{16} \rightarrow 0_{16} \rightarrow FF_{16}$. At the fall of the next count pulse after the timer reaches "FF_{16}", the value (n_{16}-1) obtained by subtracting one from the reload latch value is set (re-loaded) into the timer, and the count resumes. At the rise of the next count pulse after the timer reaches "FF_{16}", an overflow occurs, and an interrupt request is generated. Timer count timing is shown in Figure 2.6.2.

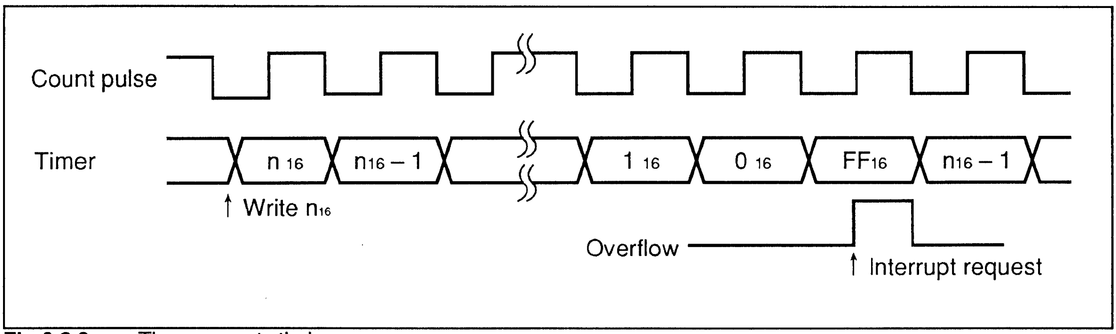


Fig.2.6.2 Timer count timing

(1) Timer 1

Timer 1 can be used in three modes: timer mode, event count mode, and pulse output mode. Start the count of timer 1 by setting bit 0 of the timer 1, 2 mode register (address $00F8_{16}$) to "0".

In timer mode, an interrupt is generated at constant time intervals. The count source can be selected by setting bits 1 and 2 of the timer 1, 2 mode register to specify any one of: the clock oscillation frequency $f(X_{IN})$ or the clock-function clock oscillation frequency $f(X_{CIN})$ divided by 16, the clock-function clock oscillation frequency $f(X_{CIN})$, or an external clock input through the $CNTR_0$ pin. Use bit 7 of the CPU mode register to select $f(X_{IN})$ or $f(X_{CIN})$. Do not select $f(X_{CIN})$ as the count source in the M37470M2, M37470M4, and M37470M8.

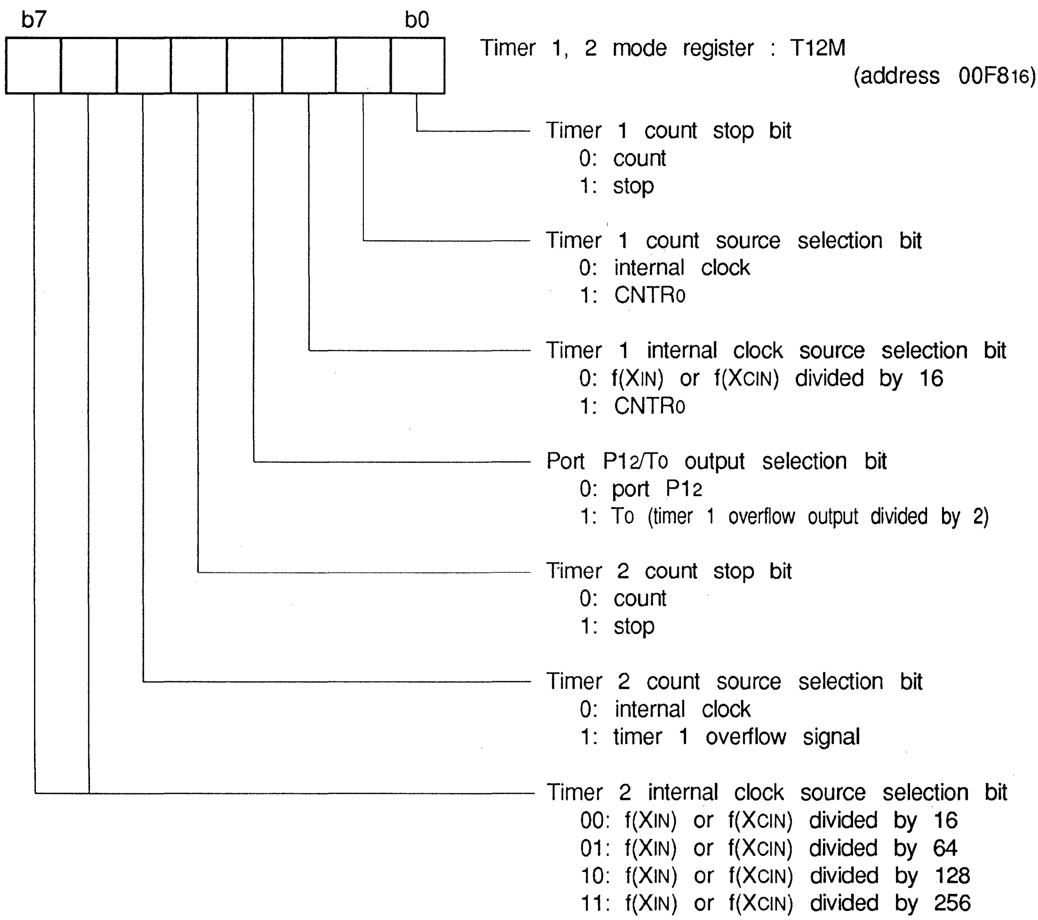
In event count mode, the operation is the same as in timer mode, except that count source is an external clock input through the $CNTR_0$ pin. The active edge of input pulses can be selected by changing bit 2 of the active edge selection register (address $00D4_{16}$). When this bit is "0", pulses input through the $CNTR_0$ pin are inverted to become count pulses; when this bit is "1", pulses input through the $CNTR_0$ pin are used unchanged as count pulses.

In pulse output mode, if bit 3 of the timer 1, 2 mode register (address $00F8_{16}$) is set to "1", port $P1_2$ becomes the timer output T_0, and a signal that is the timer 1 overflow signal divided by two is output. To activate pulse output mode, set the $P1_2$ direction register to output mode. In this case, the initial output value can be set by writing to bit 0 of the timer FF register (address $00F7_{16}$) while bit 0 of the timer mode register 2 (address $00FA_{16}$) is "1" (setting enabled).

(2) Timer 2

Timer 2 can be used in timer mode. Start the count of timer 2 by setting bit 4 of the timer 1, 2 mode register (address $00F8_{16}$) to "0". The count source can be selected by setting bits 5, 6, and 7 of the timer 1, 2 mode register to specify any one of: the clock oscillation frequency f(X_{IN}) or the clock-function clock oscillation frequency f(X_{CIN}) divided by 16, 64, 128 or 256; or the timer 1 overflow signal.

Use bit 7 of the CPU mode register to select f(X_{IN}) or f(X_{CIN}). Do not select f(X_{CIN}) as the count source in the M37470M2, M37470M4, and M37470M8.



Note: Do not select f(X_{CIN}) as the count source in the M37470M2, M37470M4, and M37470M8.

Fig.2.6.3 Structure of timer 1, 2 mode register

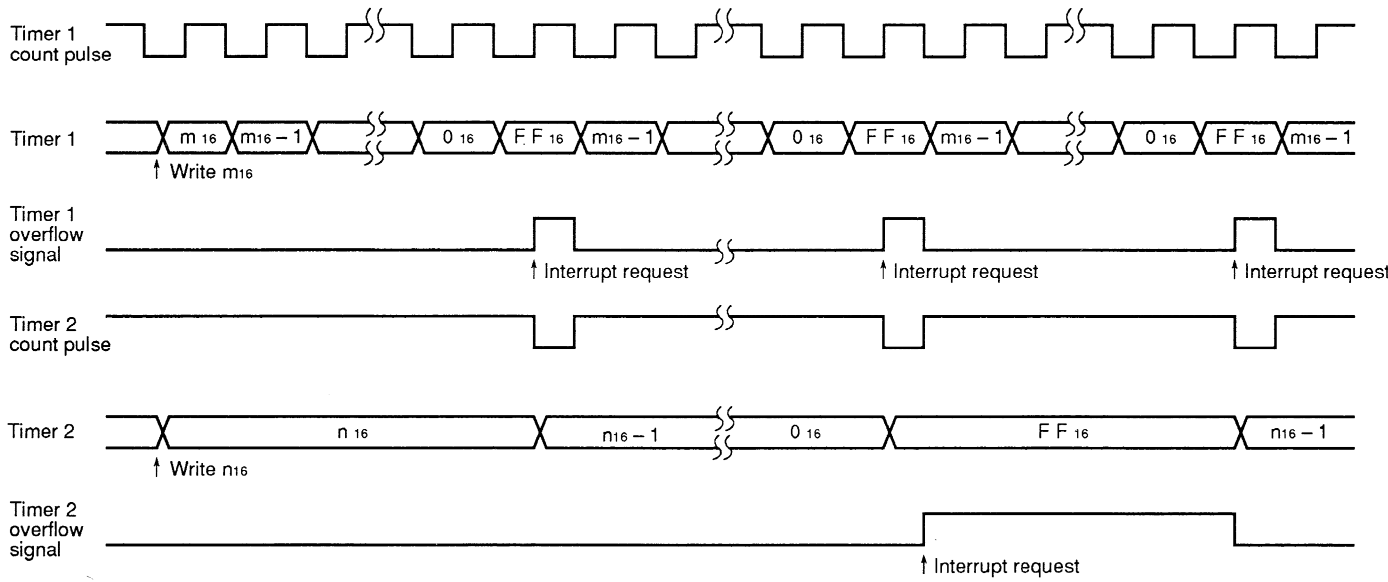


Fig.2.6.4 Timing chart for timer 2 when its count source is timer 1 overflow signal

(3) Timer 3

Timer 3 can be used in three modes: timer mode, event count mode, and PWM mode. Start the count of timer 3 by setting bit 0 of the timer 3, 4 mode register (address $00F9_{16}$) to "0".

In timer mode, an interrupt is generated at constant time intervals. The count source can be selected by setting bits 1 and 2 of the timer 3, 4 mode register and bit 6 of the timer mode register 2 (address $00FA_{16}$) to specify any one of: the clock oscillation frequency $f(X_{IN})$ or the clock-function clock oscillation frequency $f(X_{CIN})$ divided by 16, the clock-function clock oscillation frequency $f(X_{CIN})$, the timer 1 overflow signal, the timer 2 overflow signal, or an external clock input through the $CNTR_1$ pin.

Use bit 7 of the CPU mode register to select $f(X_{IN})$ or $f(X_{CIN})$. Do not select $f(X_{CIN})$ as the count source in the M37470M2, M37470M4, and M37470M8.

Note that, if bits 2 and 1 of the timer 3, 4 mode register are [10] and the timer 1 overflow signal is selected as the count source of timer 2, this timer 1 overflow signal is also the count source of timer 3, regardless of the value of bit 6 of the timer mode register 2.

In event count mode, the operation is the same as in timer mode, except that count source is an external clock input through the $CNTR_1$ pin. The active edge of input pulses can be selected by changing bit 3 of the active edge selection register (address $00D4_{16}$). When this bit is "0", pulses input through the $CNTR_1$ pin are inverted to become count pulses; when this bit is "1", pulses input through the $CNTR_1$ pin are used unchanged as count pulses.

For details of PWM mode, see "2.6.2 PWM mode".

For details of the operation when control returns from a low power consumption mode, see "2.12 Low power consumption modes".

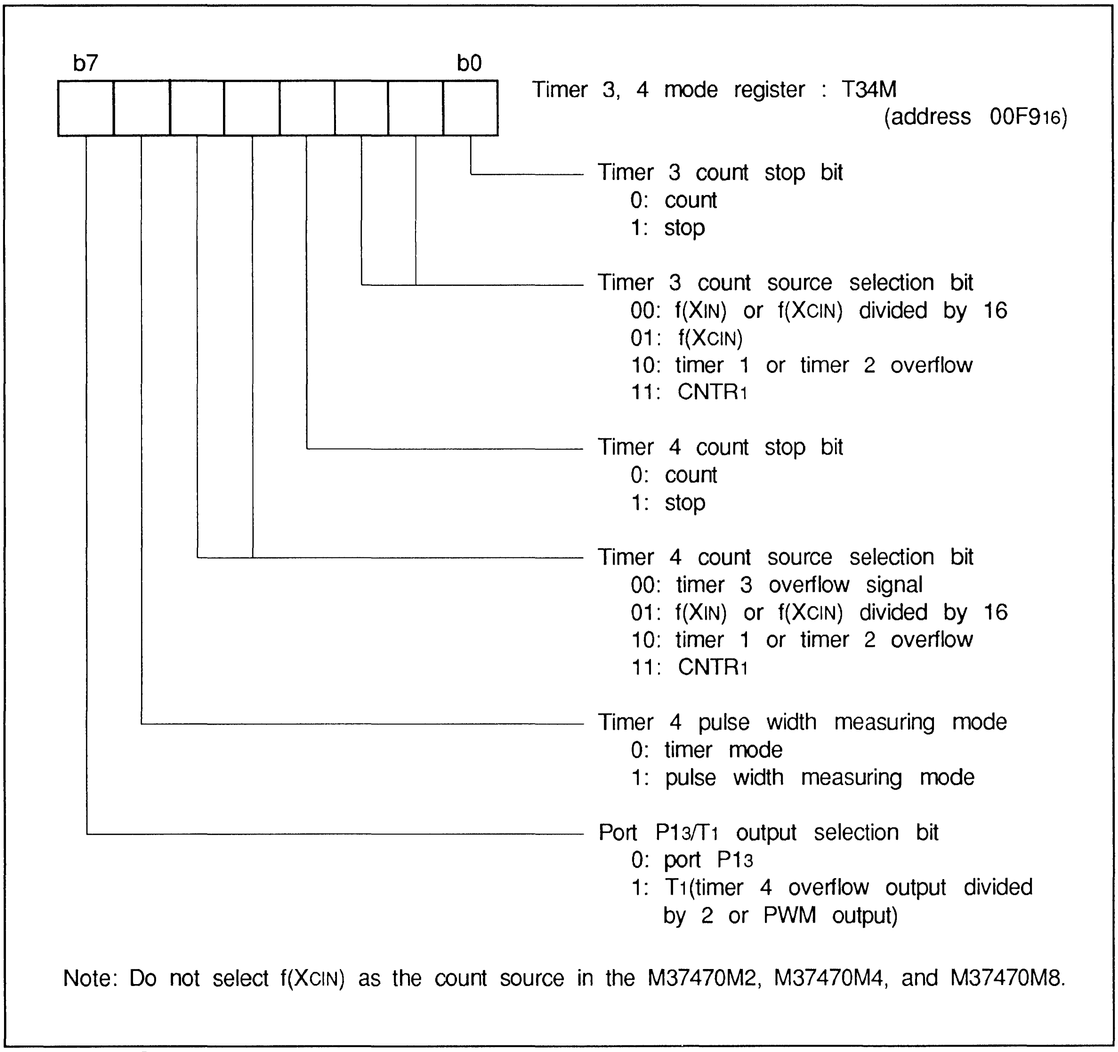


Fig.2.6.5 Structure of timer 3, 4 mode register

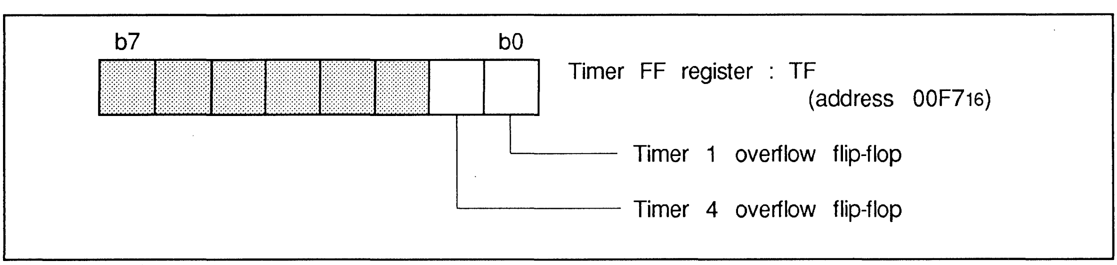


Fig.2.6.6 Structure of timer FF register

(4) Timer 4

Timer 4 can be used in five modes: timer mode, event count mode, pulse output mode, external pulse width measuring mode, and PWM mode. Start the count of timer 4 by setting bit 3 of the timer 3, 4 mode register (address $00F9_{16}$) to "0" when bit 6 of the same register is "0". Set bit 6 of the timer 3, 4 mode register to "1" to activate pulse width measurement mode, and set bit 3 to "1" to stop timer 4. Timer 4 also has an input latch function whereby the level of an external input pin can be latched in the input latch register when timer 4 overflows.

In timer mode, an interrupt is generated at constant time intervals. The count source can be selected by setting bits 4 and 5 of the timer 3, 4 mode register and bit 6 of the timer mode register 2 to specify any one of: the timer 3 overflow signal, the clock oscillation frequency $f(X_{IN})$ or the clock-function clock oscillation frequency $f(X_{CIN})$ divided by 16, the timer 1 overflow signal, the timer 2 overflow signal, or an external clock input through the $CNTR_1$ pin.

Use bit 7 of the CPU mode register to select $f(X_{IN})$ or $f(X_{CIN})$. Do not select $f(X_{CIN})$ as the count source in the M37470M2, M37470M4, and M37470M8.

Note that, if bits 5 and 4 of the timer 3, 4 mode register are [10] and the timer 1 overflow signal is selected as the count source of timer 2, this timer 1 overflow signal is also the count source of timer 4, regardless of the value of bit 6 of the timer mode register 2.

In event count mode, the operation is the same as in timer mode, except that count source is an external clock input through the $CNTR_1$ pin. The active edge of input pulses can be selected by changing bit 3 of the active edge selection register (address $00D4_{16}$). When this bit is "0", pulses input through the $CNTR_1$ pin are inverted to become count pulses; when this bit is "1", pulses input through the $CNTR_1$ pin are used unchanged as count pulses.

In pulse output mode, if bit 7 of the timer 3, 4 mode register is set to "1", port $P1_3$ becomes the timer output T_1, and a signal that is the timer 4 overflow signal divided by two is output. To activate rectangular waveform mode, set the $P1_3$ to output mode. In this case, the initial output value can be set by writing to bit 1 of the timer FF register (address $00F7_{16}$) while bit 1 of the timer mode register 2 (address $00FA_{16}$) is "1" (setting enabled).

In pulse width measuring mode, the "H" level width or the "L" level width of the signal input to either the $CNTR_0$ pin or the $CNTR_1$ pin can be measured. To activate external pulse width measuring mode, set bit 6 of the timer 3, 4 mode register to "1". Select the count source with bits 4 and 5 of the timer 3, 4 mode register, then set bit 3 to "0" to count the number of timer 4 pulses generated while the signal at the $CNTR_0$ pin or the $CNTR_1$ pin are either "H" or "L".

Select either the $CNTR_0$ pin or the $CNTR_1$ pin by bit 4 of the active edge selection register (address $00D4_{16}$), and select either the count during the "H" interval or the count during the "L" interval by bit 2 or bit 3 of the same register. When the edge selection bit is "0" (falling edge), pulses generated during the "H" interval are counted; when the edge selection bit is "1" (rising edge), pulses generated during the "L" interval are counted.

For details of PWM mode and the input latch function, see the appropriate sections.

2.6.2 PWM mode

The M37470 can output PWM waveforms from the T_1 pin, using timer 3 and timer 4.
Set bit 7 of the timer mode register 2 (address $00FA_{16}$) to "1" to set timer 3 and timer 4 to PWM mode and set bit 7 of the timer 3, 4 mode register (address $00F9_{16}$) to "1" to make port $P1_3$ the timer output T_1 pin. In this case, set the port $P1_3$ to output mode.
Select a count source in the same way as when timer 3 and timer 4 are operating as ordinary timers (in timer mode or event count mode), and put them in operating status. Do not select the timer 3 overflow signal as the count source for timer 4.
When PWM mode is active, timer 3 counts pulses and timer 4 is stopped while the PWM waveform is "L". When timer 3 overflows, the PWM waveform goes "H", timer 3 stops, and timer 4 starts to count pulses. When timer 4 overflows, the PWM waveform goes "L", timer 4 stops, and timer 3 again starts to count pulses. Therefore, the "L" level width of the PWM waveform is set by timer 3, and the "H" level width is set by timer 4.
If either timer 3 or timer 4 is written to while it is operating in PWM mode, only the corresponding latch is written to; the timer itself is not overwritten. When that timer subsequently overflows, the value obtained by subtracting one from the reload latch value is set (reloaded) to the timer. If a timer is written to while it is stopped in PWM mode, the values in both the timer and the timer latch are overwritten.
When operation changes from another mode to PWM mode, the PWM waveform starts at "L".

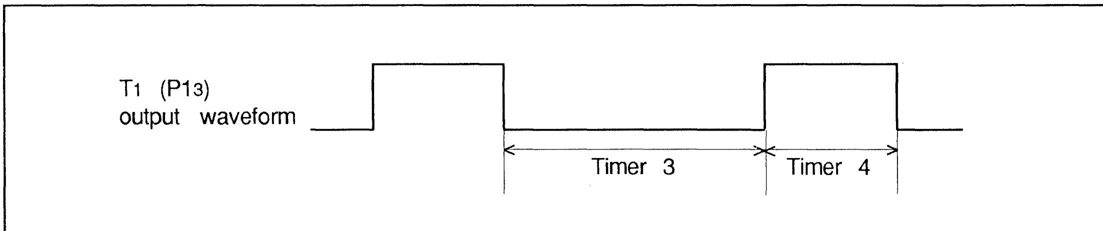


Fig.2.6.7 PWM waveform

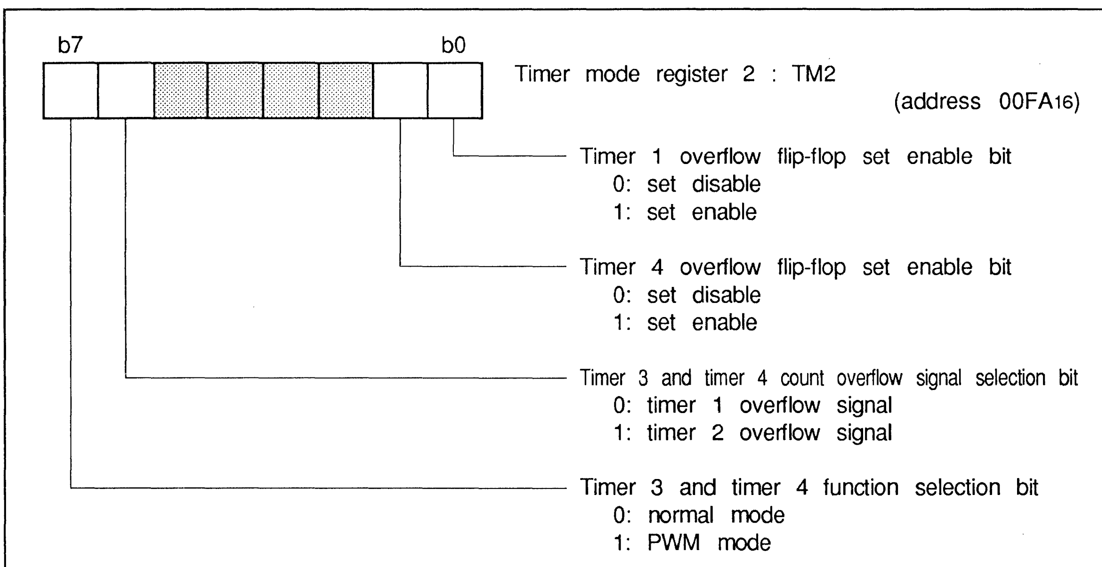


Fig.2.6.8 Structure of timer mode register 2

2.6.3 Input latch function

The M37470 has a function that latches the levels of the INT_0, INT_1, $CNTR_0$, and $CNTR_1$ signals in an input latch register (address $00D6_{16}$) after timer 4 overflows. This function enables the user to accurately determine the levels of the pins at the instant that timer 4 overflowed. The active edges of the pins' signals latched in the input latch register are selected by the corresponding bits of the active edge selection register (address $00D4_{16}$). When bit 0 of the active edge selection register is "0", the level of the INT_0 pin is inverted then latched; when it is "1", the level of the INT_0 pin is latched as it is. Similarly, when bit 1 is "0", the level of the INT_1 pin is inverted then latched; when it is "1", the level of the INT_1 pin is latched as it is. When bit 2 is "0", the level of the $CNTR_0$ pin is inverted then latched; when it is "1", the level of the $CNTR_0$ pin is latched as it is. When bit 3 is "0", the level of the $CNTR_1$ pin is inverted then latched; when it is "1", the level of the $CNTR_0$ pin is latched as it is.

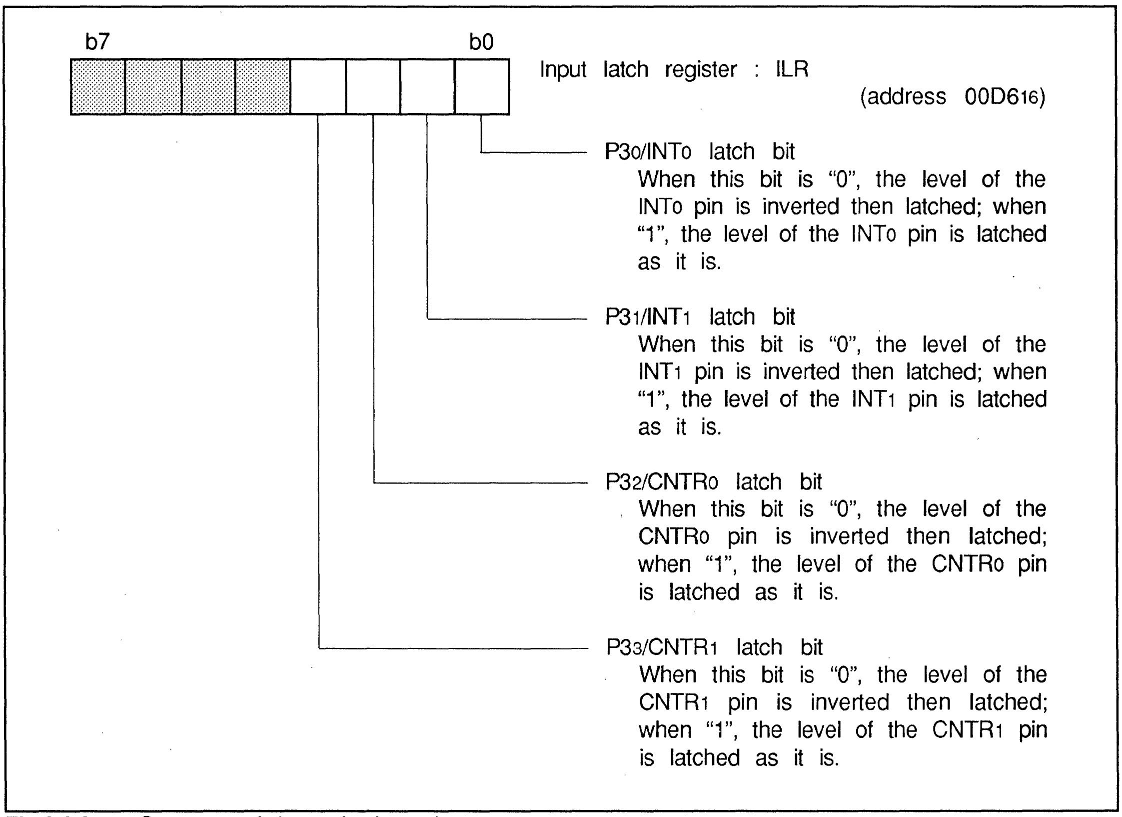


Fig.2.6.9 Structure of input latch register

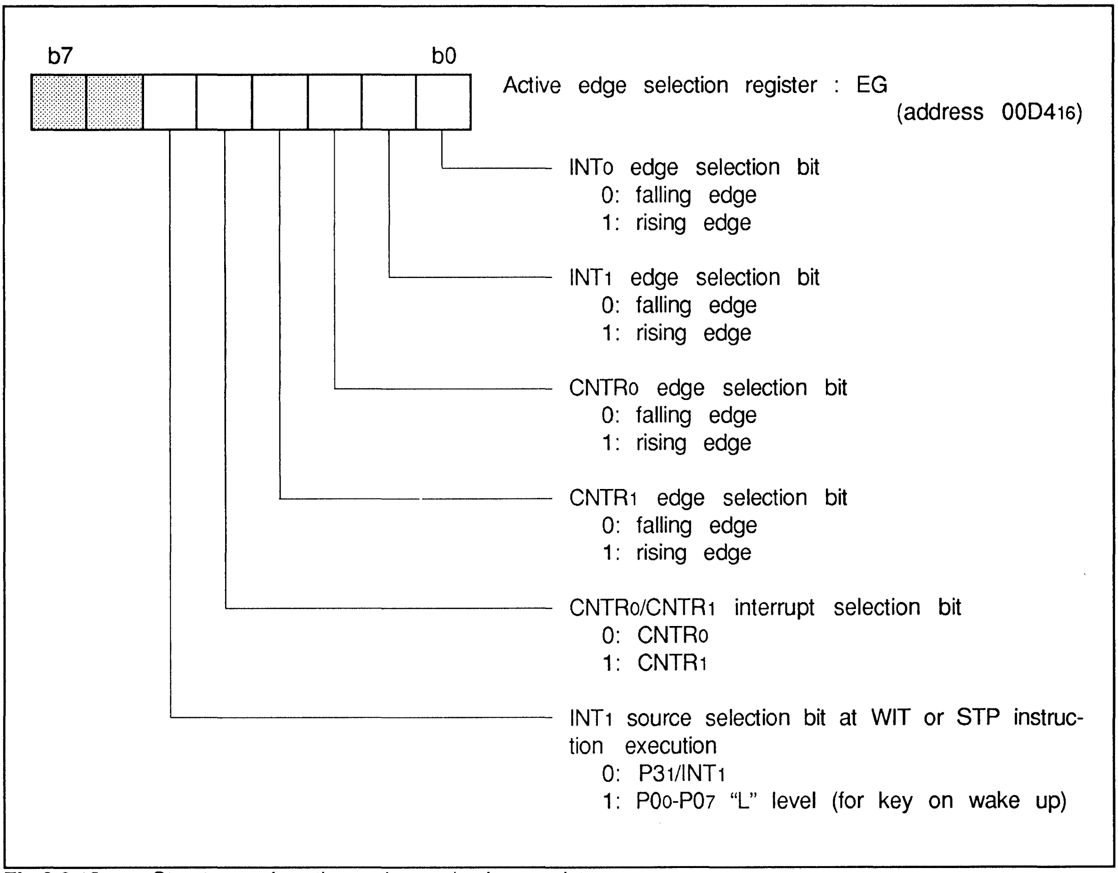


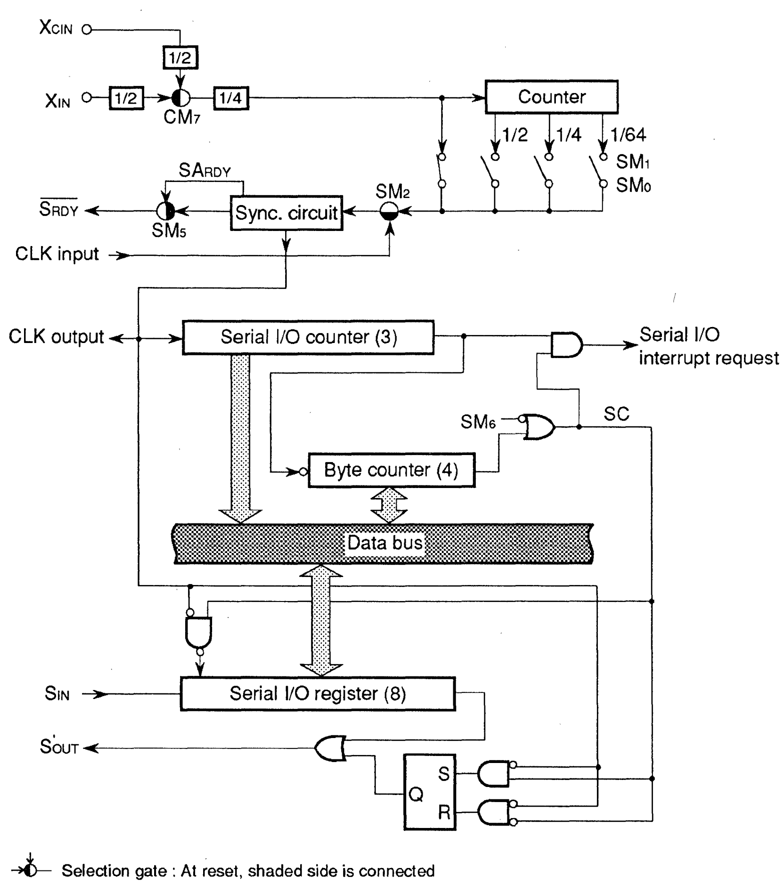
Fig.2.6.10 Structure of active edge selection register

2.7 Serial I/O

The serial I/O function can transmit and receive 8-bit data in serial by a clock-synchronized method.

2.7.1 Structure of serial I/O section

The serial I/O section consists of a serial I/O register (address $00DD_{16}$), a serial I/O mode register (address $00DC_{16}$), a synchronization clock generating circuit, and a byte counter (address $00DE_{16}$). A block diagram of the serial I/O section is shown in Figure 2.7.1, and the structure of the serial I/O mode register is shown in Figure 2.7.2.
The serial I/O register is an 8-bit serial-to-parallel conversion register for data transfer. During transmission, it sends data one bit at a time, starting at the least significant bit; during reception, it receives data one bit at a time, starting at the most significant bit. The transfer clock for serial data can be selected as one of: the clock oscillation frequency $f(X_{IN})$ or the clock-function clock oscillation frequency $f(X_{CIN})$ divided by 8, 16, 32, or 512; or an external clock. The M37470 also has a byte specification mode that uses the byte counter.



Note: The M37470M2, M37470M4, and M37470M8 do not have XCIN input.

Fig.2.7.1 Serial I/O block diagram

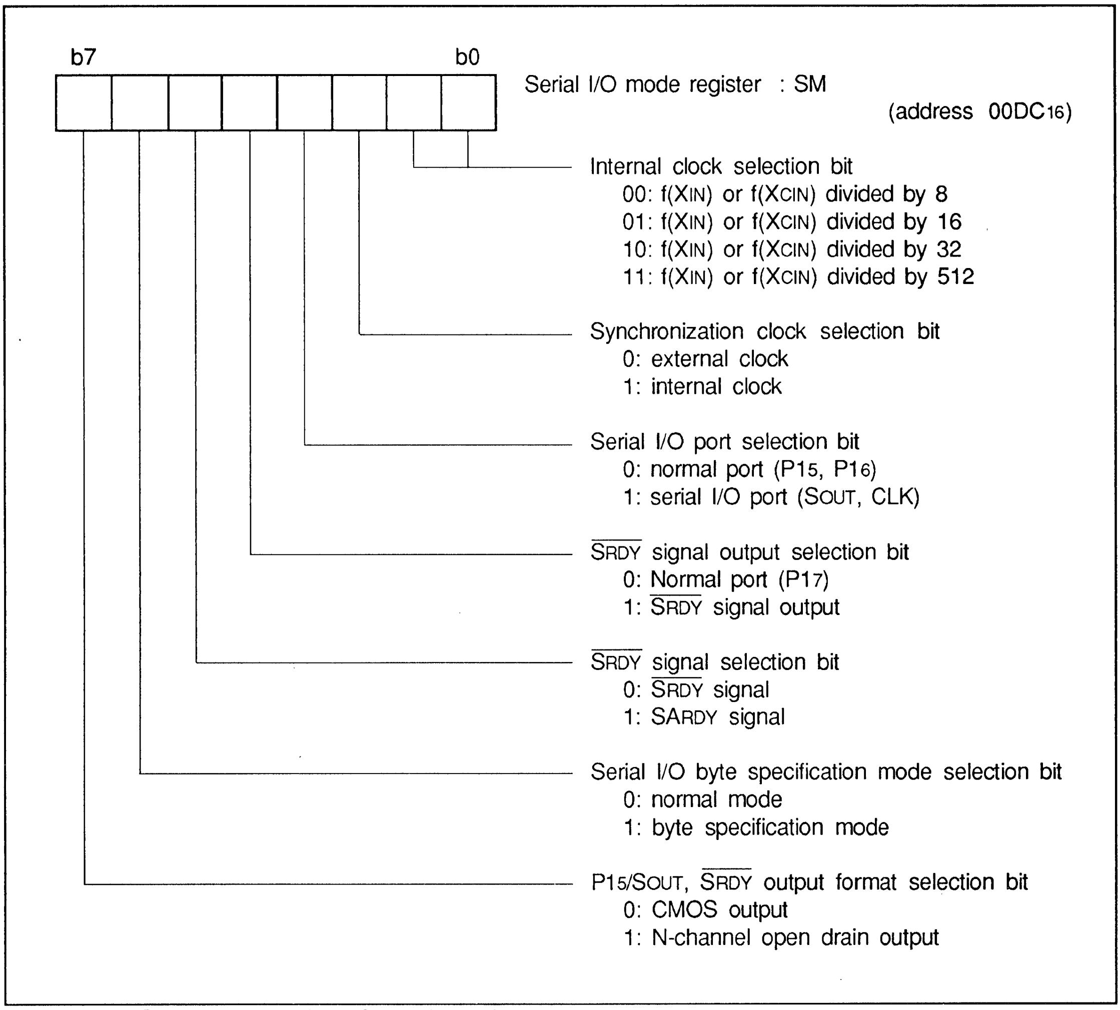
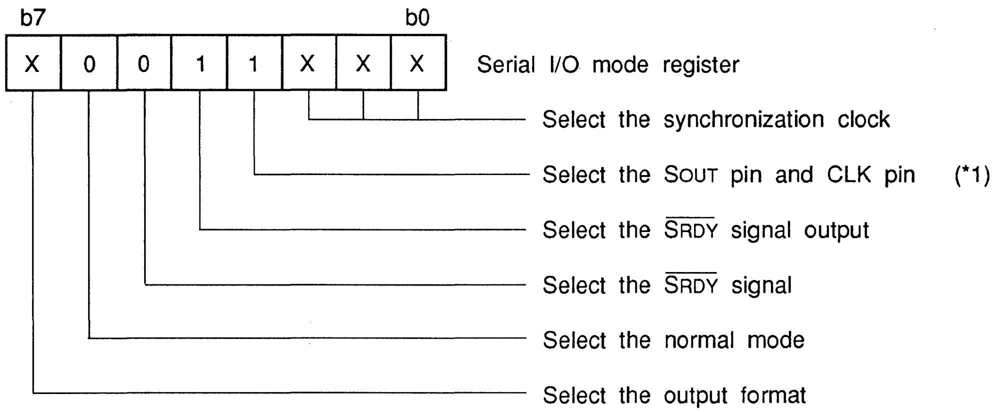


Fig.2.7.2 Structure of serial I/O mode register

2.7.2 Serial I/O data receive

An example of the setting of the serial I/O mode register for data reception is shown in Figure 2.7.3.



*1: When external clock is selected, this pin can also be used as an ordinary port. However, port P16 must be set to input mode.

Fig.2.7.3 Example of the serial I/O mode register setting (data receive)

In addition, set the bit 4 of the port P1 (SIN pin) direction register to "0" to set the port P14 to input mode.

When the above settings are completed, write dummy data to the serial I/O register (address 00DD16) (if external clock is selected, write when the transfer clock is "H")—the serial I/O counter will be set to "7", and the transfer clock will be forced to "H". The transfer clock then goes "H" → "L" → "H" eight times and stops. Receive data input through the SIN pin is then input to the serial I/O register one bit at a time in synchronism with rising edges of the transfer clock, starting at the most significant bit (bit 7), and the contents of this register are shifted toward the least significant bit each time new data is input.

Once eight bits of data have been input in this way, an interrupt request is generated at the rise of the last transfer clock, and the serial I/O interrupt request bit (bit 6 of the interrupt request register 1) is set.

When an external clock is selected, the contents of the serial I/O register will continue to shift while the transfer clock is being input, so the transfer clock must be stopped externally to prevent this shift.

Note: When the SOUT pin is selected, data written to the serial I/O register is output from the SOUT pin in synchronism with the falling edge of the transfer clock.

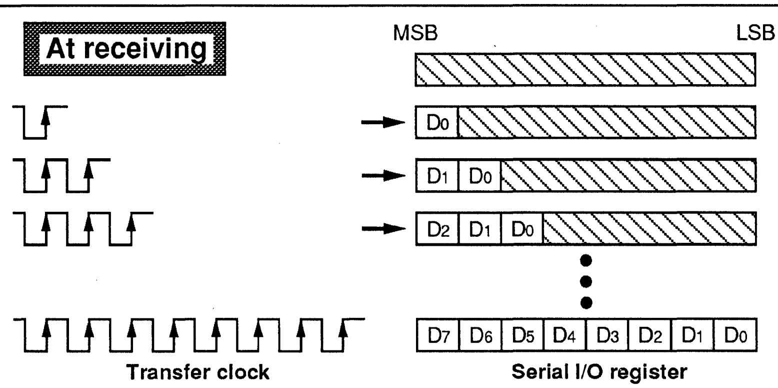


Fig.2.7.4 Serial I/O register at receiving

2.7.3 Serial I/O data transmit

An example of the setting of the serial I/O mode register for data transmission is shown in Figure 2.7.5.

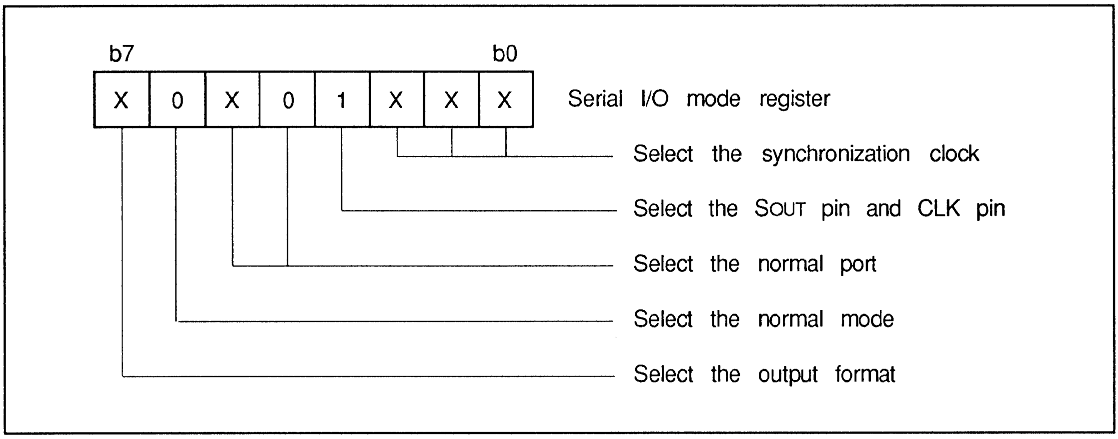


Fig.2.7.5 Example of the serial I/O mode register setting (data transmit)

When the above settings are completed, write the data to be transmitted into the serial I/O register (address $00DD_{16}$) (if external clock is selected, write when the transfer clock is "H")—the serial I/O counter will be set to "7", and the transfer clock will be forced to "H". The transfer clock then goes "H" → "L" → "H" eight times and stops. Transmission data output from the SOUT pin is then output one bit at a time from the serial I/O register in synchronism with falling edges of the transfer clock, starting with the least significant bit, and the contents of the serial I/O register are shifted toward the least significant bit each time one bit of data is output.

Once eight bits of data have been output in this way, an interrupt request is generated at the rise of the last cycle of the transfer clock, and the serial I/O interrupt request bit (bit 6 of the interrupt request register 1) is set.

When an external clock is selected, the contents of the serial I/O register will continue to shift while the transfer clock is being input, so the transfer clock must be stopped externally to prevent this shift.

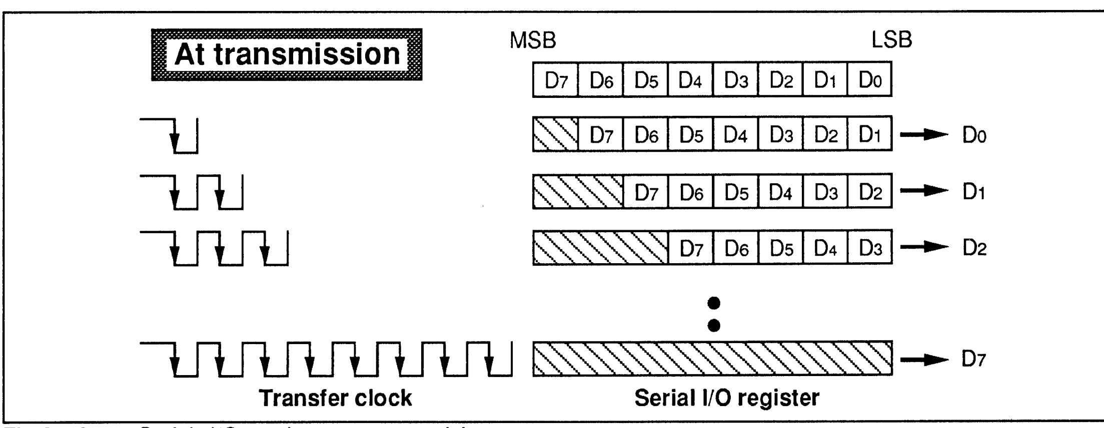


Fig.2.7.6 Serial I/O register at transmitting

Table 2.7.1 shows the corresponding between serial I/O relation ports and serial I/O mode register.

Table 2.7.1 Corresponding between serial I/O relation ports and serial I/O mode register.

Ports \ Serial I/O mode register			bit 7	bit 4	bit 3	bit 2
P17/$\overline{\text{SRDY}}$	P17 (CMOS)		×	0	×	×
	$\overline{\text{SRDY}}$	CMOS	0	1	×	×
		N-channel	1	1	×	×
P16/CLK	P16		×	×	0	×
	CLK	Input	×	×	×	0
		Output	×	×	1	1
P15/SOUT	P15	CMOS	0	×	0	×
		N-channel	1	×	0	×
	SOUT	CMOS	0	×	1	×
		N-channel	1	×	1	×

Figure 2.7.7 shows the connection example of serial I/O transfer. And Figure 2.7.8 shows the data transfer operation sequence, and Figure 2.7.9 shows its timing diagram.

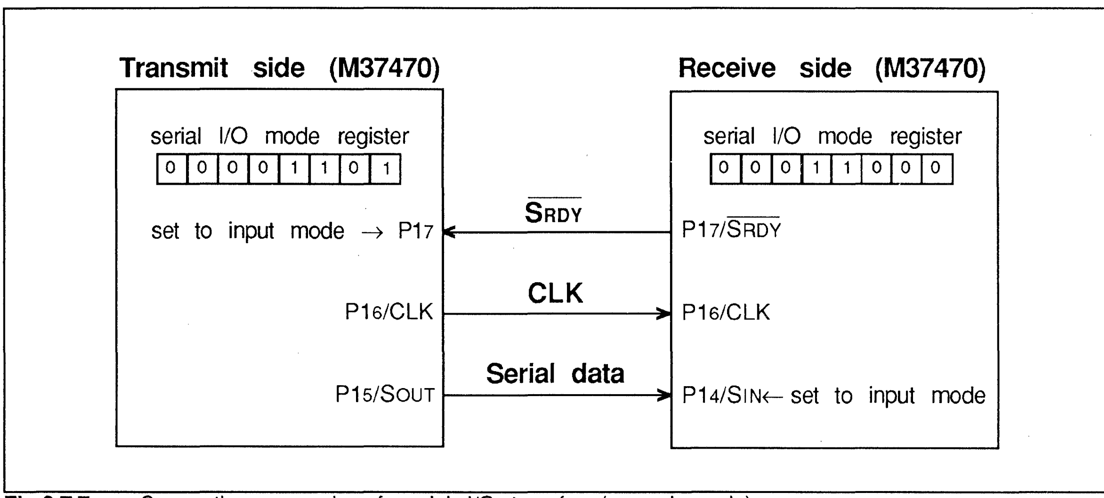


Fig.2.7.7 Connection example of serial I/O transfer (normal mode)

Transmission side		Reception side
Set port P17 to the input mode `CLB 7,$C3`		Set port P14 to the input mode `CLB 4,$C3`
Set serial I/O mode register `LDM #$0D,$DC`		Set serial I/O mode register `LDM #$18,$DC`
Clear serial I/O interrupt request bit `CLB 6,$FC`		Clear serial I/O interrupt request bit `CLB 6,$FC`
Set serial I/O interrupt enable bit `SEB 6,$FE`		Set serial I/O interrupt request bit `SEB 6,$FE`
	*	Write dummy data to the serial I/O register, set $\overline{\text{SRDY}}$ to "L"......(1) `LDM #$FF,$DD`
Check the port P17—if it is at "L", proceed to data output sequence `TEST:BBS 7,$C2,TEST`		
Write transfer data into serial I/O register......(2) `LDM #DATA,$DD`		

The above process transfers one byte of data from the transmission side to the reception side. At completion, both sides generate an interrupt request, which posts the completion of data transfer. Further data can then be transferred by repeating the processing from the asterisk (*) onward in the figure.
The steps (1) and (2) are corresponding to the figure 2.7.9.

Fig.2.7.8 Serial I/O data transfer operation sequence (normal mode)

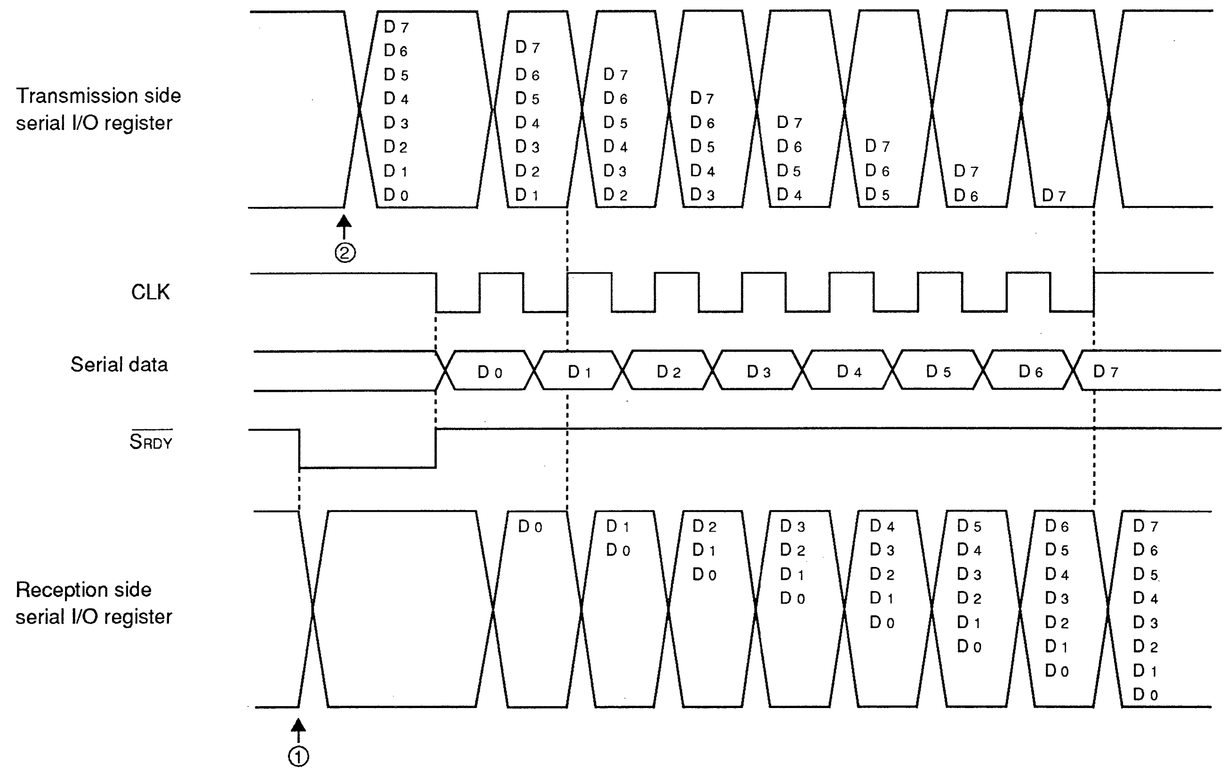
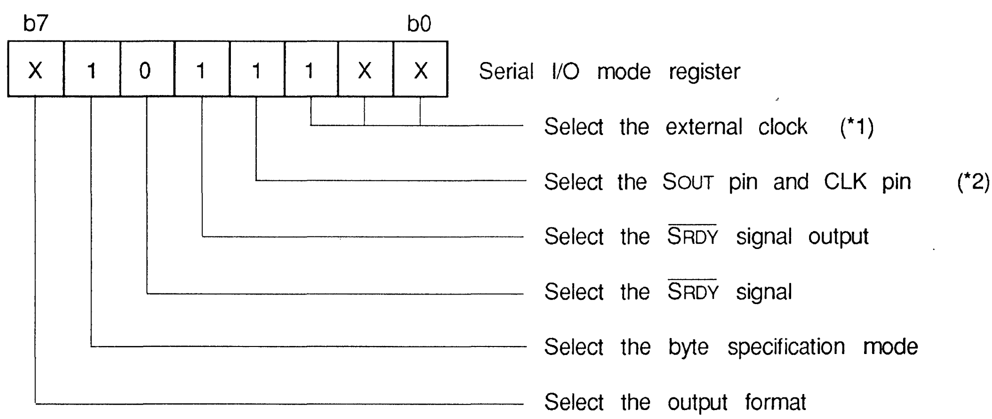


Fig.2.7.9 Timing diagram of serial I/O data transfer (normal mode)

2.7.4 Byte specification mode

The M37470's serial I/O function also has a byte specification mode which is useful when the serial I/O circuits of more than one microcomputer are connected together and data is being transferred between the microcomputers over a single serial I/O bus. In other words, this mode can lighten the software load—if only one byte of data in a specific cycle is to be transferred from among a number of bytes of data being transferred over the serial I/O bus, the user can write the number of that cycle (e.g., the nth cycle) into the byte counter to specify that the software does not need to perform serial I/O-related processing during other cycles.
An example of the setting of the serial I/O mode register for byte specification mode is shown in Figure 2.7.10.



*1: In this mode, external clock must always be selected as the synchronization clock.
*2: Can also be used as an ordinary port during reception. However, port $P1_6$ must then be set to input mode.

Fig.2.7.10 Example of the serial I/O mode register setting (byte specification mode)

For reception, set the bit 4 of the port P1 (SIN pin) direction register to "0" to set the port $P1_4$ to input mode; for transmission, set bit 3 of the serial I/O mode register to "1" to select the SOUT pin.

After setting the serial I/O mode register, write into the byte counter the number of bytes after which the serial transfer is to occur. If the value written to the byte counter is n, serial transfer occurs at the clock for the (n+1)th byte.
When the data to be transmitted (or dummy data, for data reception) is written to the serial I/O register, the transfer operation starts. The value in the byte counter is decremented by 1 at the input of each 8 cycles of the transfer clock. When this value reaches "0", serial transfer occurs for the next 8 cycles only, in exactly the same way as in normal mode. The transfer clocks up to that point do not cause any serial transfer. The byte counter is then decremented by 1 again to become 15, and the serial transfer stops.
While the value in the byte counter is not "0", the output from the SOUT pin goes "H" at the fall of the first transfer clock. Therefore, if the output format of the SOUT pin is set to N-channel open drain, the SOUT pin will be at high impedance, so there is no need to connect it to the SOUT pins of the other microcomputers.

A serial I/O interrupt request is generated only after the value in the byte counter has reached "0" and the serial transfer has ended.
An external clock can be used as the transfer clock in byte specification mode, but in that case data can be written to the serial I/O register if the value of the byte counter is not "0", even when the transfer clock is "L".
The serial I/O counter is allocated to the same address as the byte counter but, since the serial I/O counter is a read-only register, writing to the byte counter has no effect on it. The byte counter does not have a reload function, so a new value must be set in it to transfer the next data.

2.7.5 $\overline{\text{SRDY}}$ signal and SARDY signal

When the P17 pin is being used as the $\overline{\text{SRDY}}$ pin, output of either the $\overline{\text{SRDY}}$ signal or the SARDY signal can be selected by bit 5 of the serial I/O mode register.

(1) $\overline{\text{SRDY}}$ signal
The $\overline{\text{SRDY}}$ signal informs the other partner of data transfer between microcomputers that preparations for serial transfer are completed.
In normal mode, the $\overline{\text{SRDY}}$ signal goes "L" when data is written to the serial I/O register. It then goes "H" at the fall of the first transfer clock.
The $\overline{\text{SRDY}}$ signal always goes "L" when data is written to the serial I/O register, then "H" at the fall of the first transfer clock, regardless of the value in the byte counter, even if byte specification mode is activated.

(2) SARDY signal
The SARDY signal is the same as the $\overline{\text{SRDY}}$ signal in that it informs the completion of serial transfer preparations.
In normal mode, the SARDY signal goes "H" when data is written to the serial I/O register. After transfer is completed, the SARDY signal goes "L" at the rise of the final cycle of the transfer clock.
The SARDY signal always goes "H" when data is written to the serial I/O register, then "L" at the rise of the final cycle of the transfer clock after the value in the byte counter reaches "0", even if byte specification mode is activated.
In byte specification mode, if the $\overline{\text{SRDY}}$ pins of more than one microcomputer are connected together, with the SARDY signal selected and N-channel open drain selected for the output format, the SARDY signal goes "H" only when transfer preparations are completed in all of the microcomputers.

Figure 2.7.11 shows the connection example of serial I/O transfer at byte specification mode. And Figure 2.7.12 shows the data transfer operation sequence at byte specification mode, and Figure 2.7.13 shows its timing diagram.

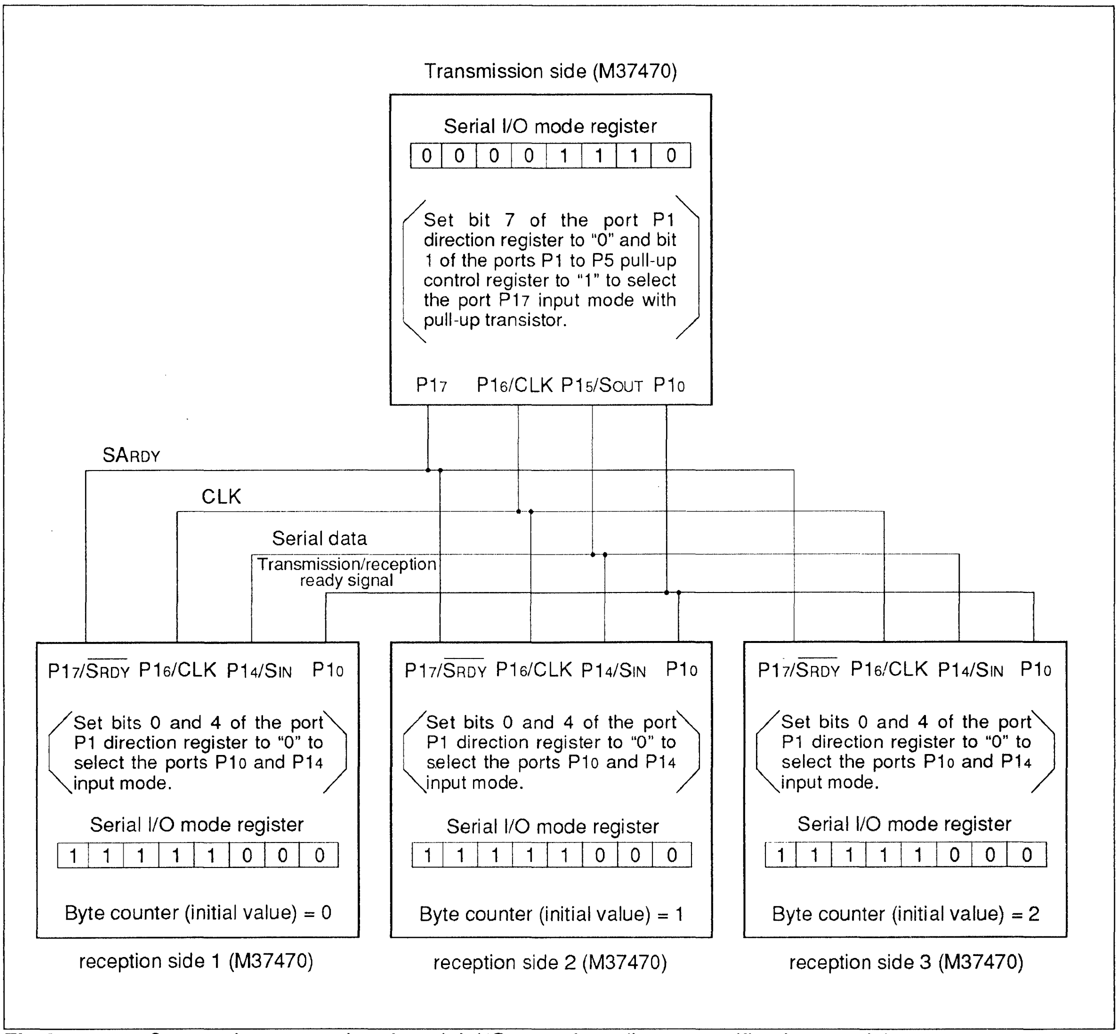


Fig.2.7.11 Connection example of serial I/O transfer (byte specification mode)

Transmission side	Reception side 1	Reception side 2	Reception side 3
• Set the port P1$_7$ to input mode. • Connect the pull-up transistor to the port P1$_7$. • Set the port P1$_0$ to output mode. CLB 7,$C3 SEB 1,$D1 SEB 0,$C3	• Set the port P1$_4$ to input mode. • Set the port P1$_0$ to input mode. CLB 4,$C3 CLB 0,$C3	• Set the port P1$_4$ to input mode. • Set the port P1$_0$ to input mode. CLB 4,$C3 CLB 0,$C3	• Set the port P1$_4$ to input mode. • Set the port P1$_0$ to input mode. CLB 4,$C3 CLB 0,$C3
• Set the serial I/O mode register. LDM #$0E,$DC	• Set the serial I/O mode register. LDM #$F8,$DC	• Set the serial I/O mode register. LDM #$F8,$DC	• Set the serial I/O mode register. LDM #$F8,$DC
• Clear the serial I/O interrupt request bit. CLB 6,$FC	• Clear the serial I/O interrupt request bit. CLB 6,$FC	• Clear the serial I/O interrupt request bit. CLB 6,$FC	• Clear the serial I/O interrupt request bit. CLB 6,$FC
• Set the serial I/O interrupt enable bit. SEB 6,$FE	• Set the serial I/O interrupt enable bit. SEB 6,$FE	• Set the serial I/O interrupt enable bit. SEB 6,$FE	• Set the serial I/O interrupt enable bit. SEB 6,$FE
• Output "L" as the transmission/reception ready signal from port P1$_0$. * CLB 0,$C2			
	• Check the port P1$_0$. If its level becomes "L", it turns into the data input sequence. TEST1:BBS 0,$C2,TEST1	• Check the port P1$_0$. If its level becomes "L", it turns into the data input sequence. TEST2:BBS 0,$C2,TEST2	• Check the port P1$_0$. If its level becomes "L", it turns into the data input sequence. TEST3:BBS 0,$C2,TEST3
	• Set the byte counter to "0". LDM #$00,$DE	• Set the byte counter to "1". LDM #$01,$DE	• Set the byte counter to "2". LDM #$02,$DE
	• Write the dummy data to serial I/O register to set the SARDY signal to "H"....(1) LDM #$FF,$DD	• Write the dummy data to serial I/O register to set the SARDY signal to "H"....(3) LDM #$FF,$DD	• Write the dummy data to serial I/O register to set the SARDY signal to "H"....(2) LDM #$FF,$DD
• Check the port P1$_7$. If its level becomes "H", it turns into the data output sequence. TEST:BBC 7,$C2,TEST			
• Return the port P1$_0$ level to "H". SEB 0,$C2			
• Write the transfer data to reception side 1 to serial I/O register....(4) LDM #DATA1,$DD			
Serial I/O interrupt occurrence by completion of transmission	Serial I/O interrupt occurrence by completion of reception		
• Write the transfer data to reception side 2 to serial I/O register....(5) LDM #DATA2,$DD			
Serial I/O interrupt occurrence by completion of transmission		Serial I/O interrupt occurrence by completion of reception	
• Write the transfer data to reception side 3 to serial I/O register....(6) LDM #DATA3,$DD			
Serial I/O interrupt occurrence by completion of transmission			Serial I/O interrupt occurrence by completion of reception

The above processing serially transfers data 1 to reception side 1, data 2 to reception side 2, and data 3 to reception side 3, all from the transmission side. A corresponding interrupt is generated when each data transfer ends, so the transmission side can know when data transfer is completed. Further data can then be transferred by repeating the processing from the asterisk (*) onward in the figure.

Fig.2.7.12 Serial I/O data transfer operation sequence (byte specification mode)

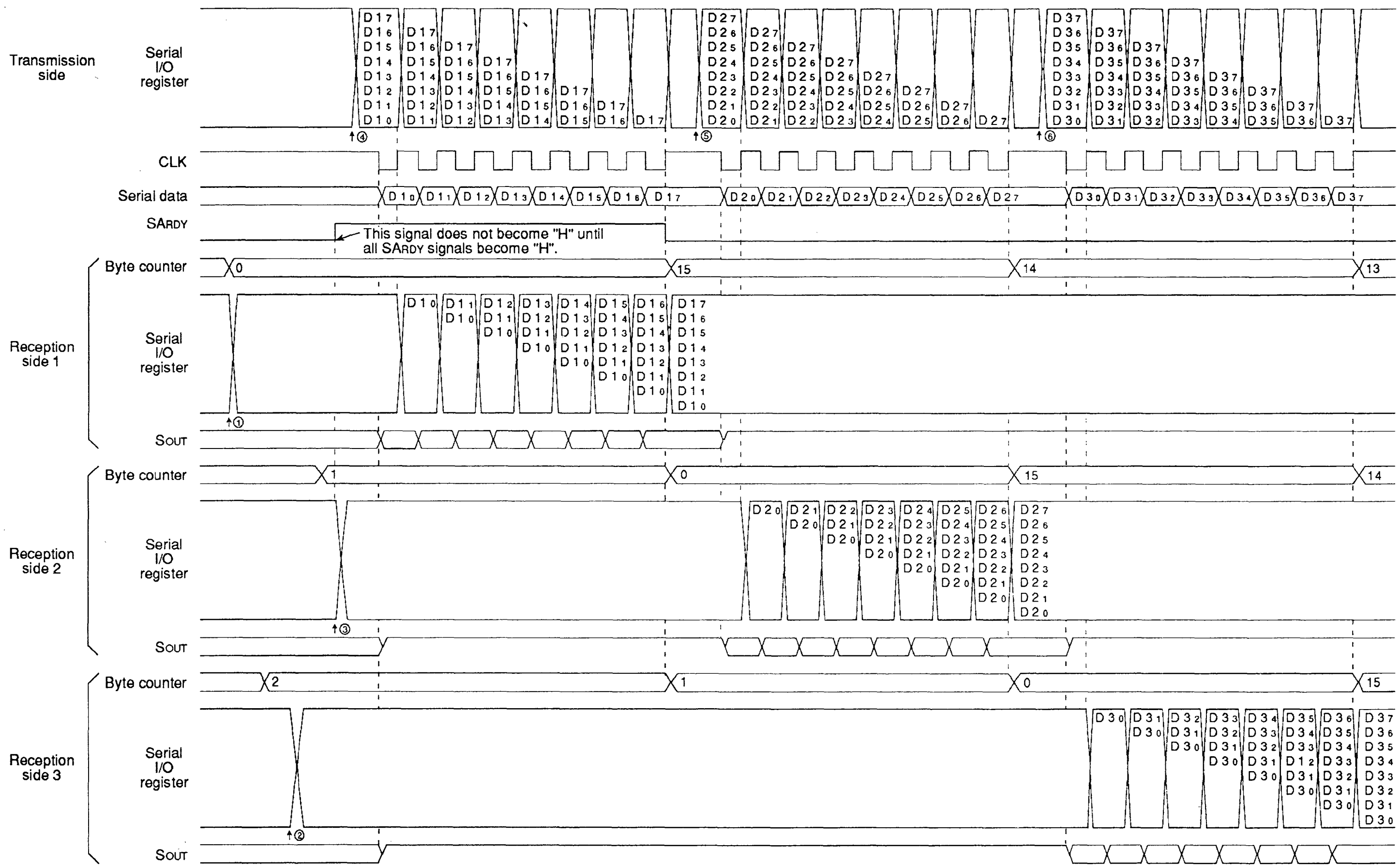


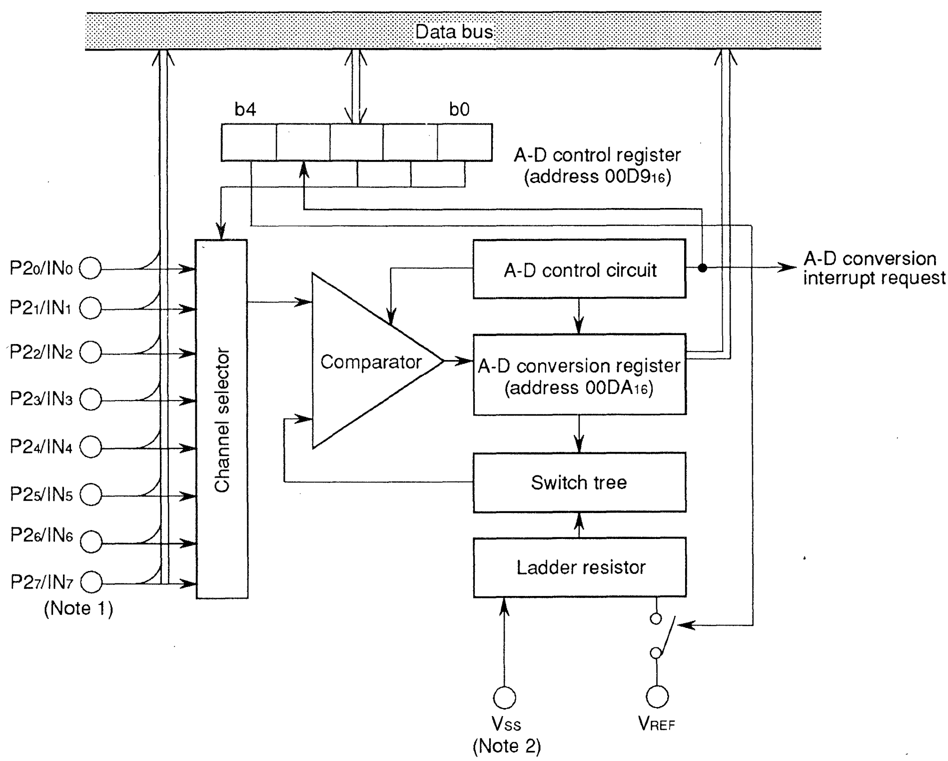
Fig.2.7.13 Timing diagram of serial I/O data transfer (byte specification mode with SARDY signal)

2.8 A-D converter

The A-D converter built into the M37470 has the following characteristics:

- Analog input pins: 8 channels (also used as port P2)
- Conversion method: Successive approximation comparison
- Resolution: 8 bits
- Non-linearity error: ±2 LSB
- Conversion speed: 25μs (at $f(X_{IN})$ = 4MHz)

A block diagram of the A-D converter is shown in Figure 2.8.1.



Note 1: The M37470M2, M37470M4, and M37470M8 do not have the $P2_4/IN_4$ to $P2_7/IN_7$ pins.
2: AV_{SS} for 56-pin QFP type.

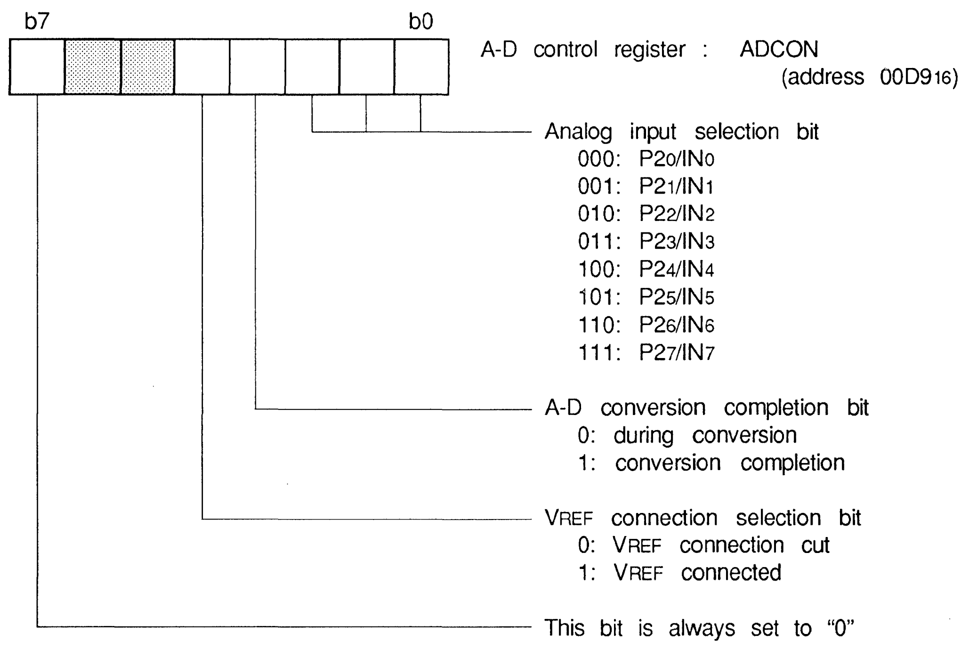
Fig.2.8.1 A-D converter block diagram

2.8.1 Block description

(1) A-D conversion register (address 00DA$_{16}$)
The A-D conversion register is a read-only register that contains the result of an A-D conversion. Do not read the contents of this register during A-D conversion.

(2) A-D control register (address 00D9$_{16}$)
The A-D control register controls the A-D conversion process. Bits 0 to 2 of this register are analog input selection bits that enable analog input from specific analog input pins. Pins that are not being used for analog input can be used as ordinary I/O ports.
Bit 3 is an A-D conversion completion bit—A-D conversion starts when "0" is written to this bit. The value of this bit remains at "0" during A-D conversion, then changes to "1" at the same time that A-D conversion ends.
Bit 4 is a V_{REF} connection selection bit that controls a connection switch between ladder resistors and the V_{REF} pin. When A-D conversion is not being used, set this bit to "0" to disconnect the ladder resistors from V_{REF} and thus reduce power consumption.

(3) Comparator and control circuit
The comparator and control circuit compare an analog input voltage with a comparison voltage then store the result in the A-D conversion register. When A-D conversion is complete, the control circuit sets the A-D conversion completion bit and the A-D conversion interrupt request bit to "1".



Note: The M37470M2, M37470M4, and M37470M8 do not have the $P2_4/IN_4$ to $P2_7/IN_7$ pins.

Fig.2.8.2 Structure of A-D control register

2.8.2 Method of use

The A-D conversion method is described below.

(1) Clear the A-D conversion interrupt request bit of the interrupt request register 1 to "0" (the A-D conversion interrupt request bit is not cleared automatically, even when A-D conversion starts).

(2) If using A-D conversion interrupts, enable interrupts by setting the A-D conversion interrupt enable bit to "1" and clearing the interrupt disable flag to "0".

(3) Set the VREF connection selection bit of the A-D control register to "1", to connect VREF to the ladder resistors.

(4) Select analog input pins by setting the analog input selection bits of the A-D control register.

(5) Clear the A-D conversion completion bit of the A-D control register to "0". This write operation starts the A-D conversion. Remember not to read the A-D conversion register during the A-D conversion.

(6) Verify the completion of the conversion from the status of the A-D conversion completion bit, the status of the A-D conversion interrupt request bit, or the presence of an A-D conversion interrupt.

(7) Read the A-D conversion register to obtain the conversion result.

Note: If the ladder resistors are disconnected from VREF, clear the VREF connection selection bit between steps 6 and 7.

2.8.3 Operation

A-D conversion starts when "0" is written to the A-D conversion completion bit. Operations within the M37470 during the A-D conversion are described below.

(1) When A-D conversion starts, the A-D conversion register is cleared to "00_{16}".

(2) Next, the most significant bit of the A-D conversion register is set to "1", and the comparison voltage V_{ref} is input to the comparator. At this point, the analog input voltage V_{IN} is compared with V_{ref}.

(3) If the result of the comparison is $V_{ref} < V_{IN}$, the most significant bit of the A-D converter register remains at "1" as set. If $V_{ref} > V_{IN}$, the most significant bit is cleared to "0".

The above steps are repeated down to the least significant bit, to convert the analog value into a digital value. The A-D conversion ends 50 machine cycles (25μs, when $f(X_{IN})$ = 4MHz) after it starts, and the conversion result is stored in the A-D conversion register.
An A-D conversion interrupt request is generated at the same time that the A-D conversion ends, and the A-D conversion interrupt request bit is set to "1". The A-D conversion completion bit is also set to "1".

Relationship between V_{ref} and V_{REF}

When n = 0 — $V_{ref} = 0$

When n = 1 to 255 — $V_{ref} = V_{REF} / 256 \times (n - 0.5)$

Where n is the value in the A-D conversion register (decimal notation).

Table 2.8.1 Change of A-D conversion register during A-D conversion

	Change of A-D conversion register	Comparison voltage (V_{ref}) value
Conversion start	0 0 0 0 0 0 0 0	0
1st comparison	1 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{512}$
2nd comparison	*1 1 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{512}$
3rd comparison	*1 *2 1 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{512}$
:	:	:
After 8th comparison	Result of A-D conversion *1 *2 *3 *4 *5 *6 *7 *8	

*1: 1st comparison result *2: 2nd comparison result
*3: 3rd comparison result *4: 4th comparison result
*5: 5th comparison result *6: 6th comparison result
*7: 7th comparison result *8: 8th comparison result

2.8.4 Definition of A-D conversion precision

A-D conversion precision is defined below (see Figure 2.8.3).

(1) Relative precision

- Zero transition error (V_{OT})

The zero transition error is the variation of the input voltage when the A-D converter output data changes from "0" to "1", from the ideal A-D converter characteristic between 0V and V_{REF}.

$$V_{OT} = (V_0 - \frac{1}{2} \times \frac{V_{REF}}{512}) / 1LSB_R \ [LSB]$$

- Full-scale transition error (V_{FST})

The full-scale transition error is the variation of the input voltage when the A-D converter output data changes from "255" to "254", from the ideal A-D converter characteristic between 0V and V_{REF}.

$$V_{FST} = \{(V_{REF} - \frac{3}{2} \times \frac{V_{REF}}{256}) - V_{254}\} / 1LSB_R \ [LSB]$$

- Non-linearity error

The non-linearity error is the variation of the actual A-D conversion characteristic, from the ideal characteristic between V_0 and V_{254}.

$$\text{Non-linearity error} = \{V_n - (1LSB_R \times n + V_0)\} / 1LSB_R \quad [LSB]$$

- Differential non-linearity error

The differential non-linearity error is the variation of the input voltage required to change the output data by "1", from the ideal characteristic between V_0 and V_{254}.

$$\text{Differential non-linearity error} = \{(V_{n+1} - V_n) - 1LSB_R\} / 1LSB_R \quad [LSB]$$

(2) Absolute precision

- Absolute precision

The absolute precision is the variation of the actual A-D conversion characteristic, from the ideal characteristic between 0V and V_{REF}.

$$\text{Absolute precision} = \{V_n - 1LSB_A \times (n + \frac{1}{2})\} / 1LSB_A \quad [LSB]$$

V_n: Analog input voltage when output data varies from n to (n+1), where n = 0 to 254

- $1LSB_R = \frac{V_{254} - V_0}{254}$ (V) → 1 LSB with respect to relative precision
- $1LSB_A = \frac{V_{REF}}{256}$ (V) → 1 LSB with respect to absolute precision

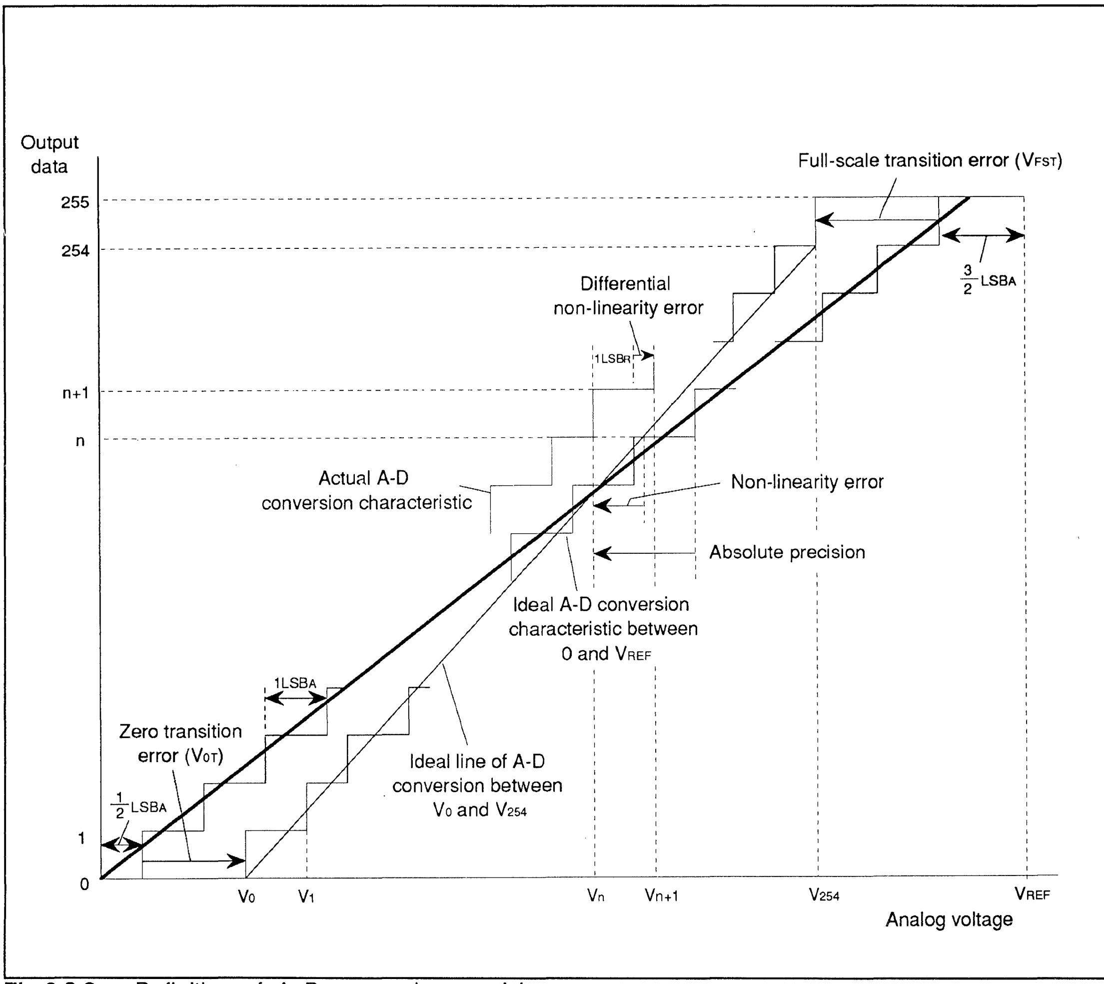
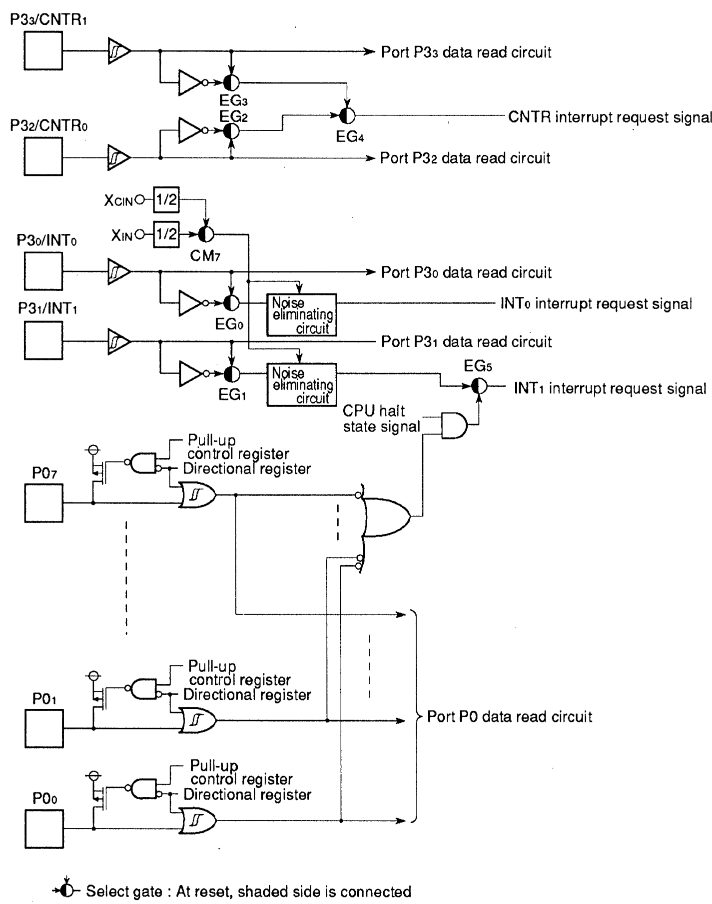


Fig.2.8.3 Definition of A–D conversion precision

2.9 Key on wake up

A key on wake up interrupt is one of the various methods available of recovering from low power consumption modes in the M37470.
To return to the normal operating status from a low power consumption mode, generate an interrupt by applying an “L” level voltage to any of the port P0 pins after bit 5 of the active edge selection register has been set to “1”. This means that the user can create an active“L” key matrix for input to the port P0 pins, so that the return to normal status can be activated by pressing a key. The interrupt vector of the key on wake up interrupt is common with that of the INT$_1$ interrupt. Select the key on wake up interrupt by setting bit 5 of the active edge selection register to “1”. If this bit is “1” in a non low power consumption mode, both INT$_1$ interrupts and key on wake up interrupts are invalid.
If activating a low power consumption mode by the STP or WIT instruction when the interrupt disabled flag is “0” and also bit 5 of the active edge selection register is “1”, set all the inputs to the port P0 pins to “H”.
For details of the low power consumption modes, see “2.12 Low power consumption modes”.



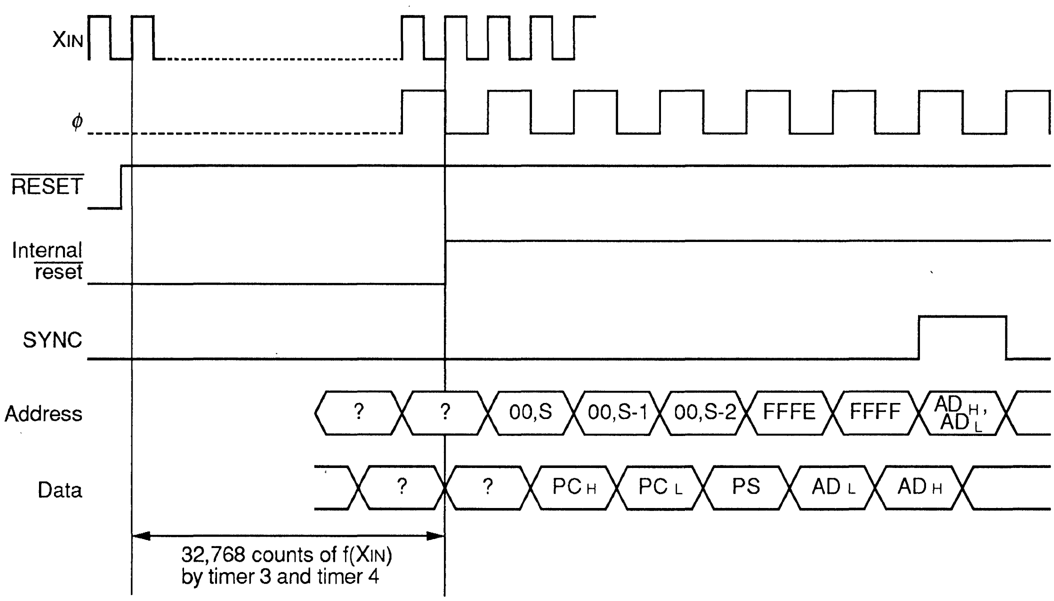
M37470M2, M37470M4, and M37470M8 do not have XCIN input.

Fig.2.9.1 Block diagram of interrupt input and key on wake up circuit

2.10 Reset circuit

2.10.1 Reset operation

If the $\overline{RESET}$ pin is held at "L" level for 2µs then returned to "H", while the supply voltage is within the recommended range, the reset is released in the sequence shown in Figure 2.10.1. The M37470 then starts executing the program from the address whose high-order byte is the contents of address $FFFF_{16}$ and whose low-order byte is the contents of address $FFFE_{16}$. To ensure that the time necessary for the oscillator to stabilize after a reset is generated, timer 3 and timer 4 are connected, and "FF_{16}" is set in timer 3 and "07_{16}" is set in timer 4. In this state, the timers count the oscillation frequency $f(X_{IN})$ divided by 16 and, when timer 4 overflows, reset is released.



Note 1. Normally, the relationship between X_{IN} and frequency of ϕ is $f(X_{IN})=2 \cdot f(\phi)$

Note 2. Question marks (?) indicate a process which is unknown because it depends on the preceding status.

Fig.2.10.1 Timing diagram at reset

2.10.2 Internal status of microcomputer at reset release

The internal status of the M37470 when reset is released is shown in Figure 2.10.2. The contents of all registers and RAM not shown in the figure are undetermined after a reset, so initialize them.

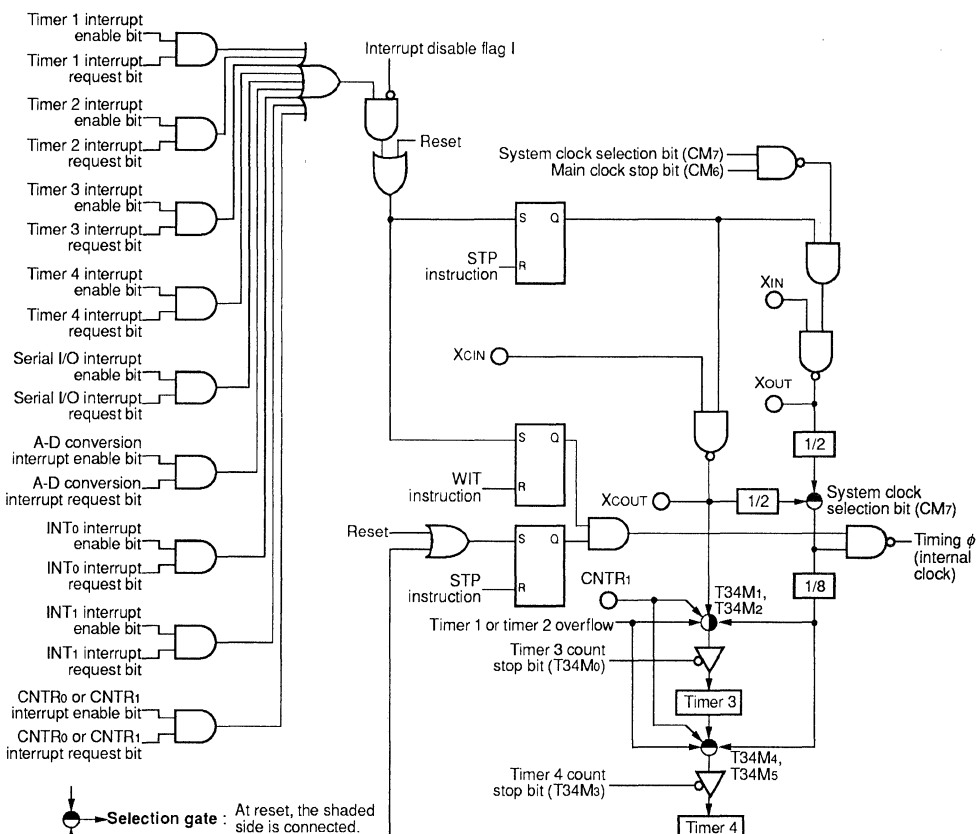
No.	Register	Address								
(1)	Port P0 directional register	($C1_{16}$) ···	00_{16}							
(2)	Port P1 directional register	($C3_{16}$) ···	00_{16}							
(3)	Port P2 directional register	($C5_{16}$) ···	00_{16}							
(4)	Port P4 directional register	($C9_{16}$) ···					0	0	0	0
(5)	Port P0 pull-up control register	($D0_{16}$) ···	00_{16}							
(6)	Port P1~P5 pull-up control register	($D1_{16}$) ···		0		0	0	0	0	0
(7)	Edge selection register	(EG) ($D4_{16}$) ···			0	0	0	0	0	0
(8)	A-D control register	($D9_{16}$) ···	0			0	1	0	0	0
(9)	Serial I /O mode register	(SM) (DC_{16}) ···	00_{16}							
(10)	Timer FF register	($F7_{16}$) ···							—	—
(11)	Timer 12 mode register	(T12M) ($F8_{16}$) ···	00_{16}							
(12)	Timer 34 mode register	(T34M) ($F9_{16}$) ···	00_{16}							
(13)	Timer mode register 2	(TM2) (FA_{16}) ···	0	0					0	0
(14)	CPU mode register	(CM) (FB_{16}) ···	0	0	0	0		0	0	0
(15)	Interrupt request register 1	(FC_{16}) ···	0	0			0	0	0	0
(16)	Interrupt request register 2	(FD_{16}) ···						0	0	0
(17)	Interrupt control register 1	(FE_{16}) ···	0	0			0	0	0	0
(18)	Interrupt control register 2	(FF_{16}) ···						0	0	0
(19)	Program counter	(PC_H) ···	Contents of address $FFFF_{16}$							
		(PC_L) ···	Contents of address $FFFE_{16}$							
(20)	Processor status register	(PS) ···	—	—	—	—	—	1	—	—

Fig.2.10.2 Internal status of microcomputer at reset release

2.11 Oscillation circuit

2.11.1 Oscillation circuit

The M37470 has a built-in oscillation circuit that provides the clock necessary for operation. This oscillation circuit consists of an oscillation gate that acts as an amplifier providing the gain necessary for oscillation and an oscillation control pre-amplifier block that controls the oscillation.
The M37470M2, M37470M4, and M37470M8 each have one built-in oscillation circuit for the main clock. The M37471M2, M37471M4, and M37471M8 each have two built-in oscillation circuits, one for the main clock and one for the clock-function clock.
The frequency input to the clock input pin X_{IN} is normally divided by two to give the internal clock ϕ, but in the M37471M2, M37471M4, and M37471M8, a signal that is half the frequency input to the clock-function clock input pin X_{CIN} can also be selected. Connect either a ceramic resonator or a crystal oscillator as an external element to the outside of this circuit.
A block diagram of the M37470's clock generating circuit is shown in Figure 2.11.1.



M37470M2, M37470M4, and M37470M8 do not have X_{CIN} input and X_{COUT} output.

Fig.2.11.1 Clock generation circuit block diagram

(1) Oscillation circuit using ceramic resonator or crystal oscillator

Examples of circuits using a ceramic resonator or crystal oscillator are shown in Figures 2.11.2 and 2.11.3. As shown in these figures, an oscillation circuit can be formed by connecting a resonator between X_{IN} (or X_{CIN}) and X_{OUT} (or X_{COUT}). Set the circuit's constants (R_d, C_{IN}, C_{OUT}, etc.) in accordance with the resonator manufacturer's recommended values.

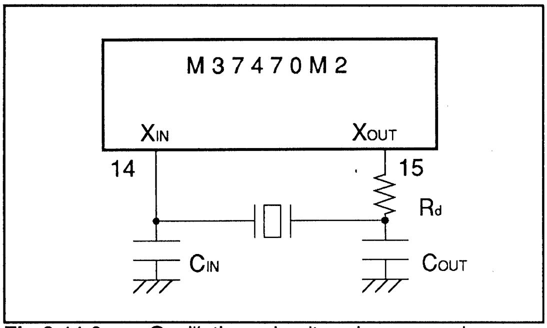


Fig.2.11.2 Oscillation circuit using ceramic resonator or crystal oscillator (M37470)

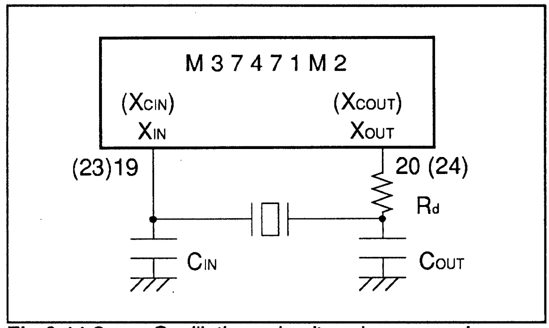


Fig.2.11.3 Oscillation circuit using ceramic resonator or crystal oscillator (M37471)

(2) External clock input circuit

An external clock signal can also be applied to the M37470. Examples of the circuit to be used in this case are shown in Figures 2.11.4 and 2.11.5. Leave the X_{OUT} (or X_{COUT}) pin open in this case.

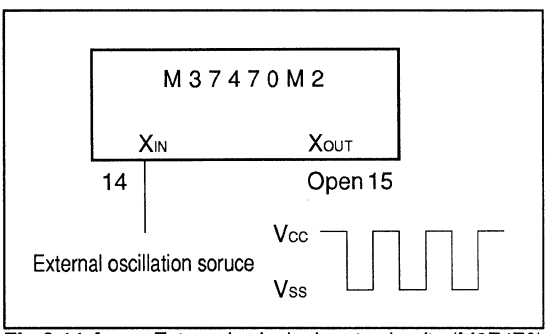


Fig.2.11.4 External clock input circuit (M37470)

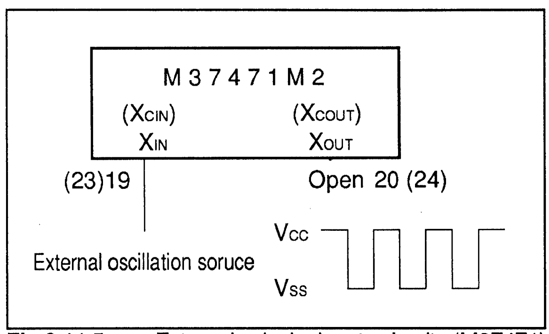
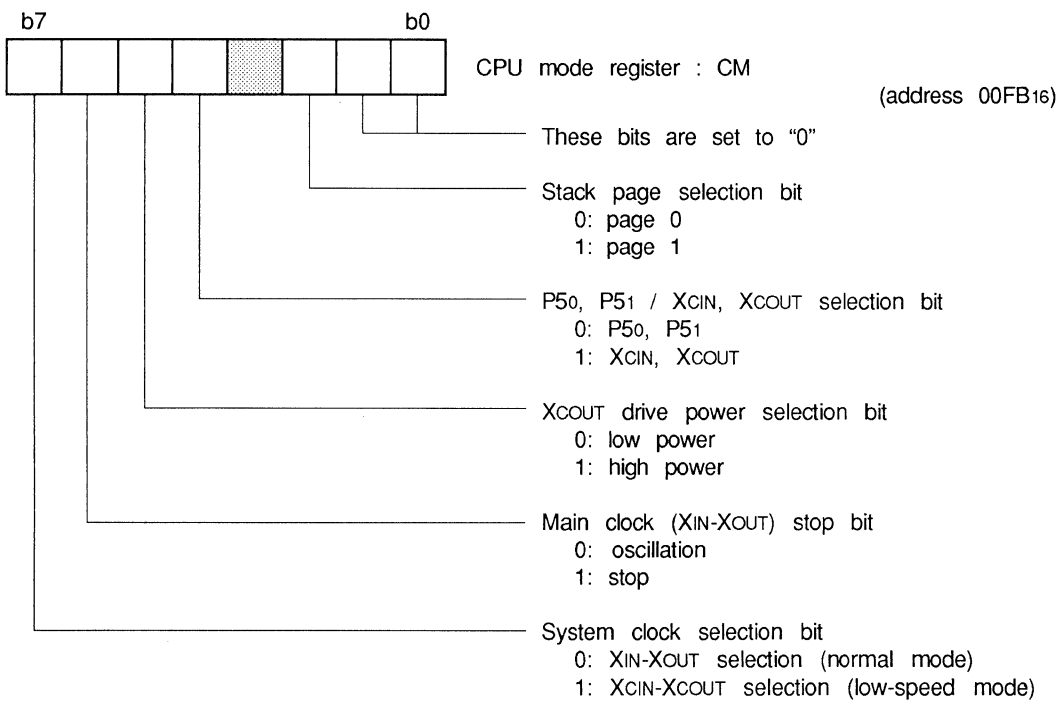


Fig.2.11.5 External clock input circuit (M37471)

2.11.2 Oscillating circuit for clock-function clock

In the M37471M2, M37471M4, and M37471M8, if bit 4 of the CPU mode register (address $00FB_{16}$) is set to "1", the clock-function clock is selected. The structure of the CPU mode register is shown in Figure 2.11.6.



Note: In the M37470M2 and M37470M4, set all of these bits to "0".
In the M37470M8, set all of these bits to "0", except for bit 2.
In the M37471M2 and M37471M4, set bit 2 to "0".

Fig.2.11.6 Structure of CPU mode register

The power source of the clock-function clock oscillation circuit is supplied power via a step-down regulator to reduce the power consumption when the M37470 is operating in clock mode (for details, see "2.12 Low power consumption modes"). In other words, the voltage applied to the V_{CC} pin is reduced by the step-down regulator to reduce power consumption. Bit 5 of the CPU mode register sets the voltage supplied to this oscillation circuit in two steps: high-power mode and low-power mode.

2.12 Low power consumption modes

In the M37470, oscillation can be stopped then restarted if necessary, as described below.

2.12.1 Stop mode

If the STP instruction is executed, oscillation stops with the internal clock ϕ at "H" (stop mode). The functions operating in stop mode are listed in Table 2.12.1.
In stop mode, the contents of all registers except the timer 3 and timer 4 registers are preserved. Therefore, after stop mode is canceled, operations can restart with exactly the same status as that at the point at which the oscillation was stopped, which greatly reduces power consumption.
The internal operation after the STP instruction is executed is as follows:

(1) Oscillation stops with the internal clock ϕ at "H".
(2) Timer 3 is automatically set to "FF_{16}" and timer 4 is automatically set to "07_{16}", and the timer 3 overflow signal is selected as the count source of timer 4. Since the count source of timer 3 is not specified, select a count source that provides the time necessary for the oscillator to stabilize.

Table 2.12.1 Functions operating in stop mode

Timers	Timers with internal count sources are stopped (timer 1 and timer 2 interrupts can be used; timer 3 and timer 4 interrupts cannot be used)
A-D converter	Stopped (A–D conversion interrupts cannot be used) Execute the STP instruction after A–D conversion has ended
Serial I/O	The internal clock is stopped, but serial I/O can operate in external clock mode (serial I/O interrupts can be used)
External interrupt	All INT_0, INT_1, $CNTR_0$, $CNTR_1$ and key on wake up interrupts are valid

To restart oscillation (recover from stop mode), either cause a reset or cause an interrupt to be received. If restart is by interrupt reception, first supply the clock to the timers to start timer 3 and timer 4 operating. When timer 4 overflow is occurred, the internal clock ϕ is supplied. This will provide the time necessary for oscillation to stabilize, if a ceramic or similar resonator is used. (For details, see "2.13 Status transitions".)
Make the following preparations immediately before executing the STP instruction:
(1) Set the timer 3 interrupt enable bit and the timer 4 interrupt enable bit to "0" (disabled), and set the timer 3 count stop bit and the timer 4 count stop bit to "0" (operating).
(2) Set the interrupt used at wake up to interrupt enabled status (set the corresponding interrupt enable bit to "1" and the interrupt disable flag to "0").
(3) Select a count source for timer 3 that provides the time necessary for the oscillator to stabilize. (After recovery, set the previous count source again.)
(4) If using the clock-function clock, set the X_{COUT} drive capability to high power.

2.12.2 Wait mode

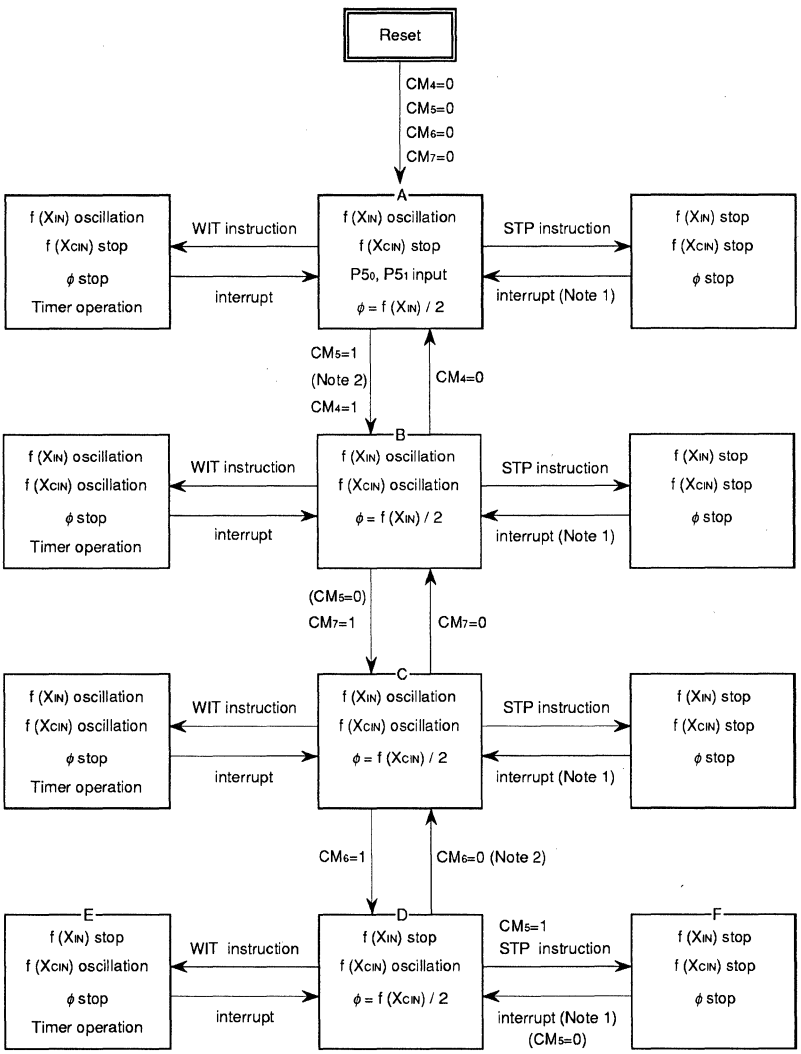
If the WIT instruction is executed, the internal clock ϕ stops at "H", but the oscillator itself does not stop (wait mode). The functions operating in wait mode are listed in Table 2.12.2.
Recovery from wait mode is done in the same way as recovery after the STP instruction, except that, since the oscillator did not stop, there is no need to provide time to enable the oscillation to stabilize—operation can start immediately.

Table 2.12.2 Functions operating in wait mode

Timer	Operating (timer 1, timer 2, timer 3, and timer 4 interrupts can be used)
A-D converter	Operating (A-D conversion interrupts can be used)
Serial I/O	Operating (Serial I/O interrupts can be used)
External interrupt	All INT_0, INT_1, $CNTR_0$, $CNTR_1$ and key on wake up interrupts are valid

2.13 Status transitions

The M37471M2, M37471M4, and M37471M8 are provided with a clock-function clock generating circuit XCIN–XCOUT in addition to the ordinary clock generating circuit XIN–XOUT, so that power consumption can be reduced by leaving these microcomputer with only the clock function operating. These two clocks are controlled by bits 6 and 7 of the CPU mode register (address 00FB16).
Status transitions of the system clock are shown in Figure 2.13.1.



Note 1. Latency time are automatically generated upon release from the STP instruction due to the connections of timer 3 and timer 4.
Note 2. When the system clock is switched over by restarting clock oscillation, a certain wait time required for oscillation to stabilize must be inserted by the program.

Fig.2.13.1 Status transitions

- Reset → normal mode (status A)
  Immediately after a reset, the main clock f(XIN) divided by 2 is selected as the internal clock ϕ, and the I/O pins XCIN and XCOUT of the clock-function clock f(XCIN) are set to be ordinary ports.
  In addition, timer 3 is set to FF_{16} and timer 4 is set to 07_{16}, f(XIN) divided by 16 is selected as the timer 3 count source, the timer 3 overflow signal is set as the timer 4 count source, and the timers start decrementing. When timer 4 overflows, internal reset is released and the program starts from the address specified by the reset vector.

- Low speed mode (status D)
  Transition from normal mode (status A) to low-speed mode in which the clock-function clock f(XCIN) divided by 2 is selected as the internal clock ϕ proceeds via status B and status C. First, to shift to status B, bit 4 of the CPU mode register is set to "1" to set the $P5_0$ and $P5_1$ pins to XCIN and XCOUT. In this case, oscillation start will be facilitated if the XCOUT drive capacity is set to high-power by setting bit 5 of the CPU mode register to "1" as well.
  After the XCIN oscillation has stabilized (provide sufficient time in the program), set bit 7 of the CPU mode register to "1" to shift to status C. Then set bit 6 of the CPU mode register to "1" to stop the oscillation of the main clock f(XIN).
  After the oscillation of the clock-function clock f(XCIN) has stabilized, set bit 5 of the CPU mode register to "0" to set low-power mode for the XCOUT drive capacity, if necessary.
  When the mode shifts from low-speed mode to normal mode, the oscillation of the main clock f(XIN) is started (status C), and f(XIN) becomes the count source for the internal clock ϕ (status B) after the oscillation has stabilized. Then set bit 4 to "0" to set XCIN and XCOUT as ordinary ports, if necessary.

- Clock mode (status E)
  In clock mode, only the clock function operates, to reduce power consumption. Clock mode (status E) is activated by executing the WIT instruction from low-speed mode (status D). When the WIT instruction is executed, the internal clock ϕ stops, and only the supply of the clock to the timers and the serial I/O function continues. Any interrupt will return the system to low-speed mode (status D) from clock mode.
  Wait mode is activated by executing the WIT instruction from any of statuses A to C. Return from wait mode is the same as that from status E, except that the status at return is different.

- Stop mode (status F)
  In stop mode, all statuses in registers, I/O ports, and internal RAM are preserved, except for those of timer 3 and timer 4, and the oscillation of both the main clock and the clock-function clock is stopped.
  Stop mode is activated by executing the STP instruction from any of statuses A to D. For details of settings necessary before the STP instruction is executed, see "2.12 Low power consumption modes".

● Return from stop mode

The reception of any interrupt will return the system from stop mode. When an interrupt is received, timer 3 and timer 4, which start the oscillation in the operating mode set when the STP instruction was executed, first start operating, but the internal clock ϕ is still stopped. When timer 4 overflows after timer 3 and timer 4 started counting, the internal clock ϕ starts operating and, at the same time, execution starts from the processing routine of the received interrupt. The address immediately after the STP instruction is pushed onto the stack as the return address at this point. The reason why the internal clock ϕ does not start operating until timer 4 overflows is to give the oscillation time to stabilize if a ceramic oscillator or a similar oscillator is used.

When the system returns from stop mode, the timer 3 and timer 4 interrupt request bits are set, so it may be necessary to clear them. In addition, the timer 1 and timer 2 interrupt request bits may also be set, depending on setting status, so clear them after return as well, if necessary. If the return is activated by a serial I/O interrupt, remember that this interrupt has a lower priority than timer 1 and timer 2 interrupts, so set timer 1 and timer 2 to either stop status or interrupt disabled status.

If stop mode is released by a reset, the contents of RAM are preserved, but the other registers will have the same status as that after an ordinary reset.

CHAPTER 3

ELECTRIC CHARACTERISTICS

3.1 Electric characteristics of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP

Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage	With respect to V_{SS} Output transistors are at "OFF" state.	–0.3 to 7	V
V_I	Input voltage X_{IN}		–0.3 to V_{CC}+0.3	V
V_I	Input voltage P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P3₀–P3₃, P4₀–P4₁, V_{REF}, $\overline{RESET}$		–0.3 to V_{CC}+0.3	V
V_O	Output voltage P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P4₀–P4₁, X_{OUT}		–0.3 to V_{CC}+0.3	V
Pd	Power dissipation	Ta=25°C	1000	mW
Topr	Operating temperature		–20 to 85	°C
Tstg	Storage temperature		–40 to 150	°C

Recommended operating conditions (V_{CC}=2.7 to 5.5V, V_{SS}=0V, Ta=–20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Supply voltage	2.7	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH}	"H" input voltage P0₀–P0₇, P1₀–P1₇, P3₀–P3₃, $\overline{RESET}$, X_{IN}	0.8V_{CC}		V_{CC}	V
V_{IH}	"H" input voltage P2₀–P2₃, P4₀–P4₁	0.7V_{CC}		V_{CC}	V
V_{IL}	"L" input voltage P0₀–P0₇, P1₀–P1₇, P3₀–P3₃	0		0.2V_{CC}	V
V_{IL}	"L" input voltage P2₀–P2₃, P4₀–P4₁	0		0.25V_{CC}	V
V_{IL}	"L" input voltage X_{IN}	0		0.16V_{CC}	V
V_{IL}	"L" input voltage $\overline{RESET}$	0		0.12V_{CC}	V
I_{OH}(sum)	"H" sum output current P0₀–P0₇, P4₀–P4₁			–30	mA
I_{OH}(sum)	"H" sum output current P1₀–P1₇, P2₀–P2₃			–30	mA
I_{OL}(sum)	"L" sum output current P0₀–P0₇, P4₀–P4₁			60	mA
I_{OL}(sum)	"L" sum output current P1₀–P1₇, P2₀–P2₃			60	mA
I_{OH}(peak)	"H" peak output current P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P4₀–P4₁			–10	mA
I_{OL}(peak)	"L" peak output current P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P4₀–P4₁			20	mA
I_{OH}(avg)	"H" average output current P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P4₀–P4₁ (Note 2)			–5	mA
I_{OL}(avg)	"L" average output current P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P4₀–P4₁ (Note 2)			10	mA
$f_{(CNTR)}$	Timer input frequency CNTR₀ (P3₂), CNTR₁ (P3₃) (Note 1)			1	MHz
$f_{(CLK)}$	Serial I/O clock input frequency CLK(P1₆) (Note 1)			1	MHz
$f(X_{IN})$	Clock oscillating frequency (Note 1)			4	MHz

Note 1 : Oscillation frequency is at 50% duty cycle.
2 : The average output current I_{OH}(avg) and I_{OL}(avg) are the average value during a 100ms.

ELECTRIC CHARACTERISTICS

3.1 Electric characteristics of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP

Electrical characteristics (V_{CC}=2.7 to 5.5V, V_{SS}=0V, Ta=–20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	"H" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$, $P4_1$	V_{CC}=5V, I_{OH}=–5mA		3			V
		V_{CC}=3V, I_{OH}=–1.5mA		2			
V_{OL}	"L" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$, $P4_1$	V_{CC}=5V, I_{OL}=10mA				2	V
		V_{CC}=3V, I_{OL}=3mA				1	
V_{T+}–V_{T-}	Hysteresis $P0_0$–$P0_7$, $P3_0$–$P3_3$	V_{CC}=5V			0.5		V
		V_{CC}=3V			0.3		
V_{T+}–V_{T-}	Hysteresis $\overline{RESET}$	V_{CC}=5V			0.5		V
		V_{CC}=3V			0.3		
V_{T+}–V_{T-}	Hysteresis $P1_6$/CLK	use as CLK input	V_{CC}=5V		0.5		V
			V_{CC}=3V		0.3		
I_{IL}	"L" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_2$, $P4_0$, $P4_1$	V_I=0V, not use pull-up transistor	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
		V_I=0V, use pull-up transistor	V_{CC}=5V	–0.25	–0.5	–1.0	mA
			V_{CC}=3V	–0.08	–0.18	–0.35	
I_{IL}	"L" input current $P3_3$	V_I=0V	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
I_{IL}	"L" input current $P2_0$–$P2_3$	V_I=0V, not use as analog input, not use pull-up transistor	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
		V_I=0V, not use as analog input, use pull-up transistor	V_{CC}=5V	–0.25	–0.5	–1.0	mA
			V_{CC}=3V	–0.08	–0.18	–0.35	
I_{IL}	"L" input current $\overline{RESET}$, X_{IN}	V_I=0V (X_{IN} is at stop mode)	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
I_{IH}	"H" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_2$, $P4_0$, $P4_1$	V_I=V_{CC}, not use pull-up transistor	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $P3_3$	V_I=V_{CC}	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $P2_0$–$P2_3$	V_I=V_{CC}, not use as analog input, not use pull-up transistor	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $\overline{RESET}$, X_{IN}	V_I=V_{CC}, (X_{IN} is at stop mode).	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{CC}	Supply current	At normal operation, A-D conversion is not executed f(X_{IN})=4MHz	V_{CC}=5V		3.5	7	mA
			V_{CC}=3V		1.8	3.6	
		At normal operation, A-D conversion is executed f(X_{IN})=4MHz	V_{CC}=5V		4	8	
			V_{CC}=3V		2	4	
		At wait mode, f(X_{IN})=4MHz	V_{CC}=5V		1	2	
			V_{CC}=3V		0.5	1	
		Stop all oscillation V_{CC}=5V	Ta=25°C		0.1	1	µA
			Ta=85°C		1	10	
V_{RAM}	RAM retention voltage	Stop all oscillation		2			V

ELECTRIC CHARACTERISTICS

3.1 Electric characteristics of M37470M2-XXXSP, M37470M4-XXXSP, and M37470M8-XXXSP

A-D converter characteristics

(V_{CC}=2.7 to 5.5V, V_{SS}=0V, Ta=–20 to 85°C, f(X_{IN})=4MHz, unless otherwise noted)

Symbol	Parameter	Test Conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	bits
—	Non-linearity error				±2	LSB
—	Differential non-linearity error				±0.9	LSB
V_{OT}	Zero transition error	V_{CC}=V_{REF}=5.12V, I_{OL}(sum)=0mA			2	LSB
		V_{CC}=V_{REF}=3.072V, I_{OL}(sum)=0mA			3	
V_{FST}	Full-scale transition error	V_{CC}=V_{REF}=5.12V			4	LSB
		V_{CC}=V_{REF}=3.072V			7	
t_{CONV}	Conversion time				25	µs
V_{VREF}	Reference input voltage		0.5V_{CC}		V_{CC}	V
R_{LADDER}	Ladder resistance value		2	5	10	kΩ
V_{IA}	Analog input voltage		0		V_{REF}	V

3.2 Electric characteristics of M37471M2-XXXSP/FP, M37471M4-XXXSP/FP, and M37471M8-XXXSP/FP

Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		–0.3 to 7	V
V_I	Input voltage X_{IN}		–0.3 to V_{CC}+0.3	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_3$, V_{REF}, $\overline{RESET}$	With respect to V_{SS} Output transistors are at "OFF" state.	–0.3 to V_{CC}+0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$, X_{OUT}		–0.3 to V_{CC}+0.3	V
Pd	Power dissipation	Ta=25°C	1000 (Note 1)	mW
Topr	Operating temperature		–20 to 85	°C
Tstg	Storage temperature		–40 to 150	°C

Note 1 : 500mW for M37471M2-XXXFP, M37471M4-XXXFP, and M37471M8-XXXFP.

Recommended operating conditions (V_{CC}=2.7 to 5.5V, V_{SS}=AV_{SS}=0V, Ta=–20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Supply voltage	2.7	5	5.5	V
V_{SS}	Supply voltage		0		V
AV_{SS}	Analog supply voltage		0		V
V_{IH}	"H" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_3$, $\overline{RESET}$, X_{IN}	0.8V_{CC}		V_{CC}	V
V_{IH}	"H" input voltage $P2_0$–$P2_7$, $P4_0$–$P4_3$, $P5_0$–$P5_3$ (Note 1)	0.7V_{CC}		V_{CC}	V
V_{IL}	"L" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_3$	0		0.2V_{CC}	V
V_{IL}	"L" input voltage $P2_0$–$P2_7$, $P4_0$–$P4_3$, $P5_0$–$P5_3$ (Note 1)	0		0.25V_{CC}	V
V_{IL}	"L" input voltage X_{IN}	0		0.16V_{CC}	V
V_{IL}	"L" input voltage $\overline{RESET}$	0		0.12V_{CC}	V
I_{OH}(sum)	"H" sum output current $P0_0$–$P0_7$, $P4_0$–$P4_3$			–30	mA
I_{OH}(sum)	"H" sum output current $P1_0$–$P1_7$, $P2_0$–$P2_7$			–30	mA
I_{OL}(sum)	"L" sum output current $P0_0$–$P0_7$, $P4_0$–$P4_3$			60	mA
I_{OL}(sum)	"L" sum output current $P1_0$–$P1_7$, $P2_0$–$P2_7$			60	mA
I_{OH}(peak)	"H" peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$			–10	mA
I_{OL}(peak)	"L" peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$			20	mA
I_{OH}(avg)	"H" average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$ (Note 4)			–5	mA
I_{OL}(avg)	"L" average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$ (Note 4)			10	mA
$f_{(CNTR)}$	Timer input frequency $CNTR_0$ ($P3_2$), $CNTR_1$ ($P3_3$) (Note 2)			1	MHz
$f_{(CLK)}$	Serial I/O clock input frequency CLK($P1_6$) (Note 2)			1	MHz
$f(X_{IN})$	Clock oscillating frequency (Note 2)			4	MHz
$f(X_{CIN})$	Clock oscillating frequency for clock function (Note 2, 3)		32	50	kHz

Note 1 : It is except to use $P5_0$ as X_{CIN}.
2 : Oscillation frequency is at 50% duty cycle.
3 : When used in the low-speed mode, the clock oscillating frequency for clock function should be $f(X_{CIN})<f(X_{IN})/3$.
4 : The average output current I_{OH}(avg) and I_{OL}(avg) are the average value during a 100ms.

ELECTRIC CHARACTERISTICS

3.2 Electric characteristics of M37471M2-XXXSP/FP, M37471M4-XXXSP/FP, and M37471M8-XXXSP/FP

Electrical characteristics (V_{CC}=2.7 to 5.5V, V_{SS}=AV_{SS}=0V, T_a=−20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test Conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	"H" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$	V_{CC}=5V, I_{OH}=−5mA		3			V
		V_{CC}=3V, I_{OH}=−1.5mA		2			
V_{OL}	"L" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$	V_{CC}=5V, I_{OL}=10mA				2	V
		V_{CC}=3V, I_{OL}=3mA				1	
V_{T+}–V_{T-}	Hysteresis $P0_0$–$P0_7$, $P3_0$–$P3_3$	V_{CC}=5V			0.5		V
		V_{CC}=3V			0.3		
V_{T+}–V_{T-}	Hysteresis $\overline{RESET}$	V_{CC}=5V			0.5		V
		V_{CC}=3V			0.3		
V_{T+}–V_{T-}	Hysteresis $P1_6$/CLK	use as CLK input	V_{CC}=5V		0.5		V
			V_{CC}=3V		0.3		
I_{IL}	"L" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_2$, $P4_0$–$P4_3$, $P5_0$–$P5_3$	V_I=0V, not use pull-up transistor	V_{CC}=5V			−5	µA
			V_{CC}=3V			−3	
		V_I=0V, use pull-up transistor	V_{CC}=5V	−0.25	−0.5	−1.0	mA
			V_{CC}=3V	−0.08	−0.18	−0.35	
I_{IL}	"L" input current $P3_3$	V_I=0V	V_{CC}=5V			−5	µA
			V_{CC}=3V			−3	
I_{IL}	"L" input current $P2_0$–$P2_7$	V_I=0V, not use as analog input, not use pull-up transistor	V_{CC}=5V			−5	µA
			V_{CC}=3V			−3	
		V_I=0V, not use as analog input, use pull-up transistor	V_{CC}=5V	−0.25	−0.5	−1.0	mA
			V_{CC}=3V	−0.08	−0.18	−0.35	
I_{IL}	"L" input current $\overline{RESET}$, X_{IN}	V_I=0V (X_{IN} is at stop mode)	V_{CC}=5V			−5	µA
			V_{CC}=3V			−3	
I_{IH}	"H" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_2$, $P4_0$–$P4_3$, $P5_0$–$P5_3$	V_I=V_{CC}, not use pull-up transistor	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $P3_3$	V_I=V_{CC}	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $P2_0$–$P2_7$	V_I=V_{CC}, not use as analog input, not use pull-up transistor	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $\overline{RESET}$, X_{IN}	V_I=V_{CC}, (X_{IN} is at stop mode)	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{CC}	Supply current	At normal operation, A-D conversion is not executed $f(X_{IN})$=4MHz	V_{CC}=5V		3.5	7	mA
			V_{CC}=3V		1.8	3.6	
		At normal operation, A-D conversion is executed $f(X_{IN})$=4MHz	V_{CC}=5V		4	8	
			V_{CC}=3V		2	4	
		At low- speed mode, X_{COUT} is low-power mode, A-D conversion is not executed $f(X_{IN})$=0Hz, $f(X_{CIN})$=32kHz, T_a=25°C	V_{CC}=5V		30	80	µA
			V_{CC}=3V		15	40	
		At wait mode, $f(X_{IN})$=4MHz	V_{CC}=5V		1	2	mA
			V_{CC}=3V		0.5	1	
		At wait mode, $f(X_{IN})$=0Hz, $f(X_{CIN})$=32kHz, X_{COUT} is low-power mode, T_a=25°C	V_{CC}=5V		3	12	µA
			V_{CC}=3V		2	8	
		Stop all oscillation V_{CC}=5V	T_a=25°C		0.1	1	
			T_a=85°C		1	10	
V_{RAM}	RAM retention voltage	Stop all oscillation		2			V

ELECTRIC CHARACTERISTICS

3.2 Electric characteristics of M37471M2-XXXSP/FP, M37471M4-XXXSP/FP, and M37471M8-XXXSP/FP

A-D converter characteristics

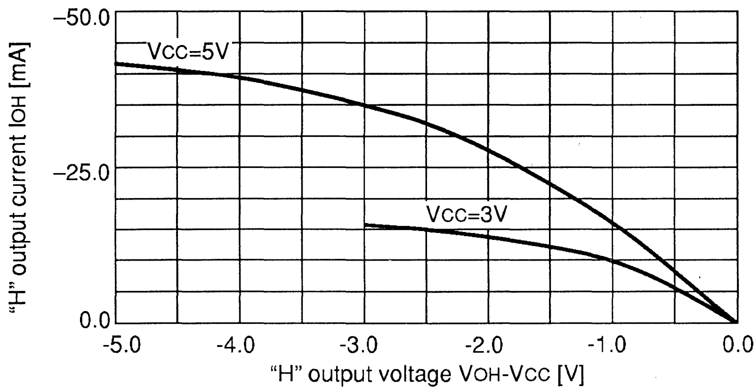
(V_{CC}=2.7 to 5.5V, V_{SS}=AV_{SS}=0V, Ta=–20 to 85°C, f(X_{IN})=4MHz, unless otherwise noted)

Symbol	Parameter	Test Condition	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	bits
—	Non-linearity error				±2	LSB
—	Differential non-linearity error				±0.9	LSB
V_{OT}	Zero transition error	V_{CC}=V_{REF}=5.12V, I_{OL}(sum)=0mA			2	LSB
		V_{CC}=V_{REF}=3.072V, I_{OL}(sum)=0mA			3	
V_{FST}	Full-scale transition error	V_{CC}=V_{REF}=5.12V			4	LSB
		V_{CC}=V_{REF}=3.072V			7	
t_{CONV}	Conversion time				25	µs
V_{VREF}	Reference input voltage		0.5V_{CC}		V_{CC}	V
R_{LADDER}	Ladder resistance value		2	5	10	kΩ
V_{IA}	Analog input voltage		0		V_{REF}	V

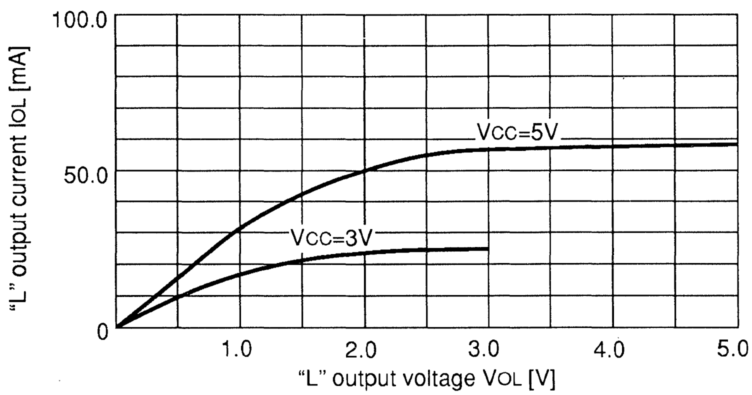
3.3 Standard characteristics

There are the characteristics example in this section. The limitative data are in section 3.1 and 3.2 "Electric characteristics".

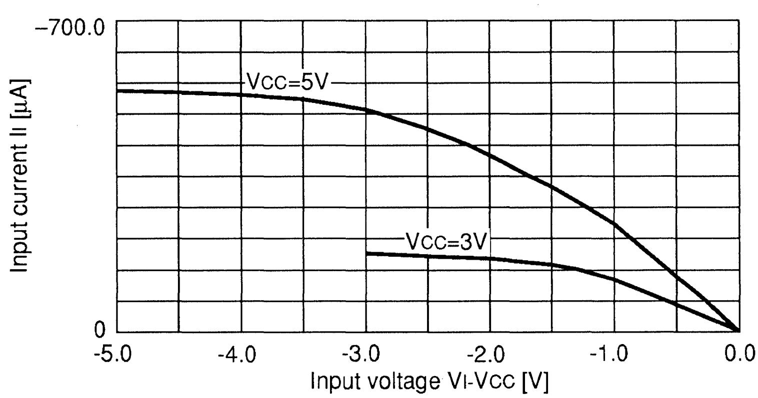
(1) Port P0o (CMOS output) P-channel side IOH-VOH chatacteristics (room tempereture)



(2) Port P0o (CMOS output) N-channel side IOL-VOL characteristics (room tempereture)



(3) Port P0o wirh pull-up transistor VI-II characteristics (room tempereture)



CHAPTER 4

BUILT-IN PROGRAMMABLE ROM VERSION

4.1 Overview

In addition to the mask ROM versions, there are other members of the 7470 Series called built-in programmable ROM versions which are microcomputers with built-in programmable ROM. One version, the window-type EPROM version, has an built-in EPROM that can be written to and erased. Another version is a one-time programmable microcomputer whose built-in PROM can be written to but not erased. Since the functions of the built-in EPROM and one-time programmable versions are exactly the same, apart from whether the ROM contents can be erased, they are both referred to as built-in PROM versions in this manual.
The built-in PROM versions have functions similar to those of the mask ROM versions, but they also have a EPROM mode that enables writing to built-in PROM.
Seven built-in PROM versions of the 7470 Series are available: the M37470E4-XXXSP (one-time programmable), the M37470E8–XXXSP (one-time programmable), the M37471E4-XXXSP/FP (one-time programmable), the M37471E8–XXXSP/FP (one-time programmable), and the M37471E8SS (window version). A brief outline of the specifications of these microcomputers is given in Table 4.1.1.

Table 4.1.1 Functions of built-in PROM versions (M37470E4-XXXSP, M37470E8-XXXSP, M37471E4-XXXSP/FP, M37471E8-XXXSP/FP, and M37471E8SS)

Parameter			Functions		
			M37470E4-XXXSP and M37470E8-XXXSP	M37471E4-XXXSP, M37471E8-XXXSP and M37471E8SS	M37471E4-XXXFP and M37471E8-XXXFP
Basic instructions			69		
Instruction execution time			1.0μs (minimum instructions, at 4MHz)		
Clock frequency			4MHz		
Memory size	PROM (Note 4)		8192 bytes (Note 1)		
	RAM		192 bytes (Note 2)		
Input/Output ports	P0, P1	I/O	8-bit × 2	8-bit × 2	
	P2	I/O	4-bit × 1	8-bit × 1	
	P3	Input	4-bit × 1	4-bit × 1	
	P4	I/O	2-bit × 1	4-bit × 1	
	P5	Input	—	4-bit × 1	
Serial I/O			8-bit × 1		
Timer			8-bit × 4 (with 8-bit latch)		
A-D converter			8-bit × 1 (4-channel)	8-bit × 1 (8-channel)	
Subroutine nesting			96 (max.) (Note 3)		
Interrupt			External 5, internal 6 and software 1		
Clock generating circuit			1 built-in (with external ceramic or quartz crystal oscillator)	2 built-in (with external ceramic or quartz crystal oscillator)	
Power supply			2.7 to 5.5V		
Power dissipation (typ.)			17.5mW (at 4MHz)		
Input/Output characteristics	Input/Output voltage		5V		
	Output current		–5 to 10mA (P0, P1, P2 and P4: CMOS 3-state)		
Operating temperature range			–20 to 85°C		
Device structure			CMOS silicon gate		
Package	One-time programmable		32-pin shrink plastic molded DIP	42-pin shrink plastic molded DIP	56-pin plastic molded QFP
	Window type		—	42-pin shrink ceramic DIP	—

Note 1 : 16384 bytes for M37470E8-XXXSP, M37471E8-XXXSP/FP and M37471E8SS.
2 : 384 bytes for M37470E8-XXXSP, M37471E8-XXXSP/FP and M37471E8SS.
3 : 192 (max.) for M37470E8-XXXSP, M37471E8-XXXSP/FP and M37471E8SS.
4 : Voltage of writing to PROM is 12.5V (corresponding to M5L27256).

4.2 Pin configuration

Figure 4.2.1 shows the pin configuration of M37470E4-XXXSP and M37470E8-XXXSP, Figure 4.2.2 shows the pin configuration of M37471E4-XXXSP, M37471E8-XXXSP and M37471E8SS, and Figure 4.2.3 shows the pin configuration of M37471E4-XXXFP and M37471E8-XXXFP. The built-in PROM versions have pin-compatibility with the mask ROM version.

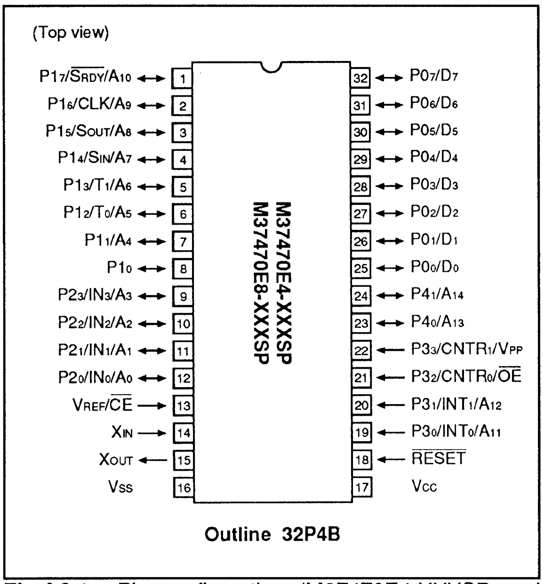


Fig.4.2.1 Pin configuration (M37470E4-XXXSP and M37470E8-XXXSP)

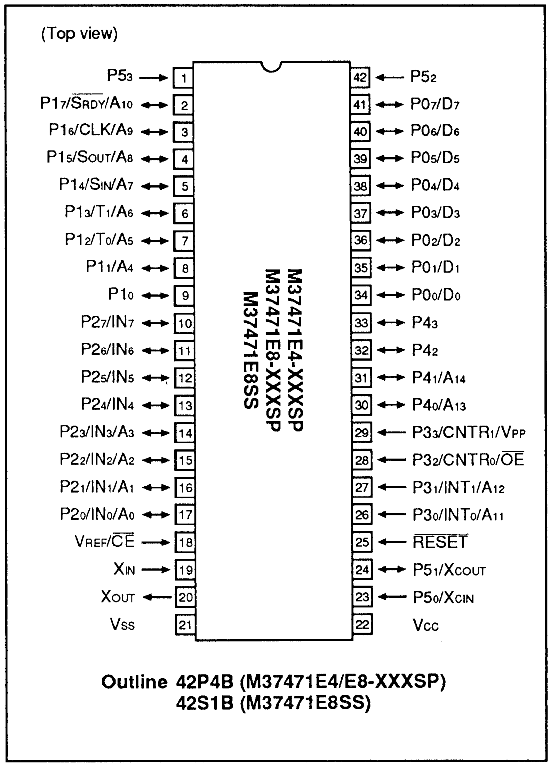


Fig.4.2.2 Pin configuration (M37471E4-XXXSP, M37471E8-XXXSP and M37471E8SS)

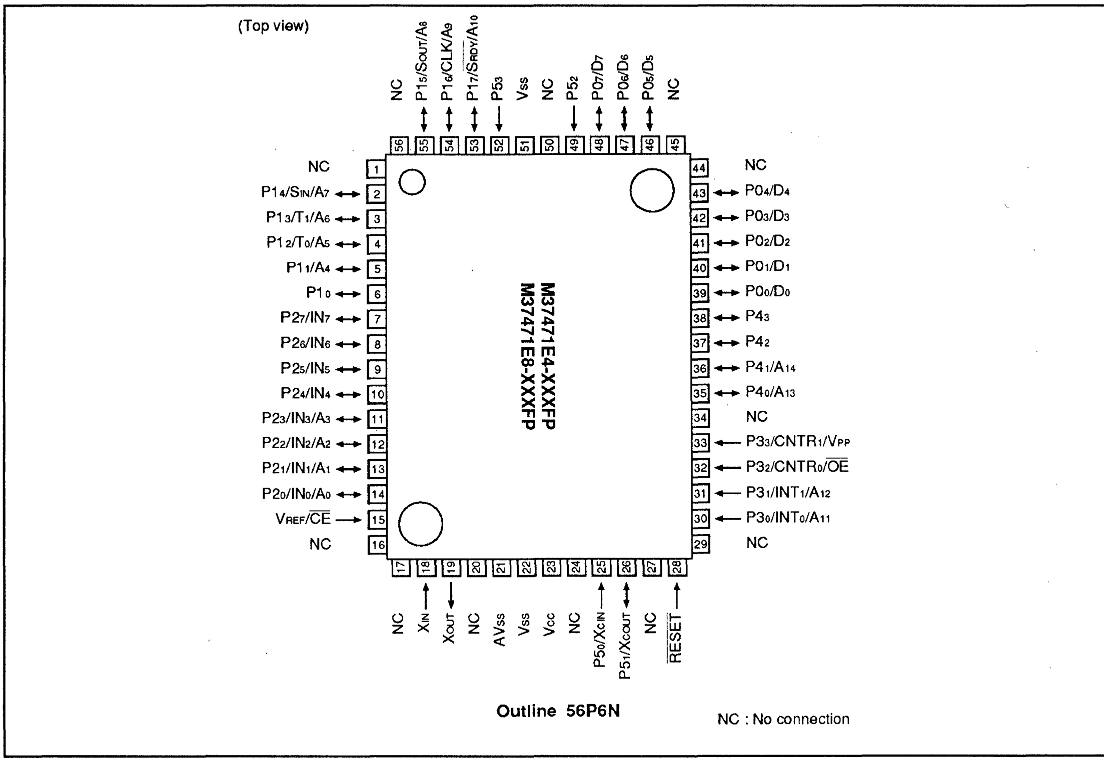


Fig.4.2.3 Pin configuration (M37471E4-XXXFP and M37471E8-XXXFP)

4.3 Block diagram

Figures 4.3.1 to 4.3.3 shows the block diagrams of M37470E4-XXXSP, M37470E8-XXXSP, M37471E4-XXXSP, M37471E8-XXXSP, M37471E8SS, M37471E4-XXXFP, and M37471E8-XXXFP.

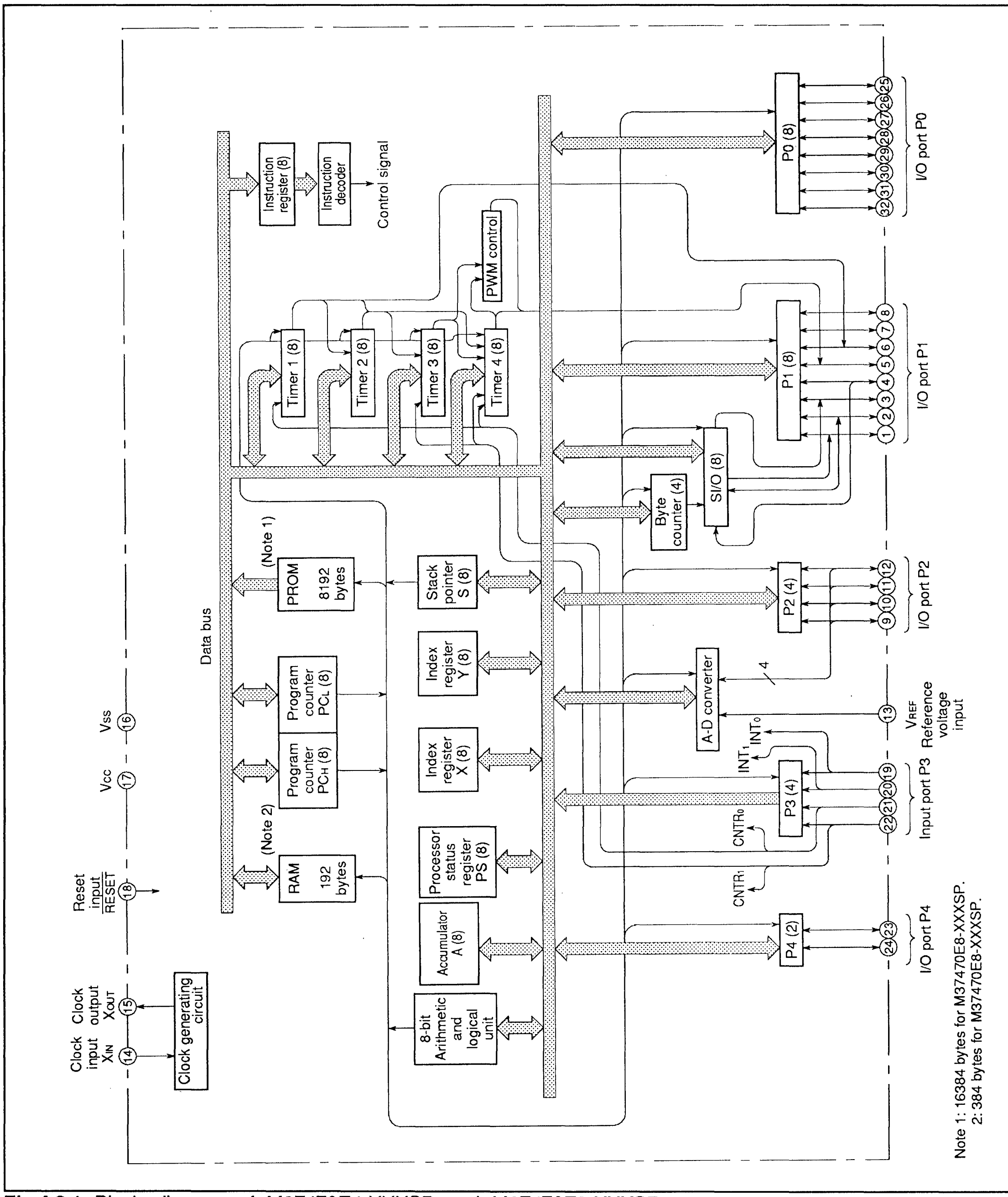
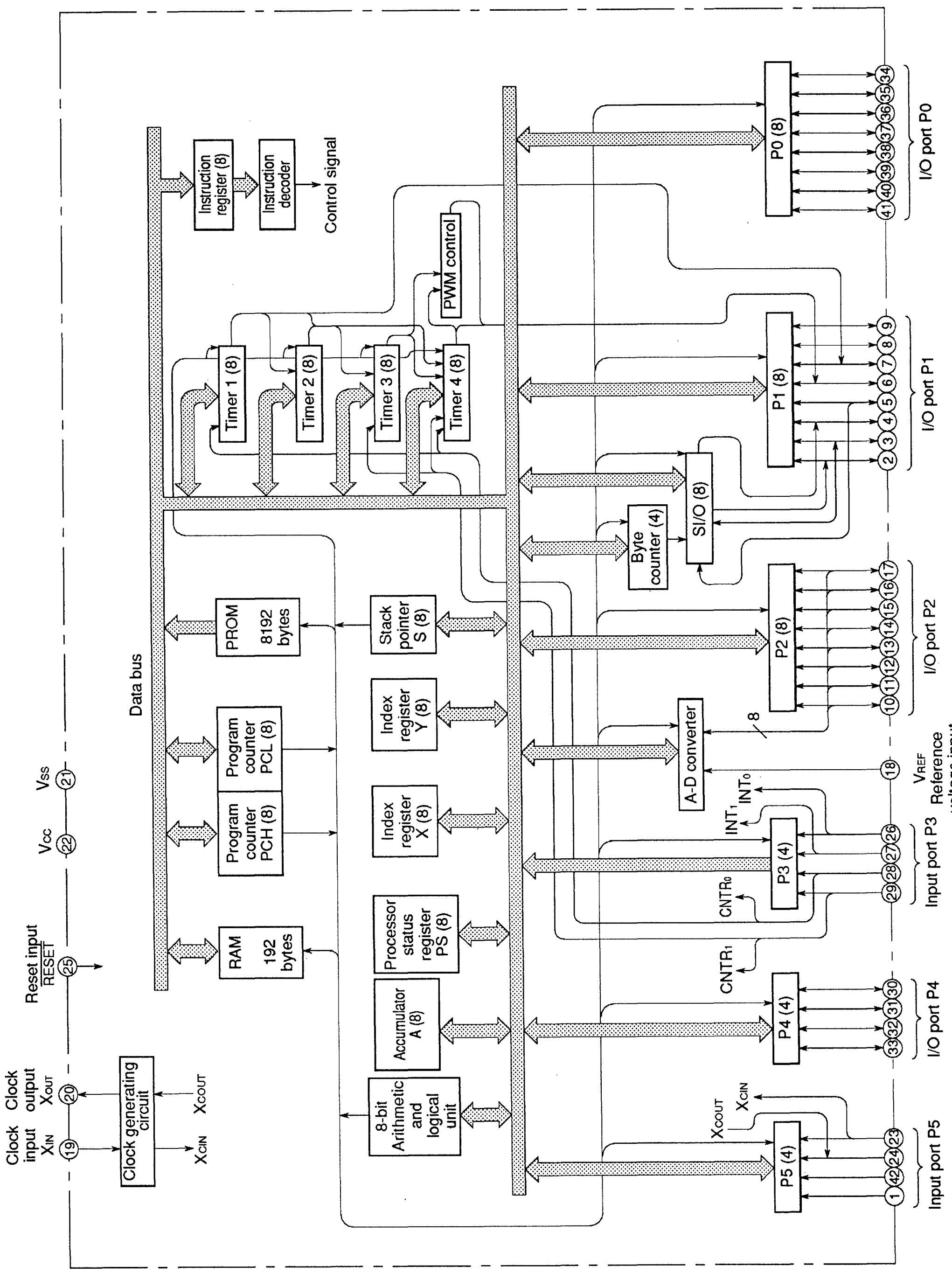
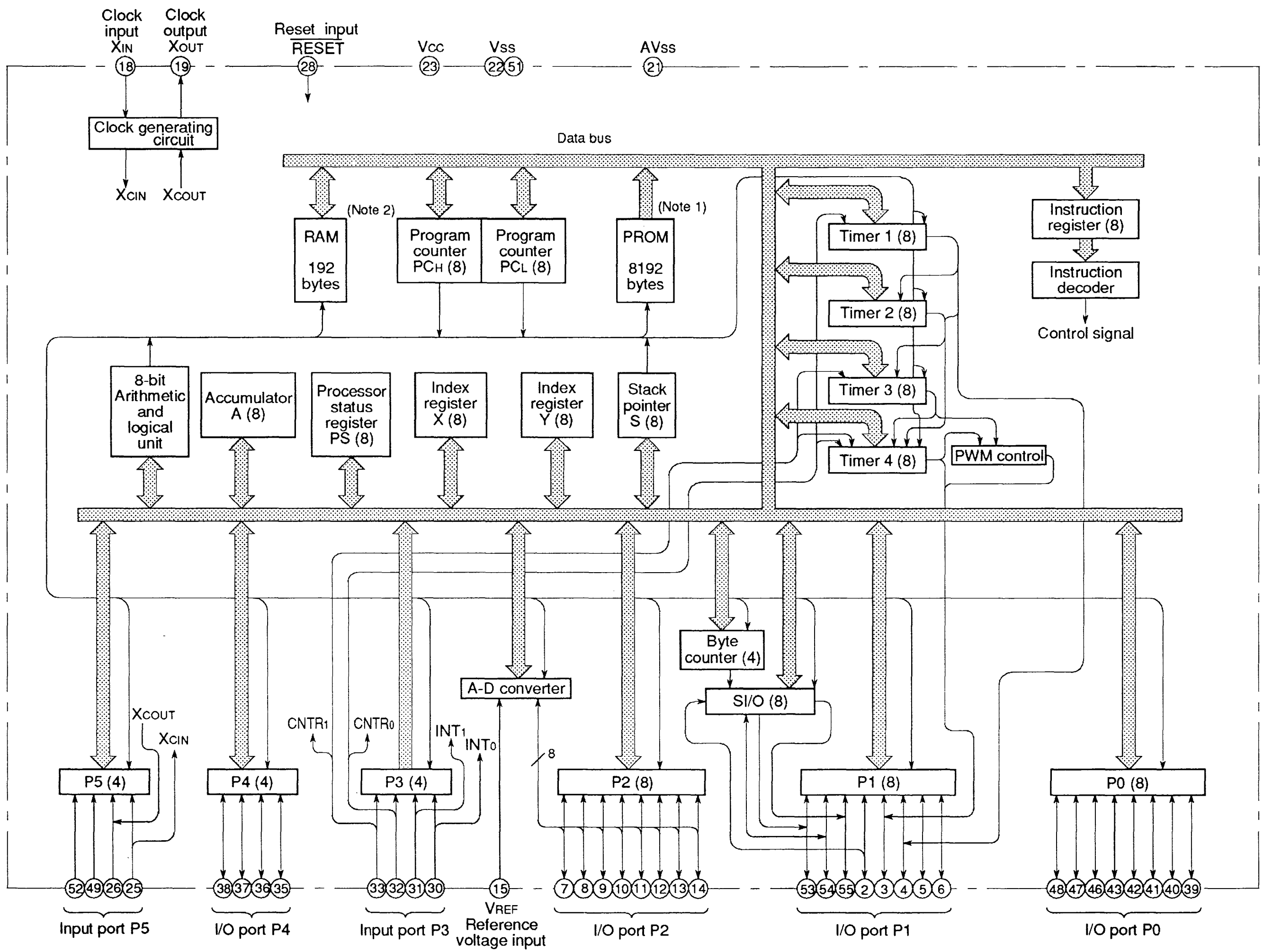


Fig.4.3.1 Block diagram of M37470E4-XXXSP and M37470E8-XXXSP



Note 1: 16384 bytes for M37471E8-XXXSP and M37471E8SS.
2: 384 bytes for M37471E8-XXXSP and M37471E8SS.

Fig.4.3.2 Block diagram of M37471E4-XXXSP and M37471E8-XXXSP(M37471E8SS)



Note 1 : 16384 bytes for M37471E8-XXXFP.
2 : 384 bytes for M37471E8-XXXFP.

Fig.4.3.3 Block diagram of M37471E4-XXXFP and M37471E8-XXXFP

4.4 EPROM mode

4.4.1 EPROM mode

The built-in PROM versions of the 7470 Series have an EPROM mode in addition to the ordinary operating mode. Use EPROM mode to write to, read from, and erase built-in PROM, in the same way as in the M5L27256 (EPROM device).
The pin assignments in EPROM mode are shown in Table 4.4.1, and pin connection diagrams are shown in Figures 4.4.1 to 4.4.3.

Table 4.4.1 Pin correspondence at the EPROM mode

Device type name	Built-in PROM version	M5L27256
Pin name	V_{CC}	V_{CC}
	$P3_3$	V_{PP}
	V_{SS}	V_{SS}
	$P1_1$ to $P1_7$, $P2_0$ to $P2_3$ $P3_0$, $P3_1$, $P4_0$, $P4_1$	A_0 to A_{14}
	$P0_0$ to $P0_7$	D_0 to D_7
	V_{REF}	$\overline{CE}$
	$P3_2$	$\overline{OE}$

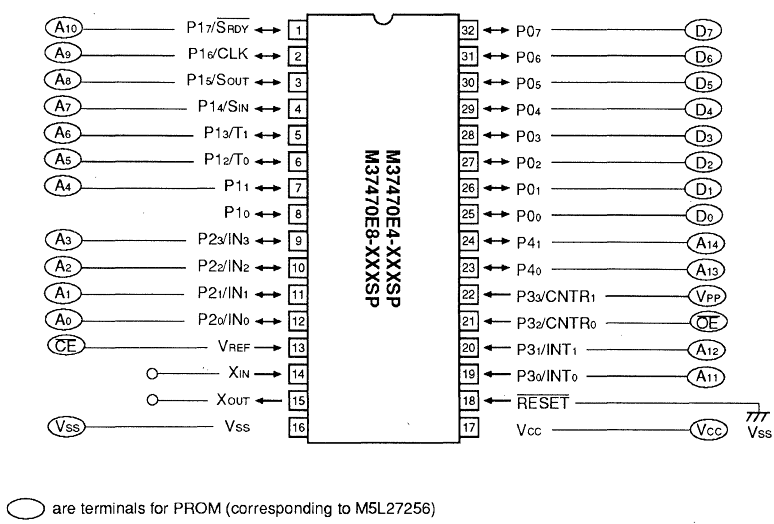


Fig.4.4.1 Pin connection at EPROM mode (M37470E4-XXXSP and M37470E8-XXXSP)

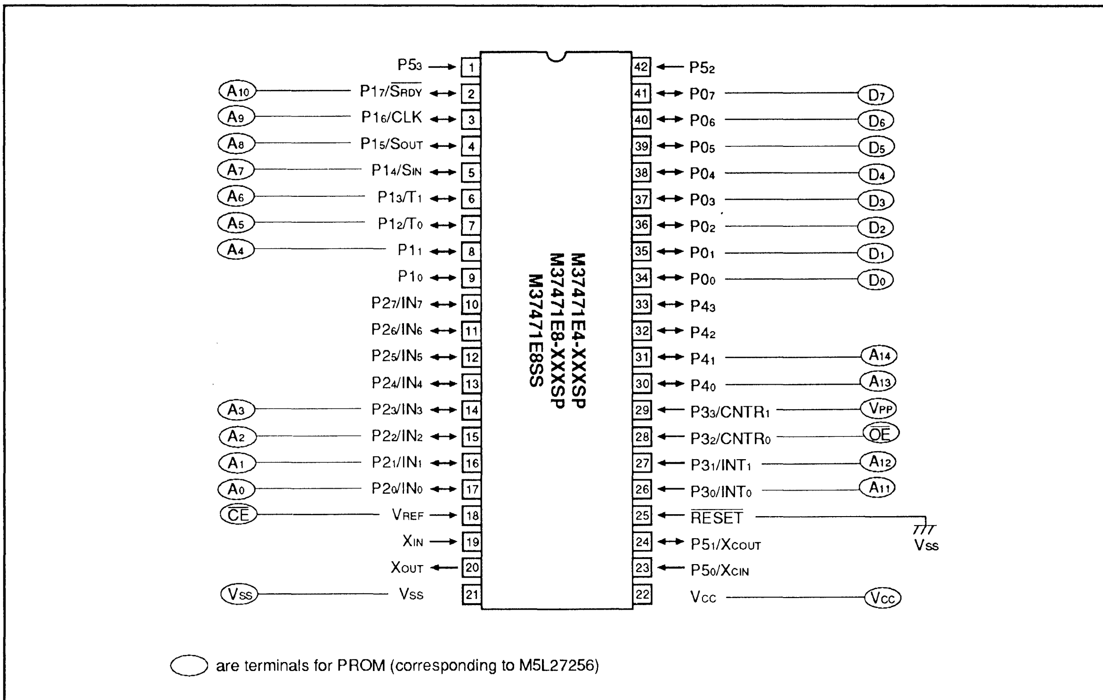


Fig.4.4.2 Pin connection at EPROM mode (M37471E4-XXXSP, M37471E8-XXXSP and M37471E8SS)

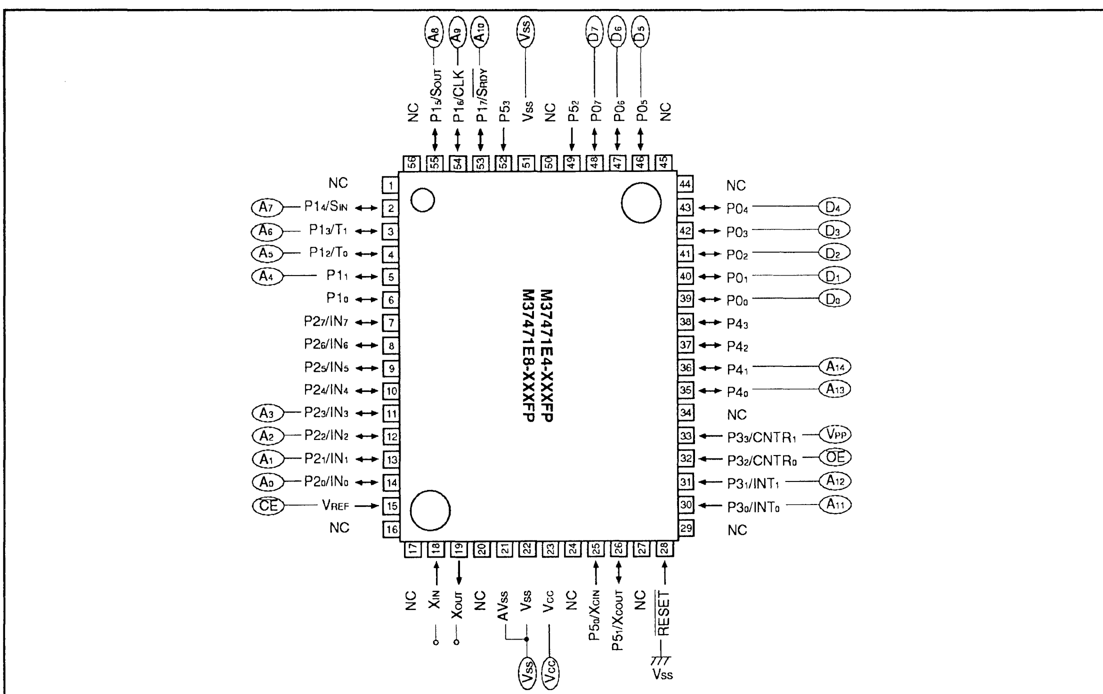


Fig.4.4.3 Pin connection at EPROM mode (M37471E4-XXXFP and M37471E8-XXXFP)

4.4.2 Pin description

Table 4.4.2 shows the pin description at ordinary operating mode and EPROM mode.

Table 4.4.2 Pin description

Pin	Name	Mode	Function
Vcc, Vss	Supply voltage	Ordinary operation/ EPROM	Supply 2.7 to 5.5V to Vcc, and 0V to Vss.
AVss	Analog power supply	Ordinary operation/ EPROM	Acts as ground level input pin for A-D converter. Same voltage as Vss is applied. (Note 1)
VREF	Reference voltage input	Ordinary operation	Acts as reference voltage input pin for the A–D converter.
	Mode input	EPROM	Becomes $\overline{CE}$ input.
$\overline{RESET}$	Reset input	Ordinary operation	Specifies reset when held at "L" for at least 2μs.
		EPROM	Connect to Vss.
XIN	Clock input	Ordinary operation/ EPROM	Acts as input and output pins interfacing with the internal clock generating circuit. Connect a ceramic resonator or crystal oscillator between the XIN and XOUT pins to set the oscillator frequency. An internal feedback resistor is connected between the XIN and XOUT pins. If an external clock is used, connect the clock source to the XIN pin and leave the XOUT pin open.
XOUT	Clock output	Ordinary operation/ EPROM	
$P0_0$–$P0_7$	I/O port P0	Ordinary operation	Acts as 8–bit I/O port with CMOS output format. When input port is selected, these pins can be connected individually to pull-up transistors. A key on wakeup function is also provided.
	Data I/O D_0–D_7	EPROM	Becomes data(D_0–D_7) I/O.
$P1_0$–$P1_7$	I/O port P1	Ordinary operation	Acts as 8–bit I/O port with CMOS output format. When input port is selected, these pins can be connected in groups of four to pull-up transistors. $P1_2$ and $P1_3$ can also be used as timer outputs T_0 and T_1, and $P1_4$, $P1_5$, $P1_6$, and $P1_7$ can also be used as SIN, SOUT, CLK, and $\overline{SRDY}$ of the serial I/O function. SOUT and $\overline{SRDY}$ outputs can be set to N–channel open drain output.
	Address input A_4–A_{10}	EPROM	Pins $P1_1$ to $P1_7$ are address(A_4–A_{10}) input pins. Leave $P1_0$ pin open.
$P2_0$–$P2_7$ (Note 2)	I/O port P2	Ordinary operation	Acts as 8–bit I/O port with CMOS output format. When input port is selected, these pins can be connected in groups of four to pull-up transistors. These pins can also be used as analog inputs IN_0 to IN_7.
	Address input A_0–A_3	EPROM	Pins $P2_0$ to $P2_3$ are address(A_0–A_3) input pins. Leave $P2_4$ to $P2_7$ open.

Note 1: For 56-pin QFP type only.

2: At M37470E4-XXXSP and M37470E8-XXXSP, there are 4-bit as $P2_0$–$P2_3$(IN_0–IN_3) only.

Table 4.4.2 Pin description

Pin	Name	Mode	Function
$P3_0$-$P3_3$	Input port P3	Ordinary operation	Acts as 4–bit input port. $P3_0$ and $P3_1$ can also be used as external interrupt input pins INT_0 and INT_1, and $P3_2$ and $P3_3$ can also be used as timer input pins $CNTR_0$ and $CNTR_1$.
	Address input A_{11}, A_{12} Mode input V_{PP} input	EPROM	Pins $P3_0$ and $P3_1$ are address(A_{11}, A_{12}) input pins. $P3_2$ becomes $\overline{OE}$ input pin. $P3_3$ becomes V_{PP} input pin. At programming and program verifing, supply V_{PP} level into this pin.
$P4_0$-$P4_3$ (Note 2)	I/O port P4	Ordinary operation	Acts as 4–bit I/O port with CMOS output format. When input port is selected, these pins can be connected in groups of four to pull-up transistors.
	Address input A_{13}, A_{14}	EPROM	Pins $P4_0$ and $P4_1$ are address(A_{13}, A_{14}) input pins. Leave pins $P4_2$ and $P4_3$ open.
$P5_0$-$P5_3$ (Note 3)	Input port P5	Ordinary operation	Acts as 4–bit input port that can be connected as a group of four pins to pull-up transistors. $P5_0$ and $P5_1$ can also be used as the X_{CIN} and X_{COUT} pins for the clock-function clock generating circuit. When using these pins as X_{CIN} and X_{COUT} pins, an internal feedback resistor is connected between them. To enable external clock input, connect the clock source to the X_{CIN} pin and leave the X_{COUT} pin open.
		EPROM	Setting to open.

Note 2: At M37470E4-XXXSP and M37470E8-XXXSP, there are 2-bit as $P4_0$ and $P4_1$ only.
3: At M37470E4-XXXSP and M37470E8-XXXSP, there are nothing.

4.4.3 Reading, writing, and erasure of built-in PROM

Activate EPROM mode in the built-in PROM versions by forcing the $\overline{RESET}$ pin to "L". In EPROM mode, the built-in PROM can be read from, written to, and erased, as described below.

(1) Reading
Apply 0V to the $\overline{RESET}$ pin and 5V to the V_{CC} pin.
Input the address signal (A_0 to A_{14}) and set the $\overline{CE}$ and $\overline{OE}$ pins to "L"—the PROM contents will appear at the data I/O pins (D_0 to D_7). If the $\overline{CE}$ pin or the $\overline{OE}$ pin is set to "H", the data I/O pins will float.

(2) Writing
Apply 0V to the $\overline{RESET}$ pin and 6V to the V_{CC} pin.
Set the $\overline{OE}$ pin to "H" and apply V_{PP} to the V_{PP} pin to activate program mode. Set the address to be written to by the address input pins (A_0 to A_{14}) and input the data in parallel through the data I/O pins (D_0 to D_7). When the $\overline{CE}$ pin is set to "L" in this status, the data is written to PROM.

(3) Erasure
Only the built-in EPROM version that has an erasure window on the package's top surface (M37471E8SS) can be erased. To erase the EPROM, shine an ultraviolet light source of wavelength 2537Å onto the window for a minimum dose of 15W·s/cm^2.

Note the following points when writing data with a PROM writer:

- M37470E4-XXXSP, and M37471E4-XXXSP/FP
  When using a PROM writer, the address range should be between 6000_{16} and $7FFF_{16}$. Read/write operations on addresses 0000_{16} to $5FFF_{16}$ cannot be performed correctly.
- M37470E8-XXXSP, M37471E8-XXXSP/FP and M37471E8SS
  When using a PROM writer, the address range should be between 4000_{16} and $7FFF_{16}$. When data is written between addresses 0000_{16} and $7FFF_{16}$, fill addresses 0000_{16} to $3FFF_{16}$ with FF_{16}.

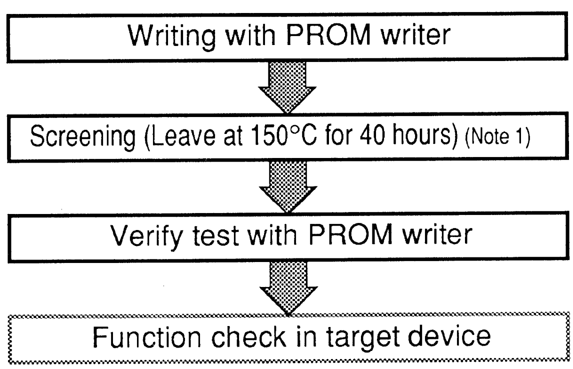
Table 4.4.3 Input/Output signal at each mode

Mode \ Pin name	$\overline{CE}$	$\overline{OE}$	V_{PP}	V_{CC}	$\overline{RESET}$	D_0~D_7
Read	V_{IL}	V_{IL}	V_{CC}	V_{CC}	0V	Output
Output disable	V_{IL}	V_{IH}	V_{CC}			Floating
Write	V_{IL}	V_{IH}	12.5V			Input
Verify	V_{IH}	V_{IL}	12.5V			Output
Write disable	V_{IH}	V_{IH}	12.5V			Floating

Note : V_{IL} means "L" input voltage, V_{IH} means "H" input voltage.

4.4.4 Notes on handling

(1) Sunlight and fluorescent light include wavelengths that will erase written data. When using the window version of the 7470 Series in read mode, always cover the transparent glass window with a light-proof seal.
(2) Mitsubishi provides light-proof seals designed to cover the transparent glass window of the window version. Make sure that the seal does not touch the lead pins of the microcomputer.
(3) Before erasing the window version, clean the transparent glass of the window. Dirt such as grease from hands and glue may hinder the passage of ultraviolet light and affect the erasure characteristics.
(4) Writing involves the use of high voltages, so make sure that excessive voltages are not used. Pay particular attention when turning on the power source.
(5) Mitsubishi does not test or screen any writing to PROM in blank one-time programmable microcomputers*1 after they have left the factory. To improve reliability after writing, we recommend that these microcomputers are written to and tested in the sequence shown in the flow diagram of Figure 4.4.4.
(*1: Blank microcomputers have nothing written in PROM when they leave the factory.)



Note 1: Since the screening temperature is higher than storage temperature, never expose to 150 °C exceeding 100 hours.
Note 2: Function checks in actual devices can also eliminate damage due to surges during handling.

Fig.4.4.4 Writing and test for blank one-time programmable type

4.5 Electric Characteristics

4.5.1 Electric characteristics of M37470E4-XXXSP and M37470E8-XXXSP

Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage	With respect to V_{SS} Output transistors are at "OFF" state.	–0.3 to 7	V
V_I	Input voltage X_{IN}		–0.3 to V_{CC}+0.3	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P3_0$–$P3_3$, $P4_0$–$P4_1$, V_{REF}, $\overline{RESET}$		–0.3 to V_{CC}+0.3 (Note 1)	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$–$P4_1$, X_{OUT}		–0.3 to V_{CC}+0.3	V
Pd	Power dissipation	Ta=25°C	1000	mW
Topr	Operating temperature		–20 to 85	°C
Tstg	Storage temperature		–40 to 150	°C

Note 1: At writing to PROM, the value for $P3_3$ is 13V.

Recommended operating conditions (V_{CC}=2.7 to 5.5V, V_{SS}=0V, Ta=–20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{CC}	Supply voltage	2.7	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH}	"H" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_3$, $\overline{RESET}$, X_{IN}	0.8V_{CC}		V_{CC}	V
V_{IH}	"H" input voltage $P2_0$–$P2_3$, $P4_0$–$P4_1$	0.7V_{CC}		V_{CC}	V
V_{IL}	"L" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_3$	0		0.2V_{CC}	V
V_{IL}	"L" input voltage $P2_0$–$P2_3$, $P4_0$–$P4_1$	0		0.25V_{CC}	V
V_{IL}	"L" input voltage X_{IN}	0		0.16V_{CC}	V
V_{IL}	"L" input voltage $\overline{RESET}$	0		0.12V_{CC}	V
I_{OH}(sum)	"H" sum output current $P0_0$–$P0_7$, $P4_0$–$P4_1$			–30	mA
I_{OH}(sum)	"H" sum output current $P1_0$–$P1_7$, $P2_0$–$P2_3$			–30	mA
I_{OL}(sum)	"L" sum output current $P0_0$–$P0_7$, $P4_0$–$P4_1$			60	mA
I_{OL}(sum)	"L" sum output current $P1_0$–$P1_7$, $P2_0$–$P2_3$			60	mA
I_{OH}(peak)	"H" peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$–$P4_1$			–10	mA
I_{OL}(peak)	"L" peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$–$P4_1$			20	mA
I_{OH}(avg)	"H" average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$–$P4_1$ (Note 3)			–5	mA
I_{OL}(avg)	"L" average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$–$P4_1$ (Note 3)			10	mA
$f_{(CNTR)}$	Timer input frequency $CNTR_0$ ($P3_2$), $CNTR_1$ ($P3_3$) (Note 2)			1	MHz
$f_{(CLK)}$	Serial I/O clock input frequency CLK($P1_6$) (Note 2)			1	MHz
f(X_{IN})	Clock oscillating frequency (Note 2)			4	MHz

Note 2: Oscillation frequency is at 50% duty cycle.
3: The average output current I_{OH}(avg) and I_{OL}(avg) are the average value during a 100ms.

Electrical characteristics (V_{CC}=2.7 to 5.5V, V_{SS}=0V, Ta=–20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test Conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	"H" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$, $P4_1$	V_{CC}=5V, I_{OH}=–5mA		3			V
		V_{CC}=3V, I_{OH}=–1.5mA		2			
V_{OL}	"L" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P4_0$, $P4_1$	V_{CC}=5V, I_{OL}=10mA				2	V
		V_{CC}=3V, I_{OL}=3mA				1	
V_{T+}–V_{T-}	Hysteresis $P0_0$–$P0_7$, $P3_0$–$P3_3$	V_{CC}=5V			0.5		V
		V_{CC}=3V			0.3		
V_{T+}–V_{T-}	Hysteresis $\overline{RESET}$	V_{CC}=5V			0.5		V
		V_{CC}=3V			0.3		
V_{T+}–V_{T-}	Hysteresis $P1_6$/CLK	use as CLK input	V_{CC}=5V		0.5		V
			V_{CC}=3V		0.3		
I_{IL}	"L" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_2$, $P4_0$, $P4_1$	V_I=0V, not use pull-up transistor	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
		V_I=0V, use pull-up transistor	V_{CC}=5V	–0.25	–0.5	–1.0	mA
			V_{CC}=3V	–0.08	–0.18	–0.35	
I_{IL}	"L" input current $P3_3$	V_I=0V	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
I_{IL}	"L" input current $P2_0$–$P2_3$	V_I=0V, not use as analog input, not use pull-up transistor	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
		V_I=0V, not use as analog input, use pull-up transistor	V_{CC}=5V	–0.25	–0.5	–1.0	mA
			V_{CC}=3V	–0.08	–0.18	–0.35	
I_{IL}	"L" input current $\overline{RESET}$, X_{IN}	V_I=0V (X_{IN} is at stop mode)	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
I_{IH}	"H" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_2$, $P4_0$, $P4_1$	V_I=V_{CC}, not use pull-up transistor	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $P3_3$	V_I=V_{CC}	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $P2_0$–$P2_3$	V_I=V_{CC}, not use as analog input, not use pull-up transistor	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $\overline{RESET}$, X_{IN}	V_I=V_{CC}, (X_{IN} is at stop mode)	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{CC}	Supply current	At normal operation, A-D conversion is not executed f(X_{IN})=4MHz	V_{CC}=5V		3.5	7	mA
			V_{CC}=3V		1.8	3.6	
		At normal operation, A-D conversion is executed f(X_{IN})=4MHz	V_{CC}=5V		4	8	
			V_{CC}=3V		2	4	
		At wait mode, f(X_{IN})=4MHz	V_{CC}=5V		1	2	
			V_{CC}=3V		0.5	1	
		Stop all oscillation V_{CC}=5V	Ta=25°C		0.1	1	µA
			Ta=85°C		1	10	
V_{RAM}	RAM retention voltage	Stop all oscillation		2			V

A-D converter characteristics

(V_{CC}=2.7 to 5.5V, V_{SS}=0V, Ta=–20 to 85°C, f(X_{IN})=4MHz, unless otherwise noted)

Symbol	Parameter	Test Conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	bits
—	Non-linearity error				±2	LSB
—	Differential non-linearity error				±0.9	LSB
V_{OT}	Zero transition error	V_{CC}=V_{REF}=5.12V, I_{OL}(sum)=0mA			2	LSB
		V_{CC}=V_{REF}=3.072V, I_{OL}(sum)=0mA			3	
V_{FST}	Full-scale transition error	V_{CC}=V_{REF}=5.12V			4	LSB
		V_{CC}=V_{REF}=3.072V			7	
t_{CONV}	Conversion time				25	µs
V_{VREF}	Reference input voltage		0.5V_{CC}		V_{CC}	V
R_{LADDER}	Ladder resistance value		2	5	10	kΩ
V_{IA}	Analog input voltage		0		V_{REF}	V

4.5.2 Electric characteristics of M37471E4-XXXSP/FP, M37471E8-XXXSP/FP and M37471E8SS

Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage	With respect to V_{SS} Output transistors are at "OFF" state.	–0.3 to 7	V
V_I	Input voltage X_{IN}		–0.3 to V_{CC}+0.3	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_3$, V_{REF}, $\overline{RESET}$		–0.3 to V_{CC}+0.3 (Note 1)	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$, X_{OUT}		–0.3 to V_{CC}+0.3	V
Pd	Power dissipation	Ta=25°C	1000(Note 2)	mW
Topr	Operating temperature		–20 to 85	°C
Tstg	Storage temperature		–40 to 150	°C

Note 1: At writing to PROM, the value for $P3_3$ is 13V.
2: 500mW for M37471E4-XXXFP and M37471E8-XXXFP.

Recommended operating conditions (V_{CC}=2.7 to 5.5V, V_{SS}=AV_{SS}=0V, Ta=–20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{CC}	Supply voltage	2.7	5	5.5	V
V_{SS}	Supply voltage		0		V
AV_{SS}	Analog supply voltage		0		V
V_{IH}	"H" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_3$, $\overline{RESET}$, X_{IN}	0.8V_{CC}		V_{CC}	V
V_{IH}	"H" input voltage $P2_0$–$P2_7$, $P4_0$–$P4_3$, $P5_0$–$P5_3$ (Note 3)	0.7V_{CC}		V_{CC}	V
V_{IL}	"L" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_3$	0		0.2V_{CC}	V
V_{IL}	"L" input voltage $P2_0$–$P2_7$, $P4_0$–$P4_3$, $P5_0$–$P5_3$ (Note 3)	0		0.25V_{CC}	V
V_{IL}	"L" input voltage X_{IN}	0		0.16V_{CC}	V
V_{IL}	"L" input voltage $\overline{RESET}$	0		0.12V_{CC}	V
I_{OH}(sum)	"H" sum output current $P0_0$–$P0_7$, $P4_0$–$P4_3$			–30	mA
I_{OH}(sum)	"H" sum output current $P1_0$–$P1_7$, $P2_0$–$P2_7$			–30	mA
I_{OL}(sum)	"L" sum output current $P0_0$–$P0_7$, $P4_0$–$P4_3$			60	mA
I_{OL}(sum)	"L" sum output current $P1_0$–$P1_7$, $P2_0$–$P2_7$			60	mA
I_{OH}(peak)	"H" peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$			–10	mA
I_{OL}(peak)	"L" peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$			20	mA
I_{OH}(avg)	"H" average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$ (Note 6)			–5	mA
I_{OL}(avg)	"L" average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$ (Note 6)			10	mA
$f_{(CNTR)}$	Timer input frequency $CNTR_0$ ($P3_2$), $CNTR_1$ ($P3_3$) (Note 4)			1	MHz
$f_{(CLK)}$	Serial I/O clock input frequency CLK($P1_6$) (Note 4)			1	MHz
f(X_{IN})	Clock oscillating frequency (Note 4)			4	MHz
f(X_{CIN})	Clock oscillating frequency for clock function (Note 4, 5)		32	50	kHz

Note 3: It is except to use $P5_0$ as X_{CIN}.
4: Oscillation frequency is at 50% duty cycle.
5: When used in the low-speed mode, the clock oscillating frequency for clock function should be f(X_{CIN})<f(X_{IN})/3.
6: The average output current I_{OH}(avg) and I_{OL}(avg) are the average value during a 100ms.

Electrical characteristics (V_{CC}=2.7 to 5.5V, V_{SS}=AV_{SS}=0V, Ta=–20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test Conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	"H" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$	V_{CC}=5V, I_{OH}=–5mA		3			V
		V_{CC}=3V, I_{OH}=–1.5mA		2			
V_{OL}	"L" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$–$P4_3$	V_{CC}=5V, I_{OL}=10mA				2	V
		V_{CC}=3V, I_{OL}=3mA				1	
V_{T+}–V_{T-}	Hysteresis $P0_0$–$P0_7$, $P3_0$–$P3_3$	V_{CC}=5V			0.5		V
		V_{CC}=3V			0.3		
V_{T+}–V_{T-}	Hysteresis $\overline{RESET}$	V_{CC}=5V			0.5		V
		V_{CC}=3V			0.3		
V_{T+}–V_{T-}	Hysteresis $P1_6$/CLK	use as CLK input	V_{CC}=5V		0.5		V
			V_{CC}=3V		0.3		
I_{IL}	"L" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_2$, $P4_0$–$P4_3$, $P5_0$–$P5_3$	V_I=0V, not use pull-up transistor	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
		V_I=0V, use pull-up transistor	V_{CC}=5V	–0.25	–0.5	–1.0	mA
			V_{CC}=3V	–0.08	–0.18	–0.35	
I_{IL}	"L" input current $P3_3$	V_I=0V	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
I_{IL}	"L" input current $P2_0$–$P2_7$	V_I=0V, not use as analog input, not use pull-up transistor	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
		V_I=0V, not use as analog input, use pull-up transistor	V_{CC}=5V	–0.25	–0.5	–1.0	mA
			V_{CC}=3V	–0.08	–0.18	–0.35	
I_{IL}	"L" input current $\overline{RESET}$, X_{IN}	V_I=0V (X_{IN} is at stop mode)	V_{CC}=5V			–5	µA
			V_{CC}=3V			–3	
I_{IH}	"H" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_2$, $P4_0$–$P4_3$, $P5_0$–$P5_3$	V_I=V_{CC}, not use pull-up transistor	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $P3_3$	V_I=V_{CC}	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $P2_0$–$P2_7$	V_I=V_{CC}, not use as analog input, not use pull-up transistor	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{IH}	"H" input current $\overline{RESET}$, X_{IN}	V_I=V_{CC}, (X_{IN} is at stop mode)	V_{CC}=5V			5	µA
			V_{CC}=3V			3	
I_{CC}	Supply current	At normal operation, A-D conversion is not executed f(X_{IN})=4MHz	V_{CC}=5V		3.5	7	mA
			V_{CC}=3V		1.8	3.6	
		At normal operation, A-D conversion is executed f(X_{IN})=4MHz	V_{CC}=5V		4	8	
			V_{CC}=3V		2	4	
		At low- speed mode, X_{COUT} is low-power mode, A-D conversion is not executed f(X_{IN})=0Hz, f(X_{CIN})=32kHz, T_a=25°C	V_{CC}=5V		30	80	µA
			V_{CC}=3V		15	40	
		At wait mode, f(X_{IN})=4MHz	V_{CC}=5V		1	2	mA
			V_{CC}=3V		0.5	1	
		At wait mode, f(X_{IN})=0Hz, f(X_{CIN})=32kHz, X_{COUT} is low-power mode, T_a=25°C	V_{CC}=5V		3	12	µA
			V_{CC}=3V		2	8	
		Stop all oscillation V_{CC}=5V	Ta=25°C		0.1	1	
			Ta=85°C		1	10	
V_{RAM}	RAM retention voltage	Stop all oscillation		2			V

A-D converter characteristics

(V_{CC}=2.7 to 5.5V, V_{SS}=AV_{SS}=0V, Ta=–20 to 85°C, f(X_{IN})=4MHz, unless otherwise noted)

Symbol	Parameter	Test Condition	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	bits
—	Non-linearity error				±2	LSB
—	Differential non-linearity error				±0.9	LSB
V_{OT}	Zero transition error	V_{CC}=V_{REF}=5.12V, I_{OL}(sum)=0mA			2	LSB
		V_{CC}=V_{REF}=3.072V, I_{OL}(sum)=0mA			3	
V_{FST}	Full-scale transition error	V_{CC}=V_{REF}=5.12V			4	LSB
		V_{CC}=V_{REF}=3.072V			7	
t_{CONV}	Conversion time				25	µs
V_{VREF}	Reference input voltage		0.5V_{CC}		V_{CC}	V
R_{LADDER}	Ladder resistance value		2	5	10	kΩ
V_{IA}	Analog input voltage		0		V_{REF}	V

APPENDIX

Appendix 1 Handling of unused pins

Pin name	Handling
$P0_0$ to $P0_7$, $P1_0$ to $P1_3$, $P1_5$, $P1_7$, $P2_0$ to $P2_7$, $P4_0$ to $P4_3$	Either leave open, or pull-up to V_{CC} or pull-down to V_{SS} by resistors. (Note 1, Note 2)
$P1_4$/S_{IN}, $P1_6$/CLK	Pull-up to V_{CC} or pull-down to V_{SS} by resistors. (Note 1, Note 2)
$P3_0$ to $P3_3$	Connect to V_{CC} or V_{SS}.
$P5_0$ to $P5_3$	Either leave open, or pull-up to V_{CC} or pull-down to V_{SS} by resistors. (Note 3)
V_{REF}	Connect to V_{CC}

Note 1: If pulling-up a pin, set the status of that pin to either input or "H" level output.

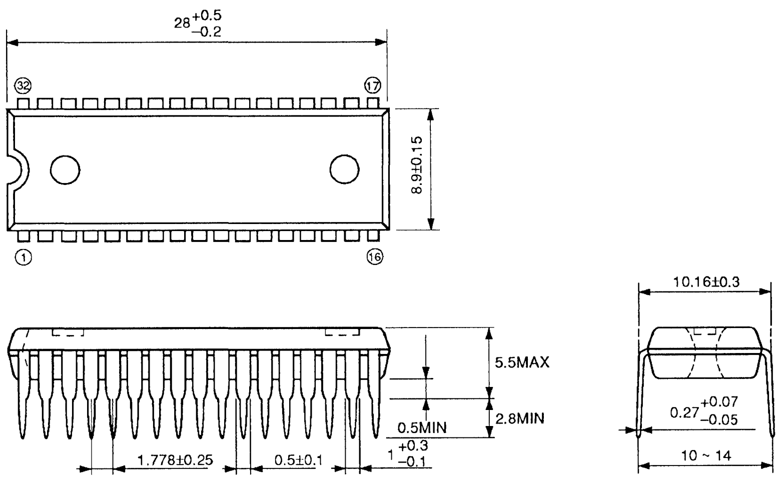
Note 2: If pulling-down a pin, set the corresponding register to make the status of that pin either input without a pull-up transistor or "L" level output.

Note 3: If pulling-down a pin, set the corresponding register to make the status of that pin input without a pull-up transistor. If not using the $P5_0$ and $P5_1$ pins, select ports $P5_0$ and $P5_1$ for them.

Appendix 2 Package outline

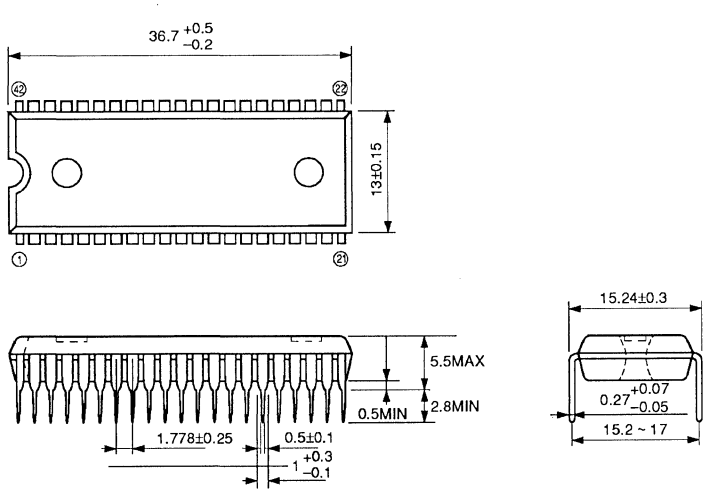
Package outline of M37470M2-XXXSP, M37470M4-XXXSP, M37470E4-XXXSP, M37470M8-XXXSP, and M37470E8-XXXSP

TYPE 32P4B 32-PIN MOLDED PLASTIC DIP (LEAD PITCH 1.778mm) Dimension in mm



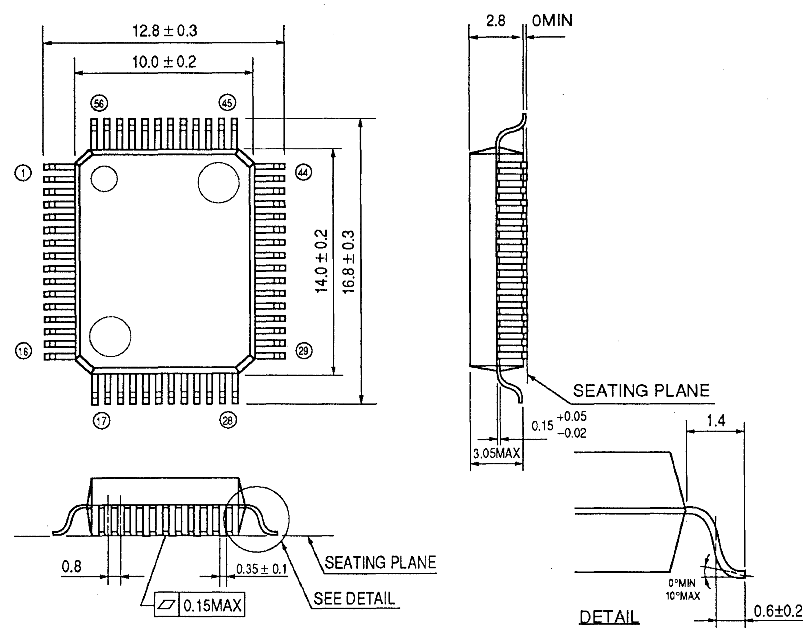
Package outline of M37471M2-XXXSP, M37471M4-XXXSP, M37471E4-XXXSP, M37471M8-XXXSP, and M37471E8-XXXSP

TYPE 32P4B 32-PIN MOLDED PLASTIC DIP (LEAD PITCH 1.778mm) Dimension in mm



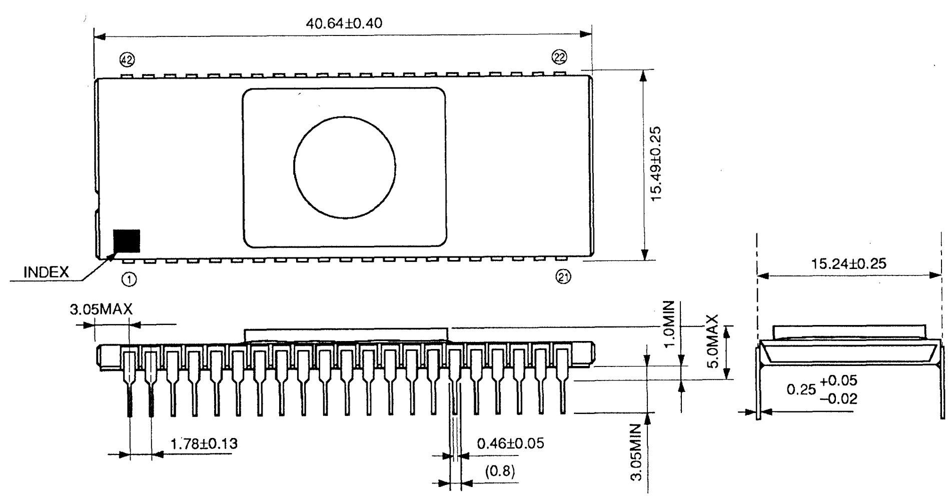
Package outline of M37471M2-XXXFP, M37471M4-XXXFP, M37471E4-XXXFP, M37471M8-XXXFP, and M37471E8-XXXFP

TYPE 56P6N 56-PIN MOLDED PLASTIC QFP (LEAD PITCH 0.8mm) Dimension in mm



Package outline of M37471E8SS

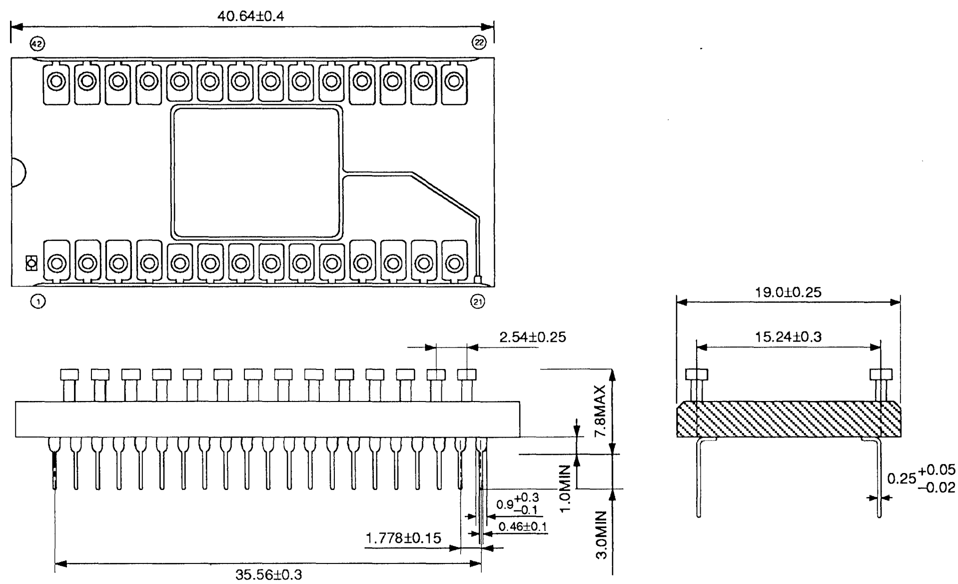
TYPE 42S1B 42-PIN CERAMIC DIP Dimension in mm



Package outline of M37471RSS

TYPE 42S1M 42-PIN PIGGYBACK DIP

Dimension in mm



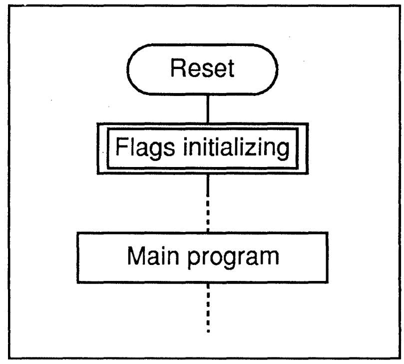
Appendix 3 Notes for usage

Note for using 7470 Series as following.

1. Note about Processor Status Register

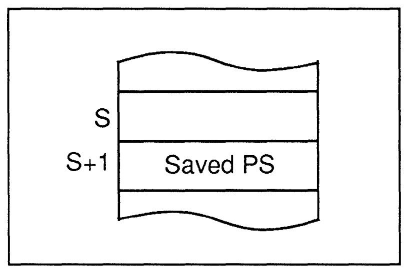
1.1 The processor status register initialization

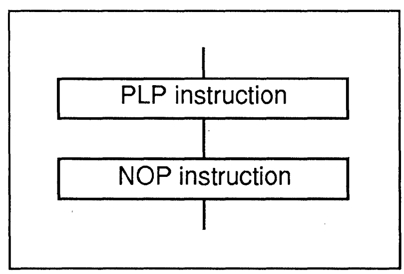
The contents of processor status register(PS) is indeterminate after reset except the I flag. Therefore the flags that influence on execution of program, are required to initialize.
Especially initialize the T flag and D flag are influenced on operation directory.



1.2 How to refer to the processor status register

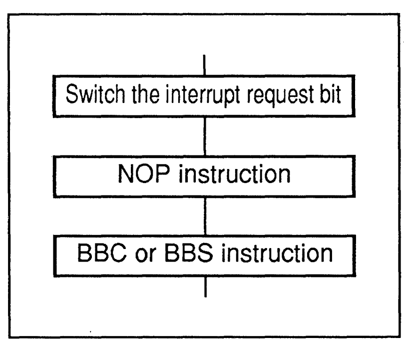
(1) Execute a PHP instruction to store the contents of processor status register(PS) into stack(S+1)
(2) Read from the contents of stack(S+1).
(3) To restore the previous PS from stack, execute a PLP instruction. However, a NOP instruction is needed after execute the PLP instruction.





2. Note for Interrupt Function

More than one instruction cycle is needed to execute the BBC or BBS instruction after switching the value of interrupt request registers(interrupt request register 1: address $00FC_{16}$ and interrupt request register 2: address $00FD_{16}$).



3. Note for Serial I/O Function

3.1 Clear the serial I/O interrupt enable bit

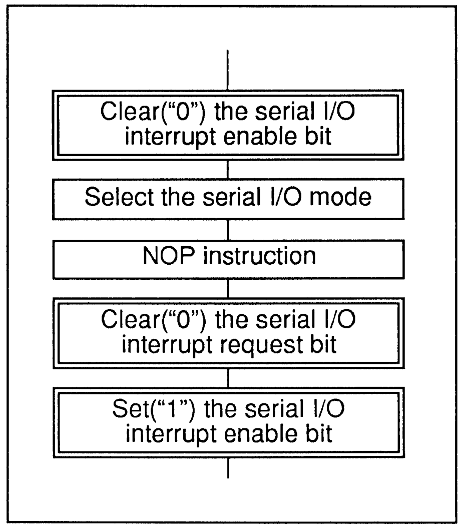
At serial I/O interrupt function, clear(set "0") the serial I/O interrupt enable bit (bit 6 of address $00FE_{16}$) by execution a CLB instruction, before setting serial I/O mode register.

3.2 At clearing the serial I/O interrupt request bit

At serial I/O interrupt function, more than one instruction cycle is needed until clearing (set "0" by a CLB instruction) the serial I/O interrupt request bit (bit 6 of address $00FC_{16}$) after setting the serial I/O mode register.

3.3 Setting the serial I/O interrupt enable bit

It is required to set "1" the serial I/O interrupt enable bit (bit 6 of address $00FE_{16}$), after clearing the serial I/O interrupt request bit.



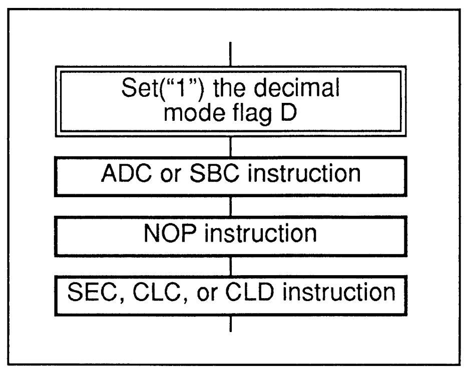
4. Note for Decimal Operation

4.1 Setting and clearing the decimal mode flag

In decimal operation mode(at D flag="1"), more than one instruction cycle is needed before SEC, CLC, or CLD instruction after the ADC or SBC instruction.

4.2 Ignored flags

The N (Negative), V (oVerflow), and Z (Zero) flags are ignored, when executing the ADC and SBC instructions during decimal mode.



5. Note for others

5.1 Timer division ratio

Timer division ratio is 1/(n+1) (where n=0 to 255).

5.2 A-D conversion

(1) A-D conversion is necessary to start after the reference voltage level is enough stability.

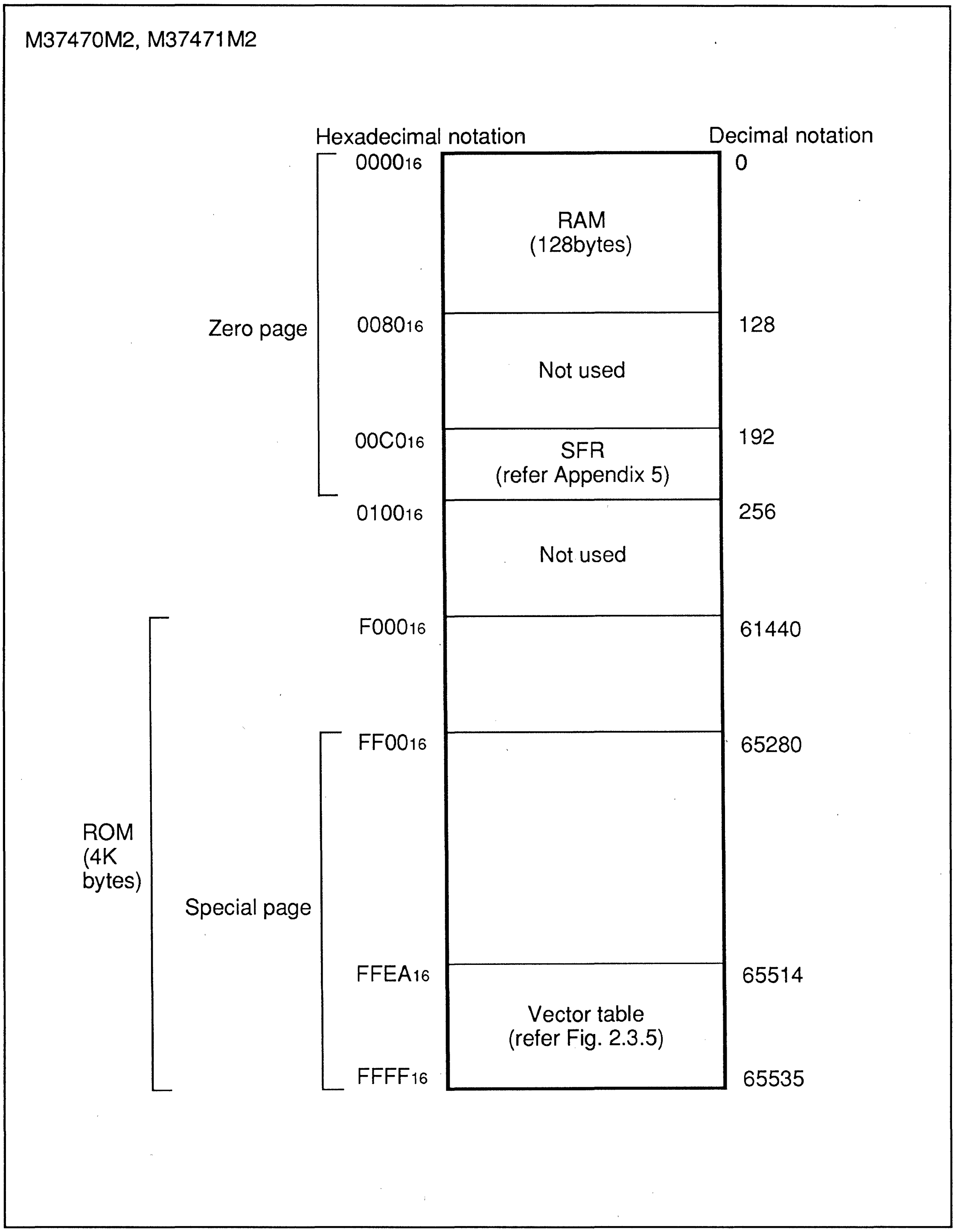
(2) The STP instruction must not be executed during A-D conversion.

5.3 Prevention plain noise and latch-up

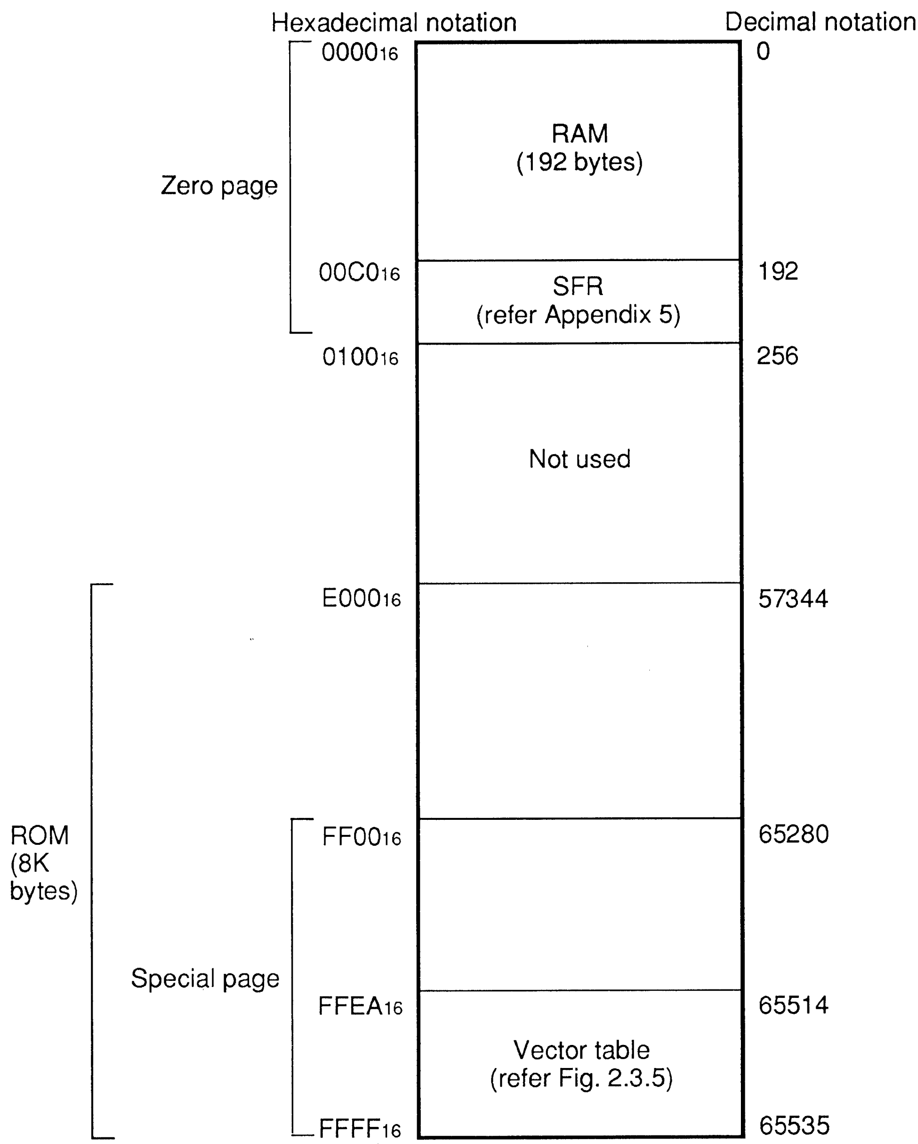
Connect the following external circuits to the M37470 to prevent noise and latch-up:

- Connect a bypass capacitor of approximately 0.1μF between the V_{CC} and V_{SS} pins, using comparatively thick wire that is as short as possible.
- Connect a bypass capacitor of approximately 0.1μF between the V_{REF} and V_{SS} (and AV_{SS} at the 56-pin flat QFP package) pins, using comparatively thick wire that is as short as possible.

Appendix 4 7470 Series memory map

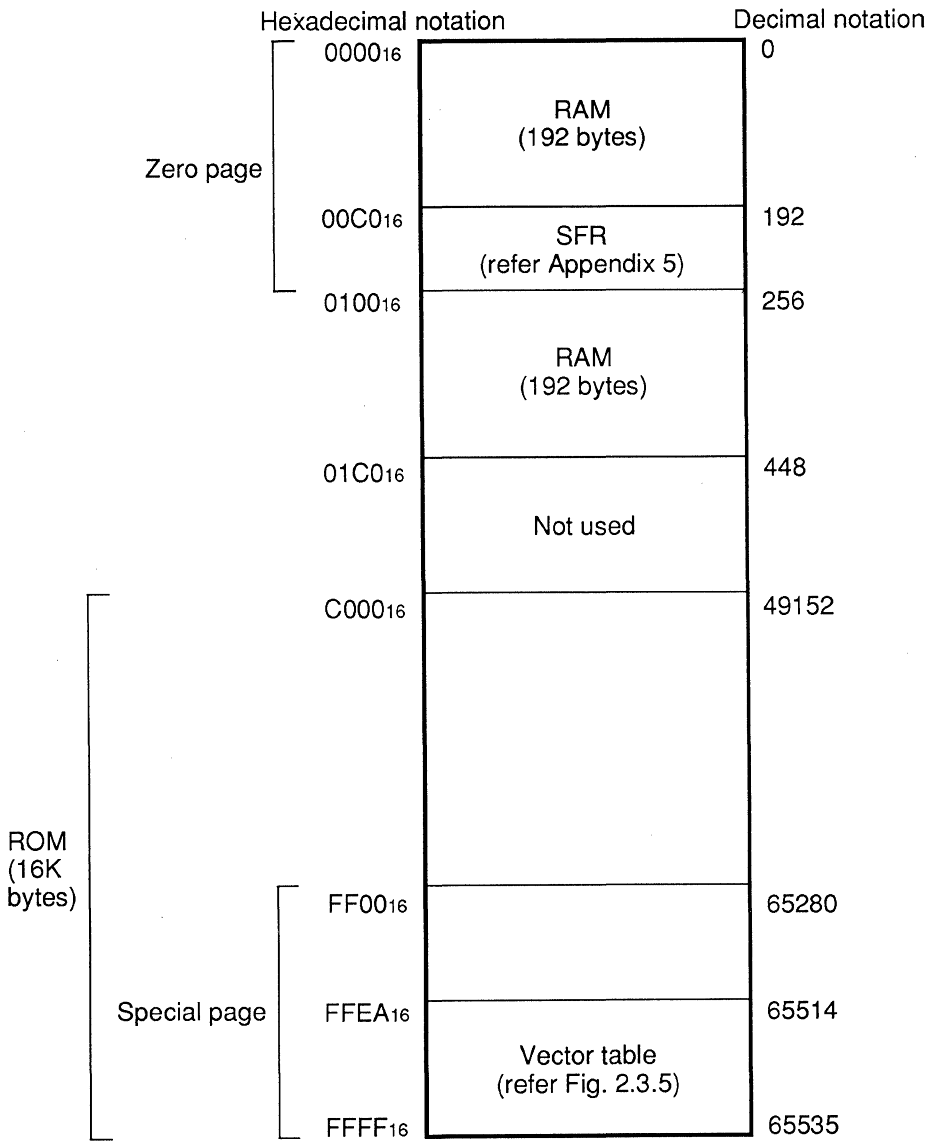


M37470M4, M37471M4



7470 Series memory map

M37470M8, M37471M8

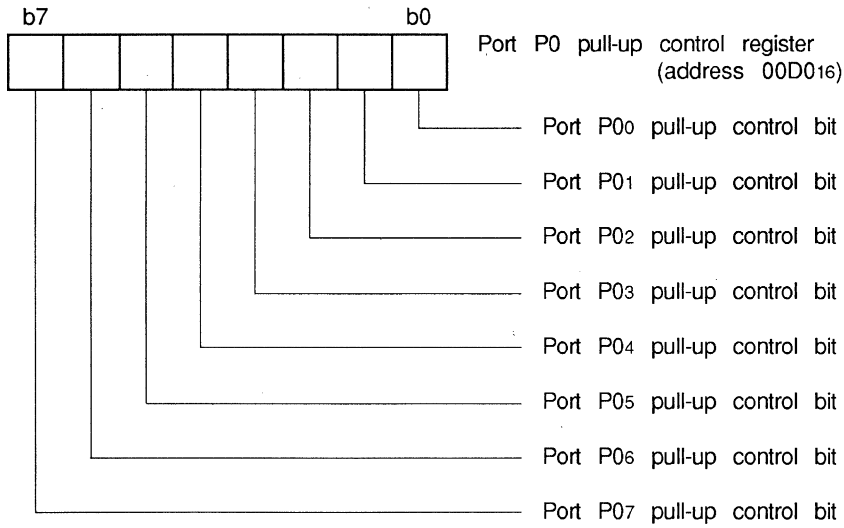


Appendix 5 SFR memory map

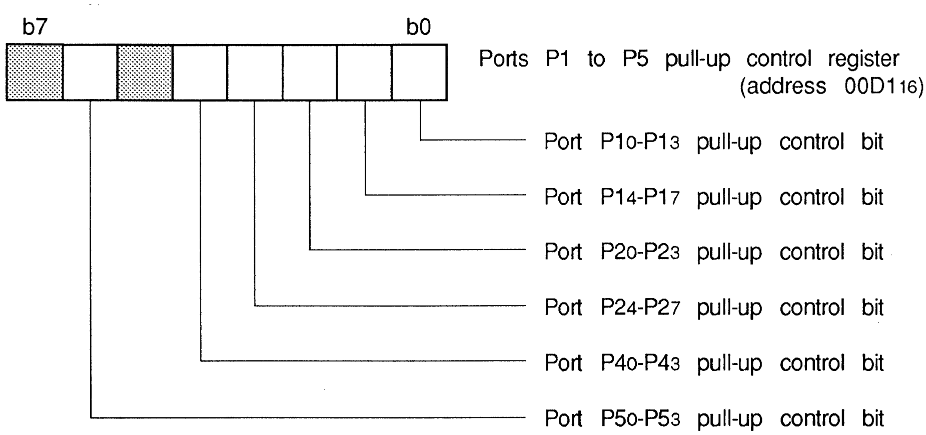
Address	Register	Address	Register
$00C0_{16}$	Port P0	$00E0_{16}$	
$00C1_{16}$	Port P0 direction register	$00E1_{16}$	
$00C2_{16}$	Port P1	$00E2_{16}$	
$00C3_{16}$	Port P1 direction register	$00E3_{16}$	
$00C4_{16}$	Port P2	$00E4_{16}$	
$00C5_{16}$	Port P2 direction register	$00E5_{16}$	
$00C6_{16}$	Port P3	$00E6_{16}$	
$00C7_{16}$		$00E7_{16}$	
$00C8_{16}$	Port P4	$00E8_{16}$	
$00C9_{16}$	Port P4 direction register	$00E9_{16}$	
$00CA_{16}$	Port P5 (Note 1)	$00EA_{16}$	
$00CB_{16}$		$00EB_{16}$	
$00CC_{16}$		$00EC_{16}$	
$00CD_{16}$		$00ED_{16}$	
$00CE_{16}$		$00EE_{16}$	
$00CF_{16}$		$00EF_{16}$	
$00D0_{16}$	Port P0 pull-up control register	$00F0_{16}$	Timer 1
$00D1_{16}$	Ports P1 to P5 pull-up control register	$00F1_{16}$	Timer 2
$00D2_{16}$		$00F2_{16}$	Timer 3
$00D3_{16}$		$00F3_{16}$	Timer 4
$00D4_{16}$	Edge polarity selection register	$00F4_{16}$	
$00D5_{16}$		$00F5_{16}$	
$00D6_{16}$	Input latch register	$00F6_{16}$	
$00D7_{16}$		$00F7_{16}$	Timer FF register
$00D8_{16}$		$00F8_{16}$	Timer 12 mode register
$00D9_{16}$	A-D control register	$00F9_{16}$	Timer 34 mode register
$00DA_{16}$	A-D conversion register	$00FA_{16}$	Timer mode register 2
$00DB_{16}$		$00FB_{16}$	CPU mode register
$00DC_{16}$	Serial I/O mode register	$00FC_{16}$	Interrupt request register 1
$00DD_{16}$	Serial I/O register	$00FD_{16}$	Interrupt request register 2
$00DE_{16}$	Serial I/O counter / Byte counter	$00FE_{16}$	Interrupt control register 1
$00DF_{16}$		$00FF_{16}$	Interrupt control register 2

Note 1: This address is not used in M37470M2, M37470M4 and M37470M8.
2: Shaded area is not used.

Appendix 6 Control registers

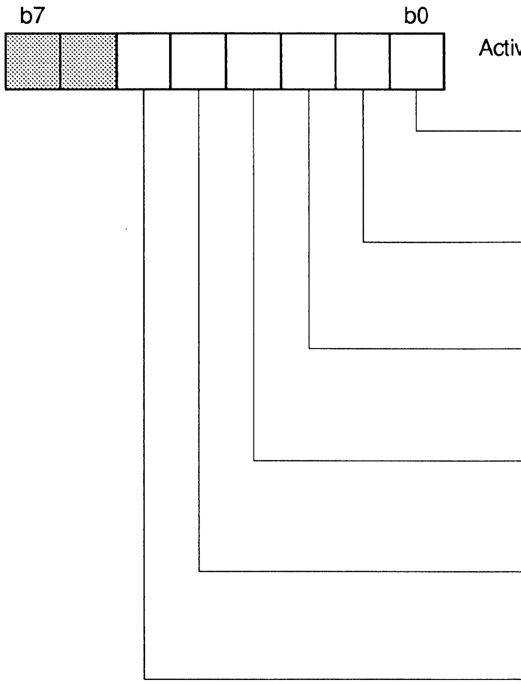


0: without pull-up transistor
1: with pull-up transistor (when input port is selected)



0: without pull-up transistor
1: with pull-up transistor (when input port is selected)

Note: M37470M2, M37470M4 and M37470M8 do not have bit 3 and bit 6.



Active edge selection register : EG
(address $00D4_{16}$)

INT_0 edge selection bit
0: falling edge
1: rising edge

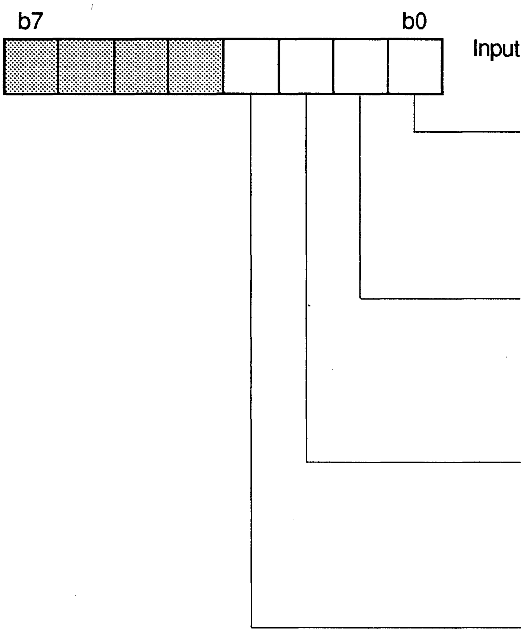
INT_1 edge selection bit
0: falling edge
1: rising edge

$CNTR_0$ edge selection bit
0: falling edge
1: rising edge

$CNTR_1$ edge selection bit
0: falling edge
1: rising edge

$CNTR_0$/$CNTR_1$ interrupt selection bit
0: $CNTR_0$
1: $CNTR_1$

INT_1 source selection bit at WIT or STP instruction execution
0: $P3_1$/INT_1
1: $P0_0$–$P0_7$ "L" level input (for key on wake up)



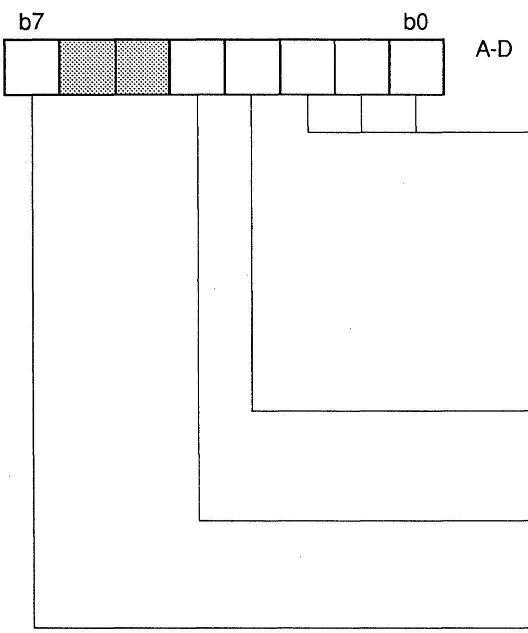
Input latch register : ILR
(address $00D6_{16}$)

$P3_0/INT_0$ latch bit
When INT_0 edge selection bit is "0", the level of the INT_0 pin is inverted then latched; when "1", the level of the INT_0 pin is latched as it is.

$P3_1/INT_1$ latch bit
When INT_1 edge selection bit is "0", the level of the INT_1 pin is inverted then latched; when "1", the level of the INT_1 pin is latched as it is.

$P3_2/CNTR_0$ latch bit
When $CNTR_0$ edge selection bit is "0", the level of the $CNTR_0$ pin is inverted then latched; when "1", the level of the $CNTR_0$ pin is latched as it is.

$P3_3/CNTR_1$ latch bit
When $CNTR_1$ edge selection bit is "0", the level of the $CNTR_1$ pin is inverted then latched; when "1", the level of the $CNTR_1$ pin is latched as it is.



A-D control register : ADCON
(address $00D9_{16}$)

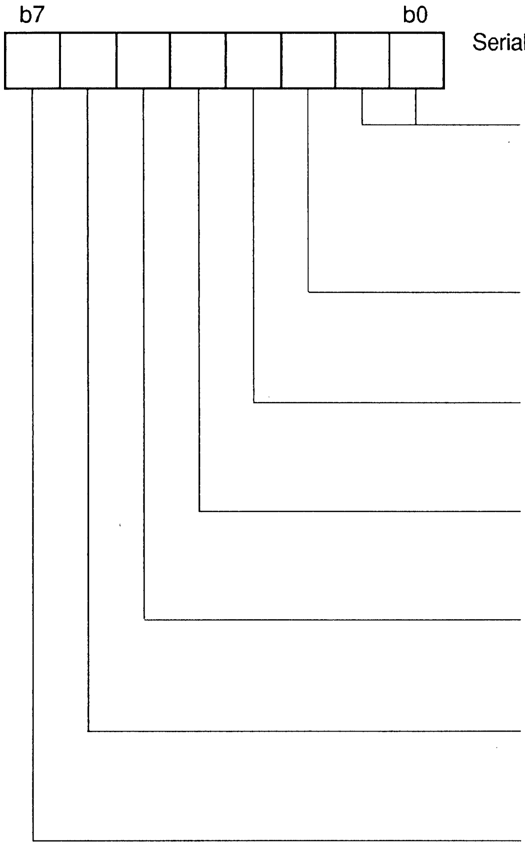
Analog input selection bit
000: $P2_0/IN_0$
001: $P2_1/IN_1$
010: $P2_2/IN_2$
011: $P2_3/IN_3$
100: $P2_4/IN_4$
101: $P2_5/IN_5$
110: $P2_6/IN_6$
111: $P2_7/IN_7$

A-D conversion completion bit
0: during conversion
1: conversion completion

V_{REF} connection selection bit
0: V_{REF} connection cut
1: V_{REF} connected

This bit is always set to "0"

Note: The M37470M2, M37470M4 and M37470M8 do not have the $P2_4/IN_4$ to $P2_7/IN_7$ pins.



Serial I/O mode register : SM
(address $00DC_{16}$)

Internal clock selection bit
- 00: $f(X_{IN})$ or $f(X_{CIN})$ divided by 8
- 01: $f(X_{IN})$ or $f(X_{CIN})$ divided by 16
- 10: $f(X_{IN})$ or $f(X_{CIN})$ divided by 32
- 11: $f(X_{IN})$ or $f(X_{CIN})$ divided by 512

Synchronization clock selection bit
- 0: external clock
- 1: internal clock

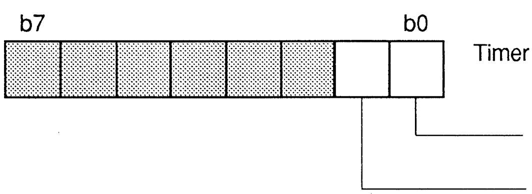
Serial I/O port selection bit
- 0: normal port ($P1_5$, $P1_6$)
- 1: serial I/O port (S_{OUT}, CLK)

$\overline{S_{RDY}}$ signal output selection bit
- 0: Normal port ($P1_7$)
- 1: $\overline{S_{RDY}}$ signal output

$\overline{S_{RDY}}$ signal selection bit
- 0: $\overline{S_{RDY}}$ signal
- 1: S_{ARDY} signal

Serial I/O byte specification mode selection bit
- 0: normal mode
- 1: byte specification mode

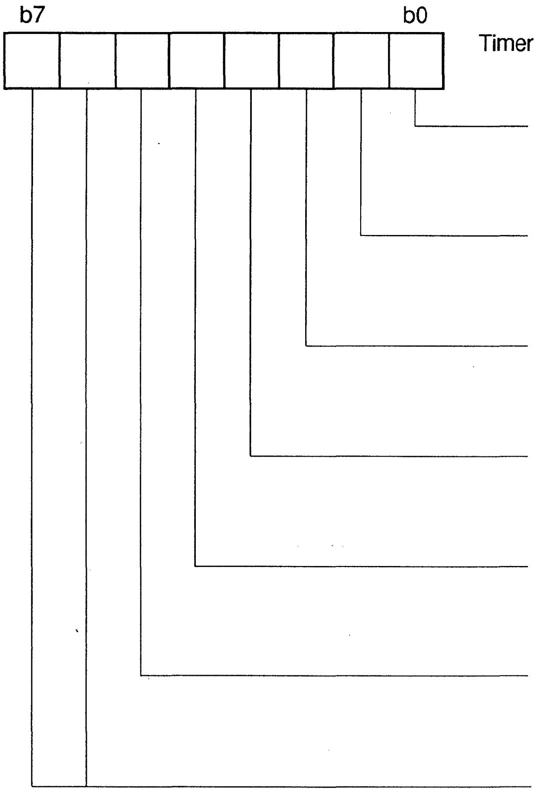
$P1_5/S_{OUT}$, $\overline{S_{RDY}}$ output format selection bit
- 0: CMOS output
- 1: N-channel open drain output



Timer FF register : TF
(address $00F7_{16}$)

Timer 1 overflow flip-flop

Timer 4 overflow flip-flop



Timer 1, 2 mode register : T12M
(address $00F8_{16}$)

Timer 1 count stop bit
0: count
1: stop

Timer 1 count source selection bit
0: internal clock
1: $CNTR_0$

Timer 1 internal clock source selection bit
0: $f(X_{IN})$ or $f(X_{CIN})$ divided by 16
1: $CNTR_0$

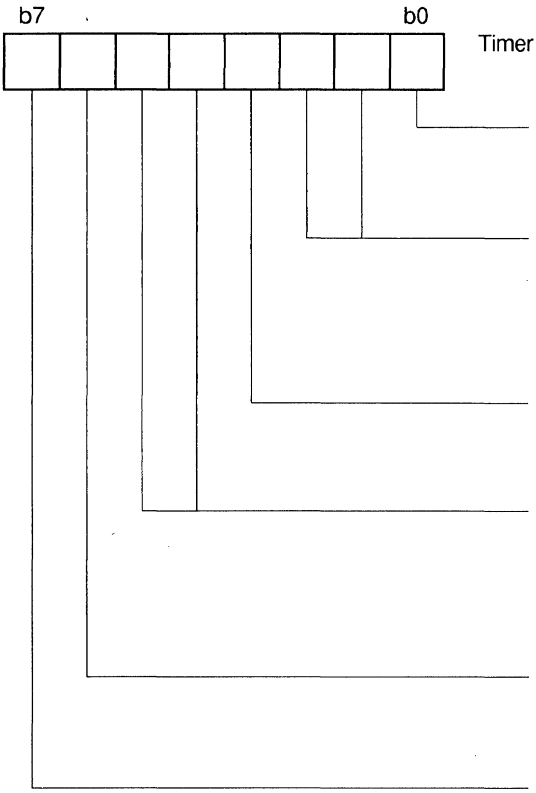
Port $P1_2/T_0$ output selection bit
0: port $P1_2$
1: T_0 (timer 1 overflow output divided by 2)

Timer 2 count stop bit
0: count
1: stop

Timer 2 count source selection bit
0: internal clock
1: timer 1 overflow signal

Timer 2 internal clock source selection bit
00: $f(X_{IN})$ or $f(X_{CIN})$ divided by 16
01: $f(X_{IN})$ or $f(X_{CIN})$ divided by 64
10: $f(X_{IN})$ or $f(X_{CIN})$ divided by 128
11: $f(X_{IN})$ or $f(X_{CIN})$ divided by 256

Note: Do not select $f(X_{CIN})$ as the count source in the M37470M2, M37470M4 and M37470M8.



Timer 3, 4 mode register : T34M
(address $00F9_{16}$)

Timer 3 count stop bit
0: count
1: stop

Timer 3 count source selection bit
00: $f(X_{IN})$ or $f(X_{CIN})$ divided by 16
01: $f(X_{CIN})$
10: timer 1 or timer 2 overflow
11: $CNTR_1$

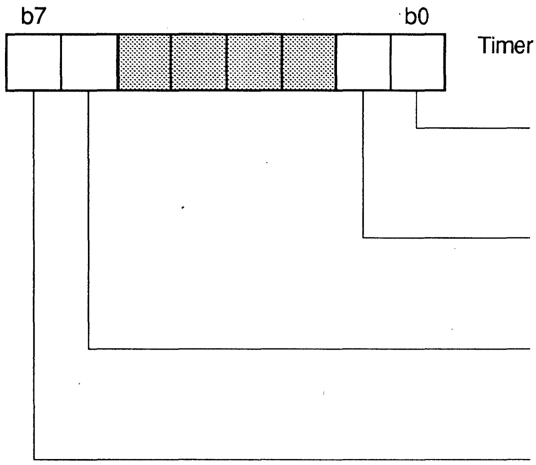
Timer 4 count stop bit
0: count
1: stop

Timer 4 count source selection bit
00: timer 3 overflow signal
01: $f(X_{IN})$ or $f(X_{CIN})$ divided by 16
10: timer 1 or timer 2 overflow
11: $CNTR_1$

Timer 4 pulse width measurement mode
0: timer mode
1: pulse width measurement mode

Port $P1_3/T_1$ output selection bit
0: port $P1_3$
1: T_1(timer 4 overflow output divided by 2 or PWM output)

Note: Do not select $f(X_{CIN})$ as the count source in the M37470M2, M37470M4 and M37470M8.



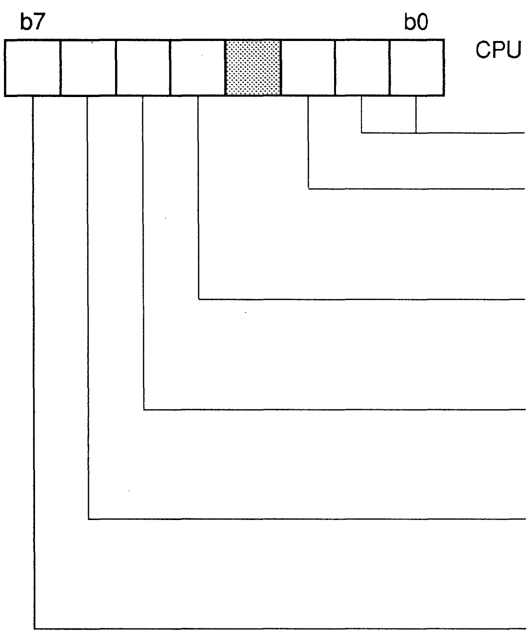
Timer mode register 2 : TM2
(address $00FA_{16}$)

Timer 1 overflow flip-flop set enable bit
0: set disable
1: set enable

Timer 4 overflow flip-flop set enable bit
0: set disable
1: set enable

Timer 3 and timer 4 count overflow signal selection bit
0: timer 1 overflow signal
1: timer 2 overflow signal

Timer 3 and timer 4 function selection bit
0: normal mode
1: PWM mode



CPU mode register : CM
(address $00FB_{16}$)

These bits are set to “0”

Stack page selection bit
0: page 0
1: page 1

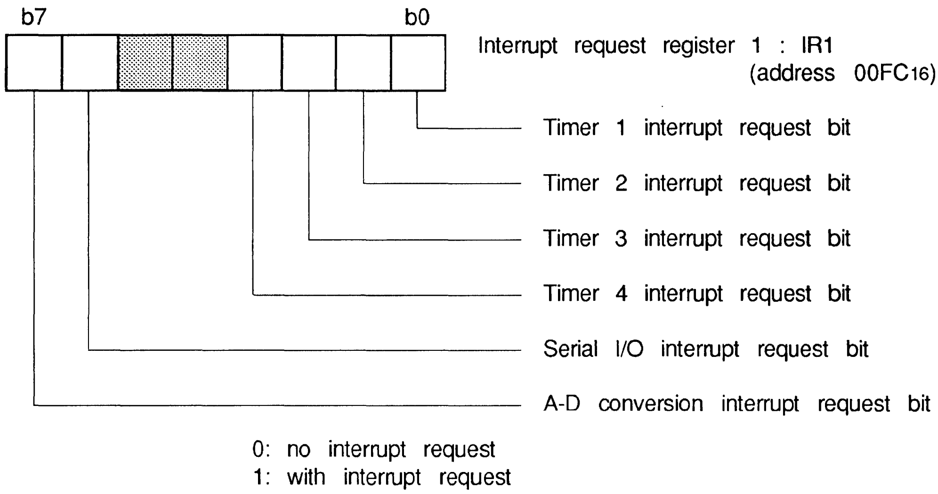
$P5_0$, $P5_1$ / X_{CIN}, X_{COUT} selection bit
0: $P5_0$, $P5_1$
1: X_{CIN}, X_{COUT}

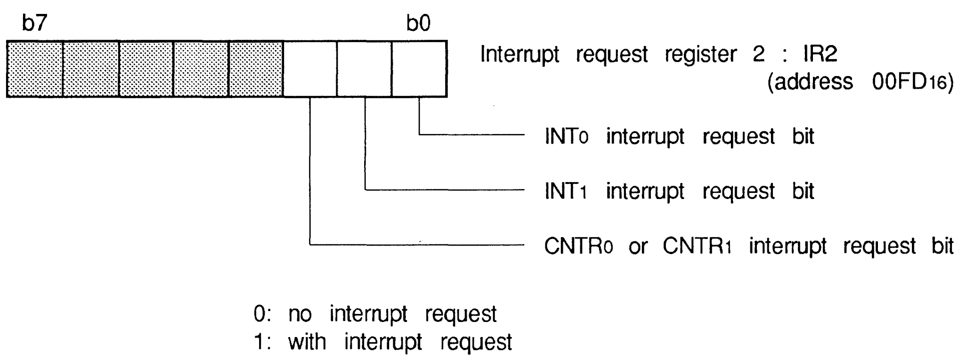
X_{COUT} drive power selection bit
0: low power
1: high power

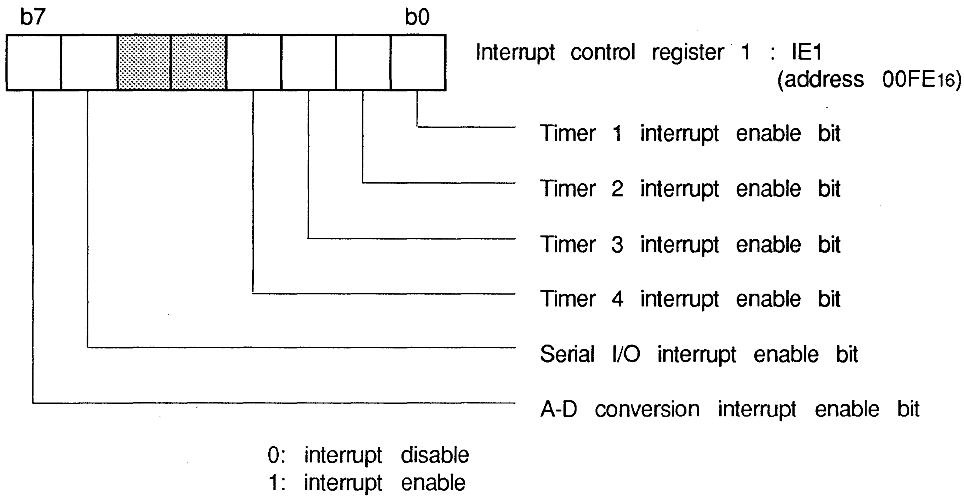
Main clock (X_{IN}-X_{OUT}) stop bit
0: oscillation
1: stop

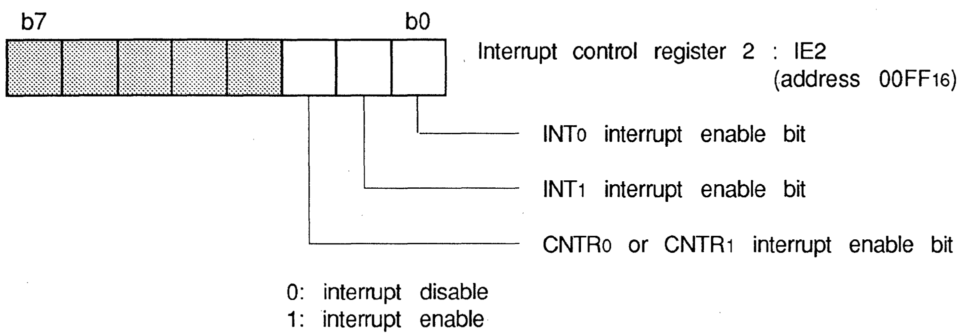
System clock selection bit
0: X_{IN}-X_{OUT} selection (normal mode)
1: X_{CIN}-X_{COUT} selection (low-speed mode)

Note: In the M37470M2 and M37470M4, set all of these bits to “0”.
In the M37470M8, set all of these bits to “0”, except for bit 2.
In the M37471M2 and M37471M4, set bit 2 to “0”.


b7
b0
Interrupt request register 1 : IR1
(address 00FC16)
Timer 1 interrupt request bit
Timer 2 interrupt request bit
Timer 3 interrupt request bit
Timer 4 interrupt request bit
Serial I/O interrupt request bit
A-D conversion interrupt request bit
0: no interrupt request
1: with interrupt request


b7
b0
Interrupt request register 2 : IR2
(address 00FD16)
INT0 interrupt request bit
INT1 interrupt request bit
CNTR0 or CNTR1 interrupt request bit
0: no interrupt request
1: with interrupt request


b7
b0
Interrupt control register 1 : IE1
(address 00FE16)
Timer 1 interrupt enable bit
Timer 2 interrupt enable bit
Timer 3 interrupt enable bit
Timer 4 interrupt enable bit
Serial I/O interrupt enable bit
A-D conversion interrupt enable bit
0: interrupt disable
1: interrupt enable


b7
b0
Interrupt control register 2 : IE2
(address 00FF16)
INT0 interrupt enable bit
INT1 interrupt enable bit
CNTR0 or CNTR1 interrupt enable bit
0: interrupt disable
1: interrupt enable

Appendix 7 Machine language instruction table

Classification		SYMBOL	FUNCTION	FLAG NVTBDIZC	INSTRUCTION CODE D7D6D5D4 D3D2D1D0	HEX	BYTE NUMBER	CYCLE NUMBER	NOTE
Data Transfer	Load	LDA #$nn	(A) ← nn	○×××××○×	1010 1001 <B2>	A9	2	2	2
		LDA $zz	(A) ← (M) where M=(zz)	○×××××○×	1010 0101 <B2>	A5	2	3	2
		LDA $zz,X	(A) ← (M) where M=(zz+(X))	○×××××○×	1011 0101 <B2>	B5	2	4	2
		LDA $hhll	(A) ← (M) where M=(hhll)	○×××××○×	1010 1101 <B2> <B3>	AD	3	4	2
		LDA $hhll,X	(A) ← (M) where M=(hhll+(X))	○×××××○×	1011 1101 <B2> <B3>	BD	3	5	2
		LDA $hhll,Y	(A) ← (M) where M=(hhll+(Y))	○×××××○×	1011 1001 <B2> <B3>	B9	3	5	2
		LDA ($zz,X)	(A) ← (M) where M=((zz+(X)+1)(zz+(X)))	○×××××○×	1010 0001 <B2>	A1	2	6	2
		LDA ($zz),Y	(A) ← (M) where M=((zz+1)(zz)+(Y))	○×××××○×	1011 0001 <B2>	B1	2	6	2
		LDX #$nn	(X) ← nn	○×××××○×	1010 0010 <B2>	A2	2	2	
		LDX $zz	(X) ← (M) where M=(zz)	○×××××○×	1010 0110 <B2>	A6	2	3	
		LDX $zz,Y	(X) ← (M) where M=(zz+(Y))	○×××××○×	1011 0110 <B2>	B6	2	4	
		LDX $hhll	(X) ← (M) where M=(hhll)	○×××××○×	1010 1110 <B2> <B3>	AE	3	4	
		LDX $hhll,Y	(X) ← (M) where M=(hhll+(Y))	○×××××○×	1011 1110 <B2> <B3>	BE	3	5	
		LDY #$nn	(Y) ← nn	○×××××○×	1010 0000 <B2>	A0	2	2	
		LDY $zz	(Y) ← (M) where M=(zz)	○×××××○×	1010 0100 <B2>	A4	2	3	
		LDY $zz,X	(Y) ← (M) where M=(zz+(X))	○×××××○×	1011 0100 <B2>	B4	2	4	
		LDY $hhll	(Y) ← (M) where M=(hhll)	○×××××○×	1010 1100 <B2> <B3>	AC	3	4	
		LDY $hhll,X	(Y) ← (M) where M=(hhll+(X))	○×××××○×	1011 1100 <B2> <B3>	BC	3	5	
		LDM #$nn,$zz	(M) ← nn where M=(zz)	××××××××	0011 1100 <B2> <B3>	3C	3	4	
	Store	STA $zz	(M) ← (A) where M=(zz)	××××××××	1000 0101 <B2>	85	2	4	
		STA $zz,X	(M) ← (A) where M=(zz+(X))	××××××××	1001 0101 <B2>	95	2	5	
		STA $hhll	(M) ← (A) where M=(hhll)	××××××××	1000 1101 <B2> <B3>	8D	3	5	
		STA $hhll,X	(M) ← (A) where M=(hhll+(X))	××××××××	1001 1101 <B2> <B3>	9D	3	6	
		STA $hhll,Y	(M) ← (A) where M=(hhll+(Y))	××××××××	1001 1001 <B2> <B3>	99	3	6	
		STA ($zz,X)	(M) ← (A) where M=((zz+(X)+1)(zz+(X)))	××××××××	1000 0001 <B2>	81	2	7	
		STA ($zz),Y	(M) ← (A) where M=((zz+1)(zz)+(Y))	××××××××	1001 0001 <B2>	91	2	7	
		STX $zz	(M) ← (X) where M=(zz)	××××××××	1000 0110 <B2>	86	2	4	
		STX $zz,Y	(M) ← (X) where M=(zz+(Y))	××××××××	1001 0110 <B2>	96	2	5	
		STX $hhll	(M) ← (X) where M=(hhll)	××××××××	1000 1110 <B2> <B3>	8E	3	5	
		STY $zz	(M) ← (Y) where M=(zz)	××××××××	1000 0100 <B2>	84	2	4	
		STY $zz,X	(M) ← (Y) where M=(zz+(X))	××××××××	1001 0100 <B2>	94	2	5	
		STY $hhll	(M) ← (Y) where M=(hhll)	××××××××	1000 1100 <B2> <B3>	8C	3	6	
	Transfer	TAX	(X) ← (A)	○×××××○×	1010 1010	AA	1	2	
		TXA	(A) ← (X)	○×××××○×	1000 1010	8A	1	2	
		TAY	(Y) ← (A)	○×××××○×	1010 1000	A8	1	2	
		TYA	(A) ← (Y)	○×××××○×	1001 1000	98	1	2	
		TSX	(X) ← (S)	○×××××○×	1011 1010	BA	1	2	
		TXS	(S) ← (X)	××××××××	1001 1010	9A	1	2	
	Stack Operation	PHA	(M(S)) ← (A) , (S) ← (S) − 1	××××××××	0100 1000	48	1	3	
		PHP	(M(S)) ← (PS) , (S) ← (S) − 1	××××××××	0000 1000	08	1	3	
		PLA	(S) ← (S) + 1 , (A) ← (M(S))	○×××××○×	0110 1000	68	1	4	
		PLP	(S) ← (S) + 1 , (PS) ← (M(S))	Previous status in stack	0010 1000	28	1	4	

APPENDIX 7

Machine language instruction table

Parameter / Classification		SYMBOL	FUNCTION	FLAG N V T B D I Z C	INSTRUCTION CODE D7 D6 D5 D4 D3 D2 D1 D0	HEX	BYTE NUMBER	CYCLE NUMBER	NOTE
Operation	Add and Substruct	ADC #$nn	(A) ← (A) + nn + (C)	○ ○ × × × × ○ ○	0 1 1 0 1 0 0 1 <B2>	69	2	2	1
		ADC $zz	(A) ← (A) + (M) + (C) where M=(zz)	○ ○ × × × × ○ ○	0 1 1 0 0 1 0 1 <B2>	65	2	3	1
		ADC $zz,X	(A) ← (A) + (M) + (C) where M=(zz+(X))	○ ○ × × × × ○ ○	0 1 1 1 0 1 0 1 <B2>	75	2	4	1
		ADC $hhll	(A) ← (A) + (M) + (C) where M=(hhll)	○ ○ × × × × ○ ○	0 1 1 0 1 1 0 1 <B2> <B3>	6D	3	4	1
		ADC $hhll,X	(A) ← (A) + (M) + (C) where M=(hhll+(X))	○ ○ × × × × ○ ○	0 1 1 1 1 1 0 1 <B2> <B3>	7D	3	5	1
		ADC $hhll,Y	(A) ← (A) + (M) + (C) where M=(hhll+(Y))	○ ○ × × × × ○ ○	0 1 1 1 1 0 0 1 <B2> <B3>	79	3	5	1
		ADC ($zz,X)	(A) ← (A) + (M) + (C) where M=((zz+(X)+1)(zz+(X)))	○ ○ × × × × ○ ○	0 1 1 0 0 0 0 1 <B2>	61	2	6	1
		ADC ($zz),Y	(A) ← (A) + (M) + (C) where M=((zz+1)(zz)+(Y))	○ ○ × × × × ○ ○	0 1 1 1 0 0 0 1 <B2>	71	2	6	1
		SBC #$nn	(A) ← (A) − nn − $\overline{(C)}$	○ ○ × × × × ○ ○	1 1 1 0 1 0 0 1 <B2>	E9	2	2	1
		SBC $zz	(A) ← (A) − (M) − $\overline{(C)}$ where M=(zz)	○ ○ × × × × ○ ○	1 1 1 0 0 1 0 1 <B2>	E5	2	3	1
		SBC $zz,X	(A) ← (A) − (M) − $\overline{(C)}$ where M=(zz+(X))	○ ○ × × × × ○ ○	1 1 1 1 0 1 0 1 <B2>	F5	2	4	1
		SBC $hhll	(A) ← (A) − (M) − $\overline{(C)}$ where M=(hhll)	○ ○ × × × × ○ ○	1 1 1 0 1 1 0 1 <B2> <B3>	ED	3	4	1
		SBC $hhll,X	(A) ← (A) − (M) − $\overline{(C)}$ where M=(hhll+(X))	○ ○ × × × × ○ ○	1 1 1 1 1 1 0 1 <B2> <B3>	FD	3	5	1
		SBC $hhll,Y	(A) ← (A) − (M) − $\overline{(C)}$ where M=(hhll+(Y))	○ ○ × × × × ○ ○	1 1 1 1 1 0 0 1 <B2> <B3>	F9	3	5	1
		SBC ($zz,X)	(A) ← (A) − (M) − $\overline{(C)}$ where M=((zz+(X)+1)(zz+(X)))	○ ○ × × × × ○ ○	1 1 1 0 0 0 0 1 <B2>	E1	2	6	1
		SBC ($zz),Y	(A) ← (A) − (M) − $\overline{(C)}$ where M=((zz+1)(zz)+(Y))	○ ○ × × × × ○ ○	1 1 1 1 0 0 0 1 <B2>	F1	2	6	1
		INC A	(A) ← (A) + 1	○ × × × × × ○ ×	0 0 1 1 1 0 1 0	3A	1	2	
		INC $zz	(M) ← (M) + 1 where M=(zz)	○ × × × × × ○ ×	1 1 1 0 0 1 1 0 <B2>	E6	2	5	
		INC $zz,X	(M) ← (M) + 1 where M=(zz+(X))	○ × × × × × ○ ×	1 1 1 1 0 1 1 0 <B2>	F6	2	6	
		INC $hhll	(M) ← (M) + 1 where M=(hhll)	○ × × × × × ○ ×	1 1 1 0 1 1 1 0 <B2> <B3>	EE	3	6	
		INC $hhll,X	(M) ← (M) + 1 where M=(hhll+(X))	○ × × × × × ○ ×	1 1 1 1 1 1 1 0 <B2> <B3>	FE	3	7	
		DEC A	(A) ← (A) − 1	○ × × × × × ○ ×	0 0 0 1 1 0 1 0	1A	1	2	
		DEC $zz	(M) ← (M) − 1 where M=(zz)	○ × × × × × ○ ×	1 1 0 0 0 1 1 0 <B2>	C6	2	5	
		DEC $zz,X	(M) ← (M) − 1 where M=(zz+(X))	○ × × × × × ○ ×	1 1 0 1 0 1 1 0 <B2>	D6	2	6	
		DEC $hhll	(M) ← (M) − 1 where M=(hhll)	○ × × × × × ○ ×	1 1 0 0 1 1 1 0 <B2> <B3>	CE	3	6	
		DEC $hhll,X	(M) ← (M) − 1 where M=(hhll+(X))	○ × × × × × ○ ×	1 1 0 1 1 1 1 0 <B2> <B3>	DE	3	7	
		INX	(X) ← (X) + 1	○ × × × × × ○ ×	1 1 1 0 1 0 0 0	E8	1	2	
		DEX	(X) ← (X) − 1	○ × × × × × ○ ×	1 1 0 0 1 0 1 0	CA	1	2	
		INY	(Y) ← (Y) + 1	○ × × × × × ○ ×	1 1 0 0 1 0 0 0	C8	1	2	
		DEY	(Y) ← (Y) − 1	○ × × × × × ○ ×	1 0 0 0 1 0 0 0	88	1	2	

APPENDIX 7

Machine language instruction table

Parameter / Classification		SYMBOL	FUNCTION	FLAG NVTBDIZC	INSTRUCTION CODE D7 D6 D5 D4 D3 D2 D1 D0	HEX	BYTE NUMBER	CYCLE NUMBER	NOTE
Operation	Logic Operation	AND #$nn	(A) ← (A) ∧ nn	○×××××○×	0 0 1 0 1 0 0 1 <B2>	29	2	2	1
		AND $zz	(A) ← (A) ∧ (M) where M=(zz)	○×××××○×	0 0 1 0 0 1 0 1 <B2>	25	2	3	1
		AND $zz,X	(A) ← (A) ∧ (M) where M=(zz+(X))	○×××××○×	0 0 1 1 0 1 0 1 <B2>	35	2	4	1
		AND $hhll	(A) ← (A) ∧ (M) where M=(hhll)	○×××××○×	0 0 1 0 1 1 0 1 <B2> <B3>	2D	3	4	1
		AND $hhll,X	(A) ← (A) ∧ (M) where M=(hhll+(X))	○×××××○×	0 0 1 1 1 1 0 1 <B2> <B3>	3D	3	5	1
		AND $hhll,Y	(A) ← (A) ∧ (M) where M=(hhll+(Y))	○×××××○×	0 0 1 1 1 0 0 1 <B2> <B3>	39	3	5	1
		AND ($zz,X)	(A) ← (A) ∧ (M) where M=((zz+(X)+1)(zz+(X)))	○×××××○×	0 0 1 0 0 0 0 1 <B2>	21	2	6	1
		AND ($zz),Y	(A) ← (A) ∧ (M) where M=((zz+1)(zz)+(Y))	○×××××○×	0 0 1 1 0 0 0 1 <B2>	31	2	6	1
		ORA #$nn	(A) ← (A) ∨ nn	○×××××○×	0 0 0 0 1 0 0 1 <B2>	09	2	2	1
		ORA $zz	(A) ← (A) ∨ (M) where M=(zz)	○×××××○×	0 0 0 0 0 1 0 1 <B2>	05	2	3	1
		ORA $zz,X	(A) ← (A) ∨ (M) where M=(zz+(X))	○×××××○×	0 0 0 1 0 1 0 1 <B2>	15	2	4	1
		ORA $hhll	(A) ← (A) ∨ (M) where M=(hhll)	○×××××○×	0 0 0 0 1 1 0 1 <B2> <B3>	0D	3	4	1
		ORA $hhll,X	(A) ← (A) ∨ (M) where M=(hhll+(X))	○×××××○×	0 0 0 1 1 1 0 1 <B2> <B3>	1D	3	5	1
		ORA $hhll,Y	(A) ← (A) ∨ (M) where M=(hhll+(Y))	○×××××○×	0 0 0 1 1 0 0 1 <B2> <B3>	19	3	5	1
		ORA ($zz,X)	(A) ← (A) ∨ (M) where M=((zz+(X)+1)(zz+(X)))	○×××××○×	0 0 0 0 0 0 0 1 <B2>	01	2	6	1
		ORA ($zz),Y	(A) ← (A) ∨ (M) where M=((zz+1)(zz)+(Y))	○×××××○×	0 0 0 1 0 0 0 1 <B2>	11	2	6	1
		EOR #$nn	(A) ← (A) ∀ nn	○×××××○×	0 1 0 0 1 0 0 1 <B2>	49	2	2	1
		EOR $zz	(A) ← (A) ∀ (M) where M=(zz)	○×××××○×	0 1 0 0 0 1 0 1 <B2>	45	2	3	1
		EOR $zz,X	(A) ← (A) ∀ (M) where M=(zz+(X))	○×××××○×	0 1 0 1 0 1 0 1 <B2>	55	2	4	1
		EOR $hhll	(A) ← (A) ∀ (M) where M=(hhll)	○×××××○×	0 1 0 0 1 1 0 1 <B2> <B3>	4D	3	4	1
		EOR $hhll,X	(A) ← (A) ∀ (M) where M=(hhll+(X))	○×××××○×	0 1 0 1 1 1 0 1 <B2> <B3>	5D	3	5	1
		EOR $hhll,Y	(A) ← (A) ∀ (M) where M=(hhll+(Y))	○×××××○×	0 1 0 1 1 0 0 1 <B2> <B3>	59	3	5	1
		EOR ($zz,X)	(A) ← (A) ∀ (M) where M=((zz+(X)+1)(zz+(X)))	○×××××○×	0 1 0 0 0 0 0 1 <B2>	41	2	6	1
		EOR ($zz),Y	(A) ← (A) ∀ (M) where M=((zz+1)(zz)+(Y))	○×××××○×	0 1 0 1 0 0 0 1 <B2>	51	2	6	1
		COM $zz	(M) ← $\overline{(M)}$ where M=(zz)	○×××××○×	0 1 0 0 0 1 0 0 <B2>	44	2	5	
		BIT $zz	(A) ∧ (M) where M=(zz)	M_7M_6××××○×	0 0 1 0 0 1 0 0 <B2>	24	2	3	
		BIT $hhll	(A) ∧ (M) where M=(hhll)	M_7M_6××××○×	0 0 1 0 1 1 0 0 <B2> <B3>	2C	3	4	
		TST $zz	(M)=0 ? where M=(zz)	○×××××○×	0 1 1 0 0 1 0 0 <B2>	64	2	3	

APPENDIX 7

Machine language instruction table

Parameter / Classification		SYMBOL	FUNCTION		FLAG NVTBDIZC	INSTRUCTION CODE D7D6D5D4D3D2D1D0	HEX	BYTE NUMBER	CYCLE NUMBER	NOTE
Operation	Comparison	CMP #$nn	(A) — nn	Comparison in size	OXXXXXOO	1 1 0 0 1 0 0 1 <B2>	C9	2	2	3
Operation	Comparison	CMP $zz	(A) — (M)	where M=(zz)	OXXXXXOO	1 1 0 0 0 1 0 1 <B2>	C5	2	3	3
Operation	Comparison	CMP $zz,X	(A) — (M)	where M=(zz+(X))	OXXXXXOO	1 1 0 1 0 1 0 1 <B2>	D5	2	4	3
Operation	Comparison	CMP $hhll	(A) — (M)	where M=(hhll)	OXXXXXOO	1 1 0 0 1 1 0 1 <B2> <B3>	CD	3	4	3
Operation	Comparison	CMP $hhll,X	(A) — (M)	where M=(hhll+(X))	OXXXXXOO	1 1 0 1 1 1 0 1 <B2> <B3>	DD	3	5	3
Operation	Comparison	CMP $hhll,Y	(A) — (M)	where M=(hhll+(Y))	OXXXXXOO	1 1 0 1 1 0 0 1 <B2> <B3>	D9	3	5	3
Operation	Comparison	CMP ($zz,X)	(A) — (M)	where M=((zz+(X)+1)(zz+(X)))	OXXXXXOO	1 1 0 0 0 0 0 1 <B2>	C1	2	6	3
Operation	Comparison	CMP ($zz),Y	(A) — (M)	where M=((zz+1)(zz)+(Y))	OXXXXXOO	1 1 0 1 0 0 0 1 <B2>	D1	2	6	3
Operation	Comparison	CPX #$nn	(X) — nn		OXXXXXOO	1 1 1 0 0 0 0 0 <B2>	E0	2	2	
Operation	Comparison	CPX $zz	(X) — (M)	where M=(zz)	OXXXXXOO	1 1 1 0 0 1 0 0 <B2>	E4	2	3	
Operation	Comparison	CPX $hhll	(X) — (M)	where M=(hhll)	OXXXXXOO	1 1 1 0 1 1 0 0 <B2> <B3>	EC	3	4	
Operation	Comparison	CPY #$nn	(Y) — nn		OXXXXXOO	1 1 0 0 0 0 0 0 <B2>	C0	2	2	
Operation	Comparison	CPY $zz	(Y) — (M)	where M=(zz)	OXXXXXOO	1 1 0 0 0 1 0 0 <B2>	C4	2	3	
Operation	Comparison	CPY $hhll	(Y) — (M)	where M=(hhll)	OXXXXXOO	1 1 0 0 1 1 0 0 <B2> <B3>	CC	3	4	
Operation	Rotate and Shift	ASL A	Left shift C ← A7A6 … A1A0 ← 0		OXXXXXOO	0 0 0 0 1 0 1 0	0A	1	2	
Operation	Rotate and Shift	ASL $zz	Left shift C ← M7M6 … M1M0 ← 0	where M = (zz)	OXXXXXOO	0 0 0 0 0 1 1 0 <B2>	06	2	5	
Operation	Rotate and Shift	ASL $zz,X	Left shift C ← M7M6 … M1M0 ← 0	where M = (zz+(X))	OXXXXXOO	0 0 0 1 0 1 1 0 <B2>	16	2	6	
Operation	Rotate and Shift	ASL $hhll	Left shift C ← M7M6 … M1M0 ← 0	where M = (hhll)	OXXXXXOO	0 0 0 0 1 1 1 0 <B2> <B3>	0E	3	6	
Operation	Rotate and Shift	ASL $hhll,X	Left shift C ← M7M6 … M1M0 ← 0	where M = (hhll+(X))	OXXXXXOO	0 0 0 1 1 1 1 0 <B2> <B3>	1E	3	7	
Operation	Rotate and Shift	LSR A	Right shift 0 → A7A6 … A1A0 → C		OXXXXXOO	0 1 0 0 1 0 1 0	4A	1	2	
Operation	Rotate and Shift	LSR $zz	Right shift 0 → M7M6 … M1M0 → C	where M = (zz)	OXXXXXOO	0 1 0 0 0 1 1 0 <B2>	46	2	5	
Operation	Rotate and Shift	LSR $zz,X	Right shift 0 → M7M6 … M1M0 → C	where M = (zz+(X))	OXXXXXOO	0 1 0 1 0 1 1 0 <B2>	56	2	6	
Operation	Rotate and Shift	LSR $hhll	Right shift 0 → M7M6 … M1M0 → C	where M = (hhll)	OXXXXXOO	0 1 0 0 1 1 1 0 <B2> <B3>	4E	3	6	
Operation	Rotate and Shift	LSR $hhll,X	Right shift 0 → M7M6 … M1M0 → C	where M = (hhll+(X))	OXXXXXOO	0 1 0 1 1 1 1 0 <B2> <B3>	5E	3	7	
Operation	Rotate and Shift	ROL A	Left shift ← A7A6 … A1A0 ← C ←		OXXXXXOO	0 0 1 0 1 0 1 0	2A	1	2	
Operation	Rotate and Shift	ROL $zz	Left shift ← M7M6 … M1M0 ← C ←	where M = (zz)	OXXXXXOO	0 0 1 0 0 1 1 0 <B2>	26	2	5	
Operation	Rotate and Shift	ROL $zz,X	Left shift ← M7M6 … M1M0 ← C ←	where M = (zz+(X))	OXXXXXOO	0 0 1 1 0 1 1 0 <B2>	36	2	6	
Operation	Rotate and Shift	ROL $hhll	Left shift ← M7M6 … M1M0 ← C ←	where M = (hhll)	OXXXXXOO	0 0 1 0 1 1 1 0 <B2> <B3>	2E	3	6	
Operation	Rotate and Shift	ROL $hhll,X	Left shift ← M7M6 … M1M0 ← C ←	where M = (hhll+(X))	OXXXXXOO	0 0 1 1 1 1 1 0 <B2> <B3>	3E	3	7	
Operation	Rotate and Shift	ROR A	Right shift → C → A7A6 … A1A0 →		OXXXXXOO	0 1 1 0 1 0 1 0	6A	1	2	
Operation	Rotate and Shift	ROR $zz	Right shift → C → M7M6 … M1M0 →	where M = (zz)	OXXXXXOO	0 1 1 0 0 1 1 0 <B2>	66	2	5	
Operation	Rotate and Shift	ROR $zz,X	Right shift → C → M7M6 … M1M0 →	where M = (zz+(X))	OXXXXXOO	0 1 1 1 0 1 1 0 <B2>	76	2	6	
Operation	Rotate and Shift	ROR $hhll	Right shift → C → M7M6 … M1M0 →	where M = (hhll)	OXXXXXOO	0 1 1 0 1 1 1 0 <B2> <B3>	6E	3	6	
Operation	Rotate and Shift	ROR $hhll,X	Right shift → C → M7M6 … M1M0 →	where M = (hhll+(X))	OXXXXXOO	0 1 1 1 1 1 1 0 <B2> <B3>	7E	3	7	
Operation	Rotate and Shift	RRF $zz	M7 … M4 ⇄ M3 … M0	where M = (zz)	XXXXXXXX	1 0 0 0 0 0 1 0 <B2>	82	2	8	
Bit Management		CLB i,A	(Ai) ← 0 where i = 0 ~ 7		XXXXXXXX	i i i 1 1 0 1 1	(2i+1) × 10+B	1	2	
Bit Management		CLB i,$zz	(Mi) ← 0 where i = 0 ~ 7, M = (zz)		XXXXXXXX	i i i 1 1 1 1 1 <B2>	(2i+1) × 10+F	2	5	
Bit Management		SEB i,A	(Ai) ← 1 where i = 0 ~ 7		XXXXXXXX	i i i 0 1 0 1 1	2i × 10+B	1	2	
Bit Management		SEB i,$zz	(Mi) ← 1 where i = 0 ~ 7, M = (zz)		XXXXXXXX	i i i 0 1 1 1 1 <B2>	2i × 10+F	2	5	

APPENDIX 7

Machine language instruction table

Parameter / Classification		SYMBOL	FUNCTION	FLAG NVTBDIZC	INSTRUCTION CODE $D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$	INSTRUCTION CODE HEX	BYTE NUMBER	CYCLE NUMBER	NOTE
Flag Setting		CLC	(C) ← 0	xxxxxxx0	0 0 0 1 1 0 0 0	18	1	2	
		SEC	(C) ← 1	xxxxxxx1	0 0 1 1 1 0 0 0	38	1	2	
		CLD	(D) ← 0	xxxx0xxx	1 1 0 1 1 0 0 0	D8	1	2	
		SED	(D) ← 1	xxxx1xxx	1 1 1 1 1 0 0 0	F8	1	2	
		CLI	(I) ← 0	xxxxx0xx	0 1 0 1 1 0 0 0	58	1	2	
		SEI	(I) ← 1	xxxxx1xx	0 1 1 1 1 0 0 0	78	1	2	
		CLT	(T) ← 0	xx0xxxxx	0 0 0 1 0 0 1 0	12	1	2	
		SET	(T) ← 1	xx1xxxxx	0 0 1 1 0 0 1 0	32	1	2	
		CLV	(V) ← 0	x0xxxxxx	1 0 1 1 1 0 0 0	B8	1	2	
Branch and Return	Jump	BRA $hhll	(PC) ← (PC) + 2 + Rel	xxxxxxxx	1 0 0 0 0 0 0 0 <B2>	80	2	4	
		JMP $hhll	(PC) ← hhll	xxxxxxxx	0 1 0 0 1 1 0 0 <B2> <B3>	4C	3	3	
		JMP ($hhll)	(PC_L) ← (hhll), (PC_H) ← (hhll+1)	xxxxxxxx	0 1 1 0 1 1 0 0 <B2> <B3>	6C	3	5	
		JMP ($zz)	(PC_L) ← (zz), (PC_H) ← (zz+1)	xxxxxxxx	1 0 1 1 0 0 1 0 <B2>	B2	2	4	
		JSR $hhll	(M(S)) ← (PC_H), (S) ← (S)—1, (M(S)) ← (PC_L), (S) ← (S)—1, and (PC) ← hhll	xxxxxxxx	0 0 1 0 0 0 0 0 <B2> <B3>	20	3	6	
		JSR ($zz)	(M(S)) ← (PC_H), (S) ← (S)—1, (M(S)) ← (PC_L), (S) ← (S)—1, (PC_L) ← (zz), and (PC_H) ← (zz+1)	xxxxxxxx	0 0 0 0 0 0 1 0 <B2>	02	2	7	
		JSR \$hhll	(M(S)) ← (PC_H), (S) ← (S)—1, (M(S)) ← (PC_L), (S) ← (S)—1, (PC_L) ← ll, and (PC_H) ← FF	xxxxxxxx	0 0 1 0 0 0 1 0 <B2>	22	2	5	
	Branch	BBC i, A,$hhll	When(Ai)=0 (PC) ← (PC) + 2 + Rel Where i=0~7 When(Ai)=1 (PC) ← (PC) + 2	xxxxxxxx	i i i 1 0 0 1 1 <B2>	(2i+1) × 10+3	2	4	4
		BBC i, $zz,$hhll	When(Mi)=0 (PC) ← (PC) + 3 + Rel Where i=0~7 When(Mi)=1 (PC) ← (PC) + 3	xxxxxxxx	i i i 1 0 1 1 1 <B2> <B3>	(2i+1) × 10+3	3	5	4
		BBS i,A,$hhll	When(Ai)=1 (PC) ← (PC) + 2 + Rel Where i=0~7 When(Ai)=0 (PC) ← (PC) + 2	xxxxxxxx	i i i 0 0 0 1 1 <B2>	2i × 10+3	2	4	4
		BBS i, $zz,$hhll	When(Mi)=1 (PC) ← (PC) + 3 + Rel Where i=0~7 When(Mi)=0 (PC) ← (PC) + 3	xxxxxxxx	i i i 0 0 1 1 1 <B2> <B3>	2i × 10+7	3	5	4
		BCC $hhll	When(C)=0 (PC) ← (PC) + 2 + Rel When(C)=1 (PC) ← (PC) + 2	xxxxxxxx	1 0 0 1 0 0 0 0 <B2>	90	2	2	4
		BCS $hhll	When(C)=1 (PC) ← (PC) + 2 + Rel When(C)=0 (PC) ← (PC) + 2	xxxxxxxx	1 0 1 1 0 0 0 0 <B2>	B0	2	2	4
		BNE $hhll	When(Z)=0 (PC) ← (PC) + 2 + Rel When(Z)=1 (PC) ← (PC) + 2	xxxxxxxx	1 1 0 1 0 0 0 0 <B2>	D0	2	2	4
		BEQ $hhll	When(Z)=1 (PC) ← (PC) + 2 + Rel When(Z)=0 (PC) ← (PC) + 2	xxxxxxxx	1 1 1 1 0 0 0 0 <B2>	F0	2	2	4
		BPL $hhll	When(N)=0 (PC) ← (PC) + 2 + Rel When(N)=1 (PC) ← (PC) + 2	xxxxxxxx	0 0 0 1 0 0 0 0 <B2>	10	2	2	4
		BMI $hhll	When(N)=1 (PC) ← (PC) + 2 + Rel When(N)=0 (PC) ← (PC) + 2	xxxxxxxx	0 0 1 1 0 0 0 0 <B2>	30	2	2	4
		BVC $hhll	When(V)=0 (PC) ← (PC) + 2 + Rel When(V)=1 (PC) ← (PC) + 2	xxxxxxxx	0 1 0 1 0 0 0 0 <B2>	50	2	2	4
		BVS $hhll	When(V)=1 (PC) ← (PC) + 2 + Rel When(V)=0 (PC) ← (PC) + 2	xxxxxxxx	0 1 1 1 0 0 0 0 <B2>	70	2	2	
	Return	RTI	(S) ← (S) + 1, (PS) ← (M(S)), (S) ← (S) + 1, (PC_L) ← (M(S)), (S) ← (S) + 1, and (PC_H) ← (M(S))	Previous status in stack	0 1 0 0 0 0 0 0	40	1	6	
		RTS	(S) ← (S) + 1, (PC_L) ← (M(S)), (S) ← (S) + 1, (PC_H) ← (M(S)), and (PC) ← (PC) +1	xxxxxxxx	0 1 1 0 0 0 0 0	60	1	6	
Interrupt		BRK	(B) ← 1, (PC) ← (PC) + 2, (M(S)) ← (PC_H), (S) ← (S)—1, (M(S)) ← (PC_L), (S) ← (S)—1, (I) ← 1, and (PC) ← BADRS	xxx1x1xx	0 0 0 0 0 0 0 0	00	1	7	
Other		NOP	(PC) ← (PC) + 1	xxxxxxxx	1 1 1 0 1 0 1 0	EA	1	2	
Special		WIT	Internal clock source is stopped.	xxxxxxxx	1 1 0 0 0 0 1 0	C2	1	2	
		STP	Oscillation is stopped.	xxxxxxxx	0 1 0 0 0 0 1 0	42	1	2	

APPENDIX 7

Machine language instruction table

Symbol	Means
A	**Accumulator**
Ai	**Bit i of accumulator**
X	**Index register X**
Y	**Index register Y**
M	**Memory**
Mi	**Bit i of memory**
PS	**Processor status register**
S	**Stack pointer**
PC	**Program counter**
PC_L	**Lower byte of program counter**
PC_H	**Higher byte of program counter**
N	**Negative flag**
V	**Overflow flag**
T	**X modified operation mode flag**
B	**Break flag**
D	**Decimal mode flag**
I	**Interrupt disable flag**
Z	**Zero flag**
C	**Carry flag**
#	**Immediate mode**
$	**Hexadecimal**

Symbol	Means
\	**Special page mode**
hh	**Higher byte of address (0~255)**
ll	**Lower byte of address (0~255)**
zz	**Zero Page address (0~255)**
nn	**Data at 0~255**
i	**Data at 0~7**
iii	**Data at 0~7**
<B2>	**Second byte of instruction**
<B3>	**Third byte of instruction**
Rel	**Relative address**
BADRS	**Break address**
←	**Direction of data transfer**
( )	**Contents of register or memory**
+	**Add**
—	**Subtract**
∨	**Logical OR**
∧	**Logical AND**
∀	**Logical Exclusive OR**
‾	**Negative**
×	**Stable flag after execution**
○	**Variable flag after execution**

Note 1 : Listed function is when (T) = 0.
When (T) = 1, (M(X)) is entered instead of (A) and the cycle number is increased by 3.
Note 2 : Ditto. The cycle number is increased by 2.
Note 3 : Ditto. The cycle number is increased by 1.
Note 4 : The cycle number is increased by 2 when a branch is occurred.

Appendix 8 Instruction code table

D7~D4 \ D3~D0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND,X	JSR ZP,IND	BBS 0,A	—	ORA ZP	ASL ZP	BBS 0,ZP	PHP	ORA IMM	ASL A	SEB 0,A	—	ORA ABS	ASL ABS	SEB 0,ZP
0001	1	BPL	ORA IND,Y	CLT	BBC 0,A	—	ORA ZP,X	ASL ZP,X	BBC 0,ZP	CLC	ORA ABS,Y	DEC A	CLB 0,A	—	ORA ABS,X	ASL ABS,X	CLB 0,ZP
0010	2	JSR ABS	AND IND,X	JSR SP	BBS 1,A	BIT ZP	AND ZP	ROL ZP	BBS 1,ZP	PLP	AND IMM	ROL A	SEB 1,A	BIT ABS	AND ABS	ROL ABS	SEB 1,ZP
0011	3	BMI	AND IND,Y	SET	BBC 1,A	—	AND ZP,X	ROL ZP,X	BBC 1,ZP	SEC	AND ABS,Y	INC A	CLB 1,A	LDM ZP	AND ABS,X	ROL ABS,X	CLB 1,ZP
0100	4	RTI	EOR IND,X	STP	BBS 2,A	COM ZP	EOR ZP	LSR ZP	BBS 2,ZP	PHA	EOR IMM	LSR A	SEB 2,A	JMP ABS	EOR ABS	LSR ABS	SEB 2,ZP
0101	5	BVC	EOR IND,Y	—	BBC 2,A	—	EOR ZP,X	LSR ZP,X	BBC 2,ZP	CLI	EOR ABS,Y	—	CLB 2,A	—	EOR ABS,X	LSR ABS,X	CLB 2,ZP
0110	6	RTS	ADC IND,X	—	BBS 3,A	TST ZP	ADC ZP	ROR ZP	BBS 3,ZP	PLA	ADC IMM	ROR A	SEB 3,A	JMP IND	ADC ABS	ROR ABS	SEB 3,ZP
0111	7	BVS	ADC IND,Y	—	BBC 3,A	—	ADC ZP,X	ROR ZP,X	BBC 3,ZP	SEI	ADC ABS,Y	—	CLB 3,A	—	ADC ABS,X	ROR ABS,X	CLB 3,ZP
1000	8	BRA	STA IND,X	RRF ZP	BBS 4,A	STY ZP	STA ZP	STX ZP	BBS 4,ZP	DEY	—	TXA	SEB 4,A	STY ABS	STA ABS	STX ABS	SEB 4,ZP
1001	9	BCC	STA IND,Y	—	BBC 4,A	STY ZP,X	STA ZP,X	STX ZP,X	BBC 4,ZP	TYA	STA ABS,Y	TXS	CLB 4,A	—	STA ABS,X	—	CLB 4,ZP
1010	A	LDY IMM	LDA IND,X	LDX IMM	BBS 5,A	LDY ZP	LDA ZP	LDX ZP	BBS 5,ZP	TAY	LDA IMM	TAX	SEB 5,A	LDY ABS	LDA ABS	LDX ABS	SEB 5,ZP
1011	B	BCS	LDA IND,Y	JMP ZP,IND	BBC 5,A	LDY ZP,X	LDA ZP,X	LDX ZP,Y	BBC 5,ZP	CLV	LDA ABS,Y	TSX	CLB 5,A	LDY ABS,X	LDA ABS,X	LDX ABS,Y	CLB 5,ZP
1100	C	CPY IMM	CMP IND,X	WIT	BBS 6,A	CPY ZP	CMP ZP	DEC ZP	BBS 6,ZP	INY	CMP IMM	DEX	SEB 6,A	CPY ABS	CMP ABS	DEC ABS	SEB 6,ZP
1101	D	BNE	CMP IND,Y	—	BBC 6,A	—	CMP ZP,X	DEC ZP,X	BBC 6,ZP	CLD	CMP ABS,Y	—	CLB 6,A	—	CMP ABS,X	DEC ABS,X	CLB 6,ZP
1110	E	CPX IMM	SBC IND,X	—	BBS 7,A	CPX ZP	SBC ZP	INC ZP	BBS 7,ZP	INX	SBC IMM	NOP	SEB 7,A	CPX ABS	SBC ABS	INC ABS	SEB 7,ZP
1111	F	BEQ	SBC IND,Y	—	BBC 7,A	—	SBC ZP,X	INC ZP,X	BBC 7,ZP	SED	SBC ABS,Y	—	CLB 7,A	—	SBC ABS,X	INC ABS,X	CLB 7,ZP

Appendix 9 Mask ROM ordering method

When placing an order, submit the following.

(1) M37470M2-XXXSP mask ROM ordering
- M37470M2-XXXSP prescribed confirmation statements (use the pages 139 and 140)
- EPROMs in which the data are written 3 sets DIP type 27128, 27256 or 27512
- 32P4B mark specification statement (use the page 151)

(2) M37470M4-XXXSP mask ROM ordering
- M37470M4-XXXSP prescribed confirmation statements (use the pages 141 and 142)
- EPROMs in which the data are written 3 sets DIP type 27128, 27256 or 27512
- 32P4B mark specification statement (use the page 151)

(3) M37470M8-XXXSP mask ROM ordering
- M37470M8-XXXSP prescribed confirmation statements (use the pages 143 and 144)
- EPROMs in which the data are written 3 sets DIP type 27256 or 27512
- 32P4B mark specification statement (use the page 151)

(4) M37471M2-XXXSP mask ROM ordering
- M37471M2-XXXSP/FP prescribed confirmation statements (use the pages 145 and 146)
- EPROMs in which the data are written 3 sets DIP type 27128, 27256 or 27512
- 42P4B mark specification statement (use the page 152)

(5) M37471M2-XXXFP mask ROM ordering
- M37471M2-XXXSP/FP prescribed confirmation statements (use the pages 145 and 146)
- EPROMs in which the data are written 3 sets DIP type 27128, 27256 or 27512
- 56P6N mark specification statement (use the page 153)

(6) M37471M4-XXXSP mask ROM ordering
- M37471M4-XXXSP/FP prescribed confirmation statements (use the pages 147 and 148)
- EPROMs in which the data are written 3 sets DIP type 27128, 27256 or 27512
- 42P4B mark specification statement (use the page 152)

(7) M37471M4-XXXFP mask ROM ordering
- M37471M4-XXXSP/FP prescribed confirmation statements (use the pages 147 and 148)
- EPROMs in which the data are written 3 sets DIP type 27128, 27256 or 27512
- 56P6N mark specification statement (use the page 153)

(8) M37471M8-XXXSP mask ROM ordering
- M37471M8-XXXSP/FP prescribed confirmation statements (use the pages 149 and 150)
- EPROMs in which the data are written 3 sets DIP type 27256 or 27512
- 42P4B mark specification statement (use the page 152)

(9) M37471M8-XXXFP mask ROM ordering
- M37471M8-XXXSP/FP prescribed confirmation statements (use the pages 149 and 150)
- EPROMs in which the data are written 3 sets DIP type 27128, 27256 or 27512
- 56P6N mark specification statement (use the page 153)

GZZ–SH02–91A‹9YA0›

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37470M2-XXXSP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※.

※				Issuance signature	Responsible officer	Supervisor
	Customer	Company name	TEL ( )			
		Date issued	Date :			

※1 . Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based in this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM type

☐ 27128	☐ 27256	☐ 27512
Address	Address	Address
0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37470M2–'	0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37470M2–'	0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37470M2–'
0010_{16}–$2FFF_{16}$: (shaded)	0010_{16}–$6FFF_{16}$: (shaded)	0010_{16}–$EFFF_{16}$: (shaded)
3000_{16}–$3FFF_{16}$: ROM(4 K)	7000_{16}–$7FFF_{16}$: ROM(4 K)	$F000_{16}$–$FFFF_{16}$: ROM(4 K)

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37470M2—' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37470M2—' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = 4 D_{16}	0008_{16}	'—' = 2 D_{16}
0001_{16}	'3' = 3 3_{16}	0009_{16}	F F_{16}
0002_{16}	'7' = 3 7_{16}	$000A_{16}$	F F_{16}
0003_{16}	'4' = 3 4_{16}	$000B_{16}$	F F_{16}
0004_{16}	'7' = 3 7_{16}	$000C_{16}$	F F_{16}
0005_{16}	'0' = 3 0_{16}	$000D_{16}$	F F_{16}
0006_{16}	'M' = 4 D_{16}	$000E_{16}$	F F_{16}
0007_{16}	'2' = 3 2_{16}	$000F_{16}$	F F_{16}

GZZ—SH02—91A< 9YA0 >

Mask ROM number	

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37470M2-XXXSP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	△＊＝△$C000 △.BYTE△ 'M37470M2—'	△＊＝△$8000 △.BYTE△ 'M37470M2—'	△＊＝△$0000 △.BYTE△ 'M37470M2—'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

※ 2. Mark specification
Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (32P4B for M37470M2-XXXSP) and attach to the mask ROM confirmation form.

※ 3. Comments

GZZ—SH02—92A＜9YA0＞

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37470M4-XXXSP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※.

※ Customer	Company name	TEL ( )	Issuance signature	Responsible officer	Supervisor
	Date issued	Date :			

※1 . Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based in this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas ☐☐☐☐ (hexadecimal notation)

EPROM type

☐ 27128	☐ 27256	☐ 27512
Address	Address	Address
0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37470M4—'	0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37470M4—'	0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37470M4—'
0010_{16}–$1FFF_{16}$: (shaded)	0010_{16}–$5FFF_{16}$: (shaded)	0010_{16}–$DFFF_{16}$: (shaded)
2000_{16}–$3FFF_{16}$: ROM(8 K)	6000_{16}–$7FFF_{16}$: ROM(8 K)	$E000_{16}$–$FFFF_{16}$: ROM(8 K)

(1) Set "FF_{16}" in the shaded area.

(2) Write the ASCII codes that indicates the name of the product 'M37470M4—' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37470M4—' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	'—' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'7' = 37_{16}	$000C_{16}$	FF_{16}
0005_{16}	'0' = 30_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

GZZ–SH02–92A<9YA0>

Mask ROM number	

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37470M4-XXXSP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	△∗=△$C000 △.BYTE△ 'M37470M4–'	△∗=△$8000 △.BYTE△ 'M37470M4–'	△∗=△$0000 △.BYTE△ 'M37470M4–'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

※2. Mark specification
Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (32P4B for M37470M4-XXXSP) and attach to the mask ROM confirmation form.

※3. Comments

GZZ—SH02—93A‹9YA0›

Mask ROM number	

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37470M8-XXXSP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

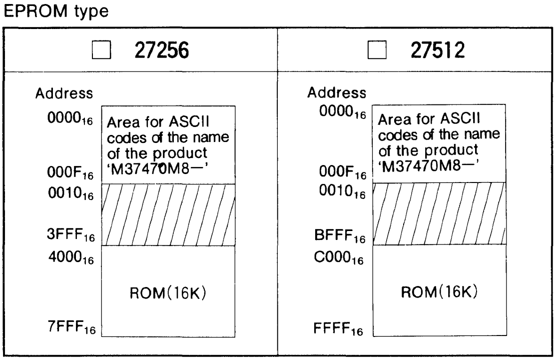
※	Customer	Company name	TEL ( )	Issuance signature	Responsible officer	Supervisor
		Date issued	Date :			

※ 1 . Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based in this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM type



(1) Set "FF$_{16}$" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37470M8—' to addresses 0000$_{16}$ to 000F$_{16}$. ASCII codes 'M37470M8—' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000$_{16}$	'M' = 4 D$_{16}$	0008$_{16}$	'A' = 4 1$_{16}$
0001$_{16}$	'3' = 3 3$_{16}$	0009$_{16}$	F F$_{16}$
0002$_{16}$	'7' = 3 7$_{16}$	000A$_{16}$	F F$_{16}$
0003$_{16}$	'4' = 3 4$_{16}$	000B$_{16}$	F F$_{16}$
0004$_{16}$	'7' = 3 7$_{16}$	000C$_{16}$	F F$_{16}$
0005$_{16}$	'0' = 3 0$_{16}$	000D$_{16}$	F F$_{16}$
0006$_{16}$	'M' = 4 D$_{16}$	000E$_{16}$	F F$_{16}$
0007$_{16}$	'8' = 3 8$_{16}$	000F$_{16}$	F F$_{16}$

GZZ—SH02—93A< 9YA0 >

Mask ROM number	

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37470M8-XXXSP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27256	27512
The pseudo-command	△* =△$8000 △.BYTE△ 'M37470M8—'	△* =△$0000 △.BYTE△ 'M37470M8—'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

※2. Mark specification
Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (32P4B for M37470M8-XXXSP) and attach to the mask ROM confirmation form.

※3. Comments

GZZ–SH02–94A<9YB0>

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37471M2-XXXSP/FP
MITSUBISHI ELECTRIC

Mask ROM number	

	Date :	
Receipt	Section head signature	Supervisor signature

Note : Please fill in all items marked※.

					Responsible officer	Supervisor
※	Customer	Company name	TEL ( )	Issuance signature		
		Date issued	Date :			

※ 1 . Confirmation
Specify the name of the product being ordered and the type of EPROMs submitted.
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based in this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Microcomputer name : ☐ M37471M2-XXXSP ☐ M37471M2-XXXFP

Checksum code for entire EPROM areas ☐☐☐☐ (hexadecimal notation)

EPROM type

☐ 27128	☐ 27256	☐ 27512
Address 0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37471M2—'	Address 0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37471M2—'	Address 0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37471M2—'
0010_{16}–$2FFF_{16}$: (shaded)	0010_{16}–$6FFF_{16}$: (shaded)	0010_{16}–$EFFF_{16}$: (shaded)
3000_{16}–$3FFF_{16}$: ROM(4 K)	7000_{16}–$7FFF_{16}$: ROM(4 K)	$F000_{16}$–$FFFF_{16}$: ROM(4 K)

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37471M2—' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37471M2—' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	'—' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'7' = 37_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'2' = 32_{16}	$000F_{16}$	FF_{16}

GZZ—SH02—94A< 9YB0 >

Mask ROM number	

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37471M2-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	△*＝△$C000 △.BYTE△ 'M37471M2—'	△*＝△$8000 △.BYTE△ 'M37471M2—'	△*＝△$0000 △.BYTE△ 'M37471M2—'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

※ 2. Mark specification
Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (42P4B for M37471M2-XXXSP ; 56P6N for M37471M2-XXXFP) and attach to the mask ROM confirmation form.

※ 3. Comments

APPENDIX 9

Mask ROM ordering method

GZZ–SH02–95A＜9YB0＞

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37471M4-XXXSP/FP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※.

※ Customer	Company name	TEL ( )	Issuance signature	Responsible officer	Supervisor
	Date issued	Date :			

※1 . Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based in this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Microcomputer name : ☐ M37471M4-XXXSP ☐ M37471M4-XXXFP

Checksum code for entire EPROM areas [| | |] (hexadecimal notation)

EPROM type

☐ 27128	☐ 27256	☐ 27512
Address	Address	Address
0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37471M4—'	0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37471M4—'	0000_{16}–$000F_{16}$: Area for ASCII codes of the name of the product 'M37471M4—'
0010_{16}–$1FFF_{16}$: (shaded)	0010_{16}–$5FFF_{16}$: (shaded)	0010_{16}–$DFFF_{16}$: (shaded)
2000_{16}–$3FFF_{16}$: ROM(8 K)	6000_{16}–$7FFF_{16}$: ROM(8 K)	$E000_{16}$–$FFFF_{16}$: ROM(8 K)

(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37471M4—' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37471M4—' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	'—' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'7' = 37_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

GZZ—SH02—95A< 9YB0 >

Mask ROM number	

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37471M4-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27128	27256	27512
The pseudo-command	△∗＝△$C000 △.BYTE△ 'M37471M4—'	△∗＝△$8000 △.BYTE△ 'M37471M4—'	△∗＝△$0000 △.BYTE△ 'M37471M4—'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

※2. Mark specification
Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (42P4B for M37471M4-XXXSP ; 56P6N for M37471M4-XXXFP) and attach to the mask ROM confirmation form.

※3. Comments

GZZ–SH02–96A‹9YB0›

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37471M8-XXXSP/FP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※.

※ Customer	Company name	TEL ( )	Issuance signature	Responsible officer	Supervisor
	Date issued	Date :			

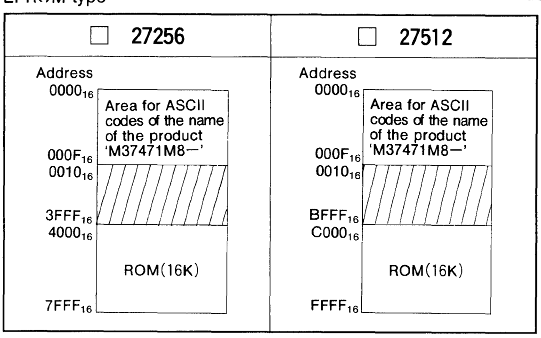
※1 . Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based in this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Microcomputer name : ☐ M37471M8-XXXSP ☐ M37471M8-XXXFP

Checksum code for entire EPROM areas ☐☐☐☐ (hexadecimal notation)

EPROM type



(1) Set "FF_{16}" in the shaded area.
(2) Write the ASCII codes that indicates the name of the product 'M37471M8—' to addresses 0000_{16} to $000F_{16}$. ASCII codes 'M37471M8—' are listed on the right. The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	'—' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'7' = 37_{16}	$000A_{16}$	FF_{16}
0003_{16}	'4' = 34_{16}	$000B_{16}$	FF_{16}
0004_{16}	'7' = 37_{16}	$000C_{16}$	FF_{16}
0005_{16}	'1' = 31_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

GZZ—SH02—96A< 9YB0 >

Mask ROM number	

SERIES MELPS 740 MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37471M8-XXXSP/FP
MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the start address of the assembler source program.

EPROM type	27256	27512
The pseudo-command	△*＝△$8000 △.BYTE△ 'M37471M8—'	△*＝△$0000 △.BYTE△ 'M37471M8—'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

※ 2 . Mark specification
Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (42P4B for M37471M8-XXXSP ; 56P6N for M37471M8-XXXFP) and attach to the mask ROM confirmation form.

※ 3 . Comments

APPENDIX 9

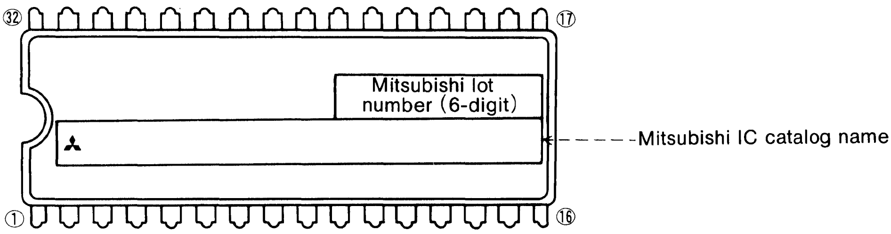
Mask ROM ordering method

32P4B (32-PIN SHRINK DIP) MARK SPECIFICATION FORM

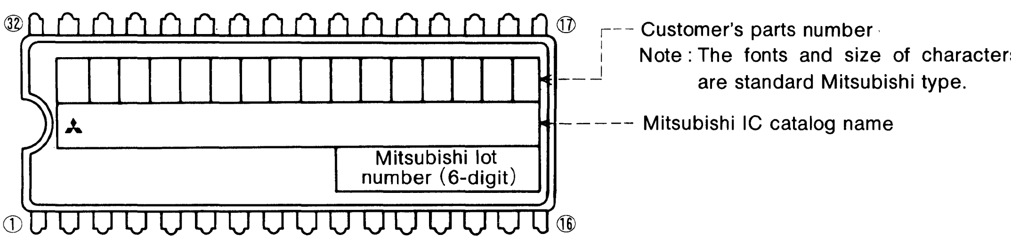
Mitsubishi IC catalog name

Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



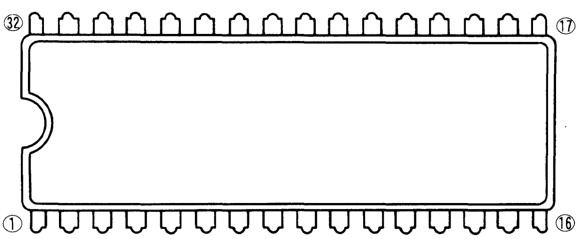
B. Customer's Parts Number + Mitsubishi Catalog Name



Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 16 characters :
Only 0~9, A~Z, +, −, /, (,), &, ©, · (period), and, (commas) are usable.
4 : If the Mitsubishi logo is not required, check the box on the right.

Mitsubishi logo is not required ☐

C. Special Mark Required



Note1 : If the special mark is to be printed, indicate the desired layout of the mark in the upper figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit) and mask ROM number (3-digit) are always marked.
2 : If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special logo required ☐

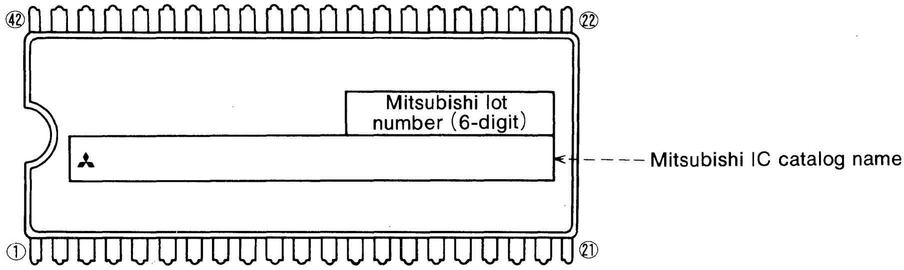
The standard Mitsubishi font is used for all characters except for a logo.

42P4B (42-PIN SHRINK DIP) MARK SPECIFICATION FORM

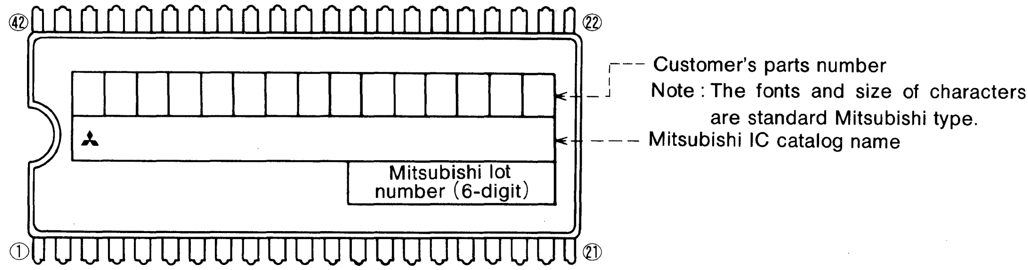
Mitsubishi IC catalog name

Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



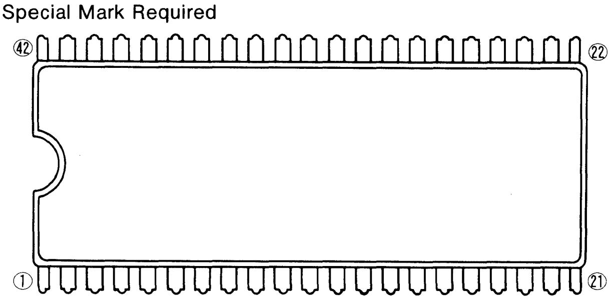
B. Customer's Parts Number + Mitsubishi Catalog Name



Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 15 characters :
Only 0~9, A~Z, +, −, ⁄, (,), &, ©, • (period), and, (commas) are usable.
4 : If the Mitsubishi logo is not required, check the box on the right.

Mitsubishi logo is not required ☐

C. Special Mark Required



Note1 : If the special mark is to be printed, indicate the desired layout of the mark in the upper figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit) and mask ROM number (3-digit) are always marked.
2 : If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.
The standard Mitsubishi font is used for all characters except for a logo.

Special logo required ☐

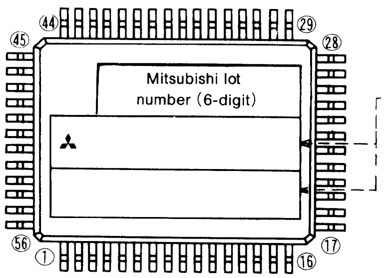
Mask ROM ordering method

56P6N (56-PIN QFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

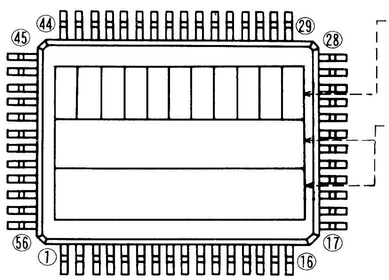
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

B. Customer's Parts Number + Mitsubishi Catalog Name



Customer's parts number
Note : The fonts and size of characters are standard Mitsubishi type.

Mitsubishi IC catalog name and Mitsubishi lot number

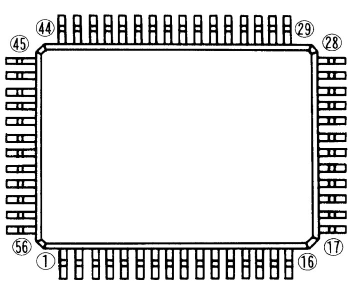
Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 11 characters :
Only 0~9, A~Z, +, −, ⁄, (,), &, ©, • (period), and, (commas) are usable.

Note4 : If the Mitsubishi logo is not required, check the box below.
Mitsubishi logo is not required

☐

5 : Arrangement of Mitsubishi IC catalog name and Mitsubishi lot number is dependent on number of Mitsubishi IC catalog name and that Mitsubishi logo is required or not.

C. Special Mark Required



Note1 : If the special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit) and mask ROM number (3-digit) are always marked.
2 : If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.
Special logo required

☐

The standard Mitsubishi font is used for all characters except for a logo.

Appendix 10 Emulator MCU M37471RSS

1. Outline

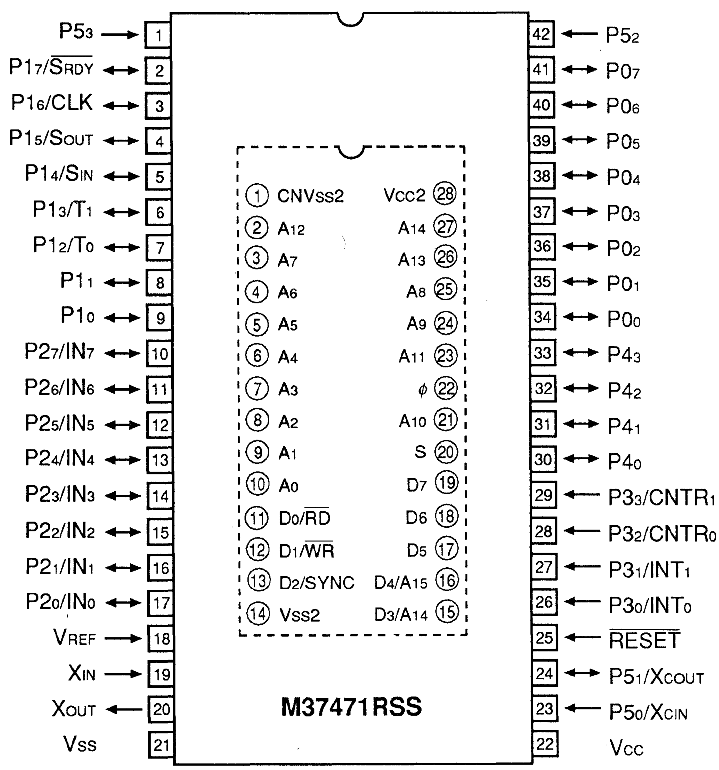
The emulator MCU M37471RSS is designed for the development of 7470 Series software. It has an emulator connected to a socket in its top surface, and it enable efficient debugging of user programs, with the help of functions such as real-time trace.

Signals from a 16-line address bus and a bi-directional data bus, and SYNC, $\overline{RD}$, $\overline{WR}$, and ϕ signals can be output from the socket in the top surface, to enable monitoring of all the internal bus data from the emulator.

A debugging system using the M37471RSS enables direct connection of the pins of the dedicated MCU for emulator to the user system, to provide a debugging environment even closer to real life.

Mitsubishi's PC4600 system is a development support system suitable for the M37471RSS.

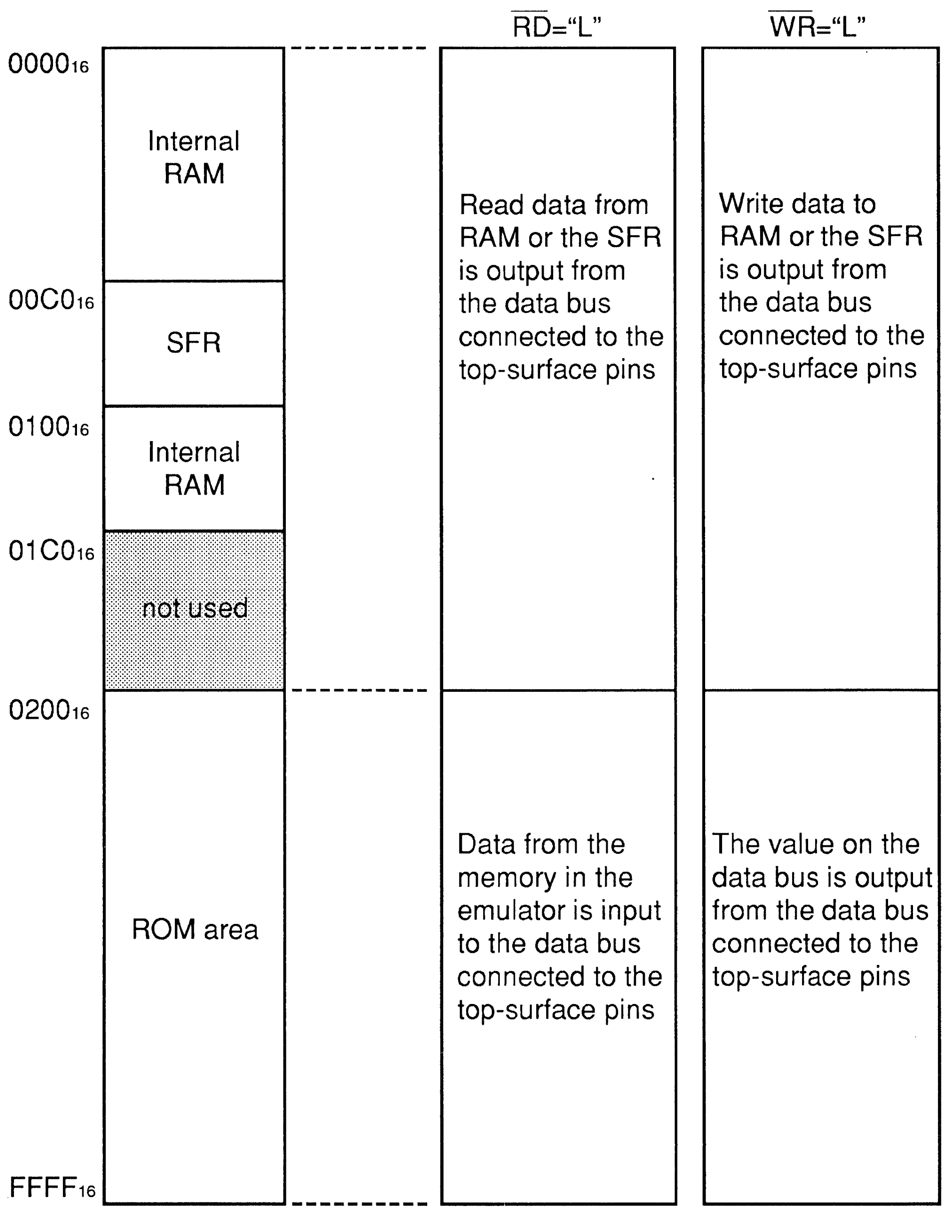
Pin configuration (top view)



Outline 42S1M

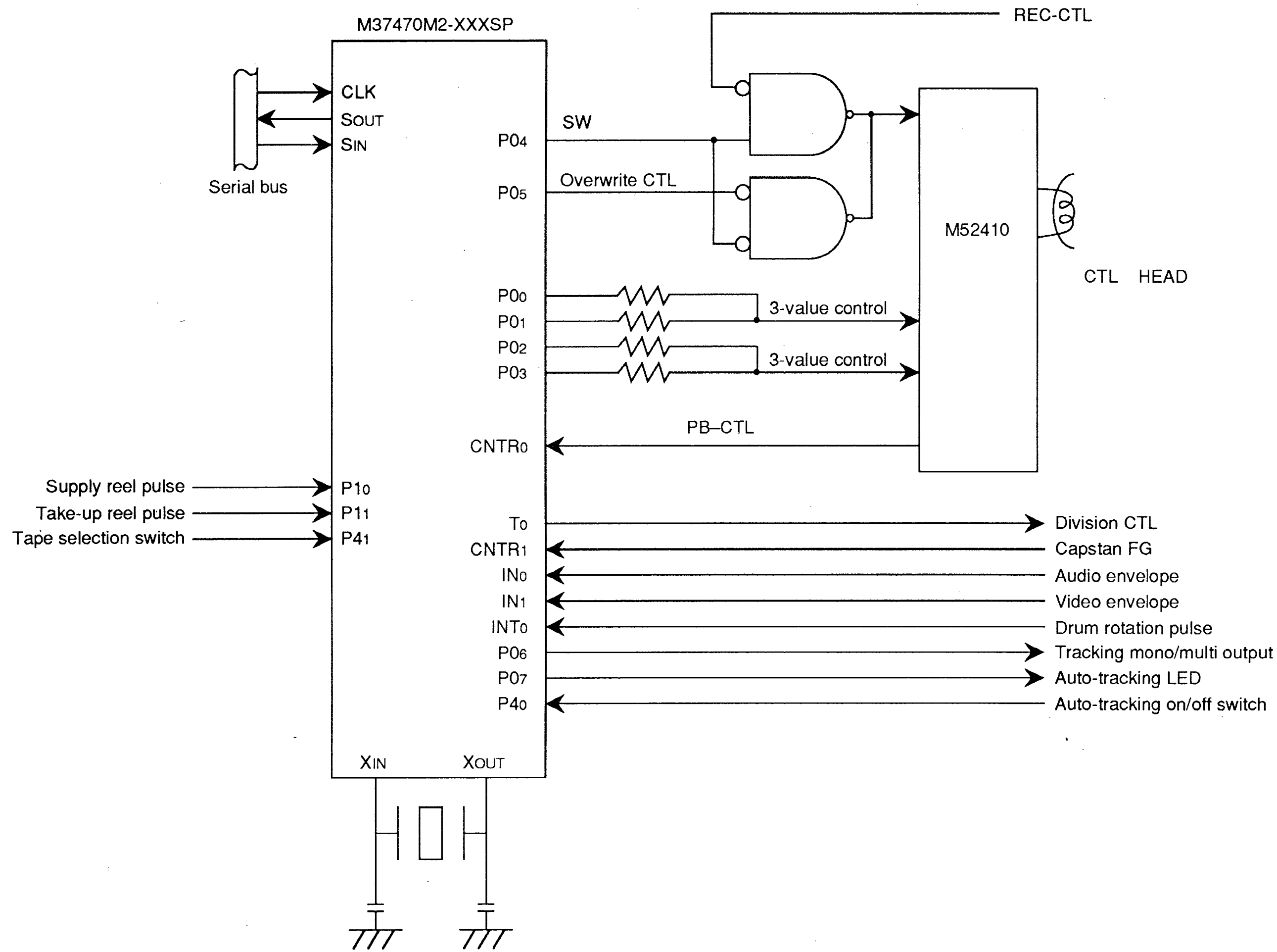
Note: The power supply voltage of the M37471RSS must be within the range of 5V±5%.

2. M37471RSS memory map



Note : It can be set the S pin level either "L" or "H".

Appendix 11 Applecation circuit example



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) 3218-3473
(03) 3218-3499
Facsimile: (03) 3214-5570

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.
41st fl., Manulife Tower, 169,
Electric Road, North Point, Hong Kong
Telex: 60800 MELCO HX
Telephone: 510-0555
Facsimile: 510-9830, 510-9822,
510-9803

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.
16980 Via Tazon, Suite 220
San Diego, CA 92128
Telephone: (619) 451-9618
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.
348 Victoria Road
Rydalmere Nsw 2116, Australia
Private Bag No.2 Rydalmere Nsw 2116
Telex: MESYDAA 126614
Telephone: (02) 684-7200
Facsimile: (02) 638-7072

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL 7470 Series

Sep. First Edition 1991
Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

MITSUBISHI SEMICONDUCTORS
7470 Series

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: MITSUBISHI DENKI BLDG., MARUNOUCHI, TOKYO 100. TELEX: J24532 CABLE: MELCO TOKYO

If these products or technologies fall under Japanese and/or COCOM strategic restrictions, diversion contrary thereto is prohibited.

H-E0495-A KI-9109 Printed in Japan (ROD)

New publication, effective Sep. 1991.
Specifications subject to change without notice.